



HAL
open science

Représentation tridimensionnelle et reconstruction 3D à partir de coupes 2D

Franck Rolland

► **To cite this version:**

Franck Rolland. Représentation tridimensionnelle et reconstruction 3D à partir de coupes 2D. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1991. Français. NNT: . tel-00339648

HAL Id: tel-00339648

<https://theses.hal.science/tel-00339648>

Submitted on 18 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TU 14034

THESE

présentée par

Franck ROLLAND

pour obtenir le titre de

Docteur de l'Université Joseph Fourier GRENOBLE I

(arrêté ministériel du 5 juillet 1984)

Spécialité Informatique

Représentation tridimensionnelle et Reconstruction 3D à partir de Coupes 2D

Thèse soutenue le 3 Septembre 1991 devant la commission d'examen.

Marc RICHETIN	Président
Philippe CINQUIN Francis SCHMITT	Rapporteurs
Nicholas AYACHE Gilles BERTRAND Jean-Marc CHASSERY Annick MONTANVERT	Examineurs

Thèse préparée au sein de l'équipe RFMQ du laboratoire TIM3 - IMAG

A Valérie

Remerciements

Je tiens à exprimer mes plus sincères remerciements à :

Monsieur *Marc RICHETIN*, Professeur à l'Université Blaise Pascal de Clermont-Ferrand, pour m'avoir fait l'honneur de bien vouloir présider ce jury ;

Messieurs *Philippe CINQUIN* et *Francis SCHMITT*, respectivement Professeur à l'Université Joseph Fourier de Grenoble et Professeur à l'Ecole Nationale Supérieure des Télécommunications de Paris, qui ont accepté d'être les rapporteurs de ce mémoire ;

Monsieur *Nicholas AYACHE*, Directeur de Recherches à l'INRIA et Monsieur *Gilles BERTRAND*, Enseignant Chercheur à l'ESIEE, qui ont bien voulu faire partie des examinateurs ;

Monsieur *Jean-Marc CHASSERY*, Directeur de Recherches CNRS, pour les conseils précieux qu'il a toujours su m'apporter, ainsi que pour la confiance qu'il m'a accordée ;

Madame *Annick MONTANVERT*, Maître de Conférences à l'Université Pierre Mendès-France de Grenoble, qui m'a encadré tout au long de cette recherche, pour la disponibilité dont elle a fait preuve. Je lui suis également reconnaissant de ses conseils et des encouragements qu'elle m'a dispensés ;

Monsieur *Gérard BRUGAL*, Professeur à l'Université Joseph Fourier de Grenoble, Directeur de l'équipe RFMQ pour son dynamisme communicatif ;

Monsieur *Guy BOURREL*, Ingénieur CNRS, pour sa disponibilité et sa promptitude à résoudre tous les problèmes machine ;

Les “informaticiens”, “mathématiciens”, “biologistes”, “musiciens”, “philosophes”, “volleyeurs”, et “skieurs “ de tous bords qui constituent l’équipe RFMQ, pour l’ambiance qu’ils contribuent à faire régner.

Ma famille (*Pierre, Suzanne, Olivier, Patrice, Gisèle, Bernard, Martine, Mael, Alwin*) pour leurs encouragements sans cesse renouvelés.

Valérie, pour elle et pour tout ce que nous partageons.

Table des matières

Remerciements.....	1
Table des matières.....	3
Introduction.....	9
1. Problématique : la reconstruction tridimensionnelle.....	11
1.1. Reconstruction de formes tridimensionnelles.....	11
1.1.1. Notion d'objet.....	11
1.1.2. Saisie de la surface.....	12
1.1.3. Inférence : passage du plan à l'espace.....	12
1.1.4. Saisie volumique.....	13
1.1.5. Utilisation et nécessité de l'obtention d'un modèle tridimensionnel.....	14
1.2. Situation du problème	15
1.3. Les moyens existants.....	18
1.3.1. Acquisition et prétraitement des données.....	18
1.3.2. Méthodes de reconstruction tridimensionnelle.....	21
1.3.2.1. Méthodes par triangulation des contours.....	21
<i>Principe de base</i>	22
<i>Améliorations</i>	23

1.3.2.2.	Méthode utilisant la triangulation de Delaunay	29
1.3.2.3.	Méthodes par inférence de formes 2D.....	31
1.3.3.	Visualisation du résultat.....	33
2.	Modèles en analyse de formes : aspect bidimensionnel.....	35
2.1.	Définitions et notions de base	35
2.1.1.	Images discrètes binaires.....	35
2.1.2.	Distances dans \mathbb{Z}^2	37
2.1.2.1.	Distance euclidienne.....	37
2.1.2.2.	Distance de Manhattan	37
2.1.2.3.	Distance de l'échiquier	38
2.1.2.4.	Distances du chanfrein	39
2.1.3.	Voisinage et connexité.....	41
2.2.	Modèles de représentation	46
2.2.1.	Contour et approximations polygonales.....	46
2.2.2.	Les arbres.....	47
2.2.3.	Les squelettes.....	50
2.2.4.	Axe médian et ligne médiane	52
2.2.5.	Représentations hiérarchiques.....	54
2.2.6.	Le graphe de la ligne médiane	54
3.	Modèles en analyse de formes : aspect tridimensionnel.....	57
3.1.	Définitions et notions de base	57
3.1.1.	Espace discret	57
3.1.2.	Distances dans \mathbb{Z}^3	59
3.1.2.1.	Distance euclidienne.....	59
3.1.2.2.	Distance d_6	59
3.1.2.3.	Distance d_{26}	61
3.1.2.4.	Distance d_{18}	62
3.1.2.5.	Distances du chanfrein	63
3.1.3.	Connexité, nombre d'Euler.....	64
3.2.	Modèles de représentation tridimensionnels	67
3.2.1.	Représentation par modélisation de la surface.....	68
3.2.2.	Les arbres.....	68
3.2.3.	Axe médian	72
3.2.4.	Les squelettes.....	74

3.2.4.1.	Principe	74
3.2.4.2.	Squelette surfacique	76
3.2.4.3.	Squelette filaire.....	78
3.2.4.4.	Parcours d'un objet 3D	79
3.2.4.5.	Reconnexion de l'axe médian	81
3.2.4.6.	Les squelettes en tant que moyen de codage.....	83
3.2.5.	Liens entre modèles bidimensionnels et modèles tridimensionnels.....	85
4.	Inférence de formes pour la reconstruction tridimensionnelle.....	89
4.1.	Méthode des modèles élastiques	90
4.2.	Méthode utilisant un modèle hiérarchique.....	94
5.	Mise en correspondance de graphes.....	99
5.1.	Méthodes de mise en correspondance de graphes.....	99
5.1.1.	Définitions	99
5.1.2.	Méthodes de mise en correspondance	103
5.1.2.1.	Méthodes métriques	104
5.1.2.2.	Méthodes de recherche par retour arrière	105
5.1.2.3.	Méthodes par recherche de cliques	108
5.2.	Méthode de mise en correspondance de graphes de la ligne médiane.....	109
5.2.1.	Construction du graphe de mise en correspondance.....	109
5.2.1.1.	Principe général.....	109
5.2.1.2.	Construction du graphe de mise en correspondance.....	110
<i>Les sommets</i>	111	
<i>Les arêtes</i>	113	
<i>Pondération des sommets</i>	114	
5.2.2.	Recherche de cliques	120
5.2.2.1.	Recherches exhaustives.....	121
5.2.2.2.	Recherches heuristiques	123
<i>Le recuit simulé</i>	125	
<i>Algorithmes génétiques</i>	128	
<i>Remarques sur les recherches heuristiques</i>	132	
5.2.3.	Résultats de la recherche de cliques.....	133
5.3.	Résultats de la mise en correspondance de graphes de la ligne médiane	136

6. Inférence structurelle de graphes de la ligne médiane	137
6.1. Propagation des appariements et résultats finals de la mise en correspondance de graphes de la ligne médiane.....	137
6.1.1. Propagation des appariements.....	137
6.1.1.1. Appariement des proéminences larges.....	138
6.1.1.2. Appariement des proéminences strictes peu déformantes	141
6.1.2. Résultats de la mise en correspondance de graphes de la ligne médiane	143
6.2. Inférence de graphes de la ligne médiane	143
6.2.1. Appariement sommet principal \leftrightarrow sommet principal	144
6.2.2. Appariement sommet principal \leftrightarrow sommet de proéminence stricte	146
6.2.3. Appariement sommet principal \leftrightarrow sommet de proéminence large	147
6.2.4. Appariement sommet de proéminence stricte \leftrightarrow sommet de proéminence stricte	148
6.2.5. Appariement sommet de proéminence large \leftrightarrow sommet de proéminence large.....	149
6.2.6. Appariement sommet de proéminence stricte \leftrightarrow sommet de proéminence large.....	150
6.2.7. Appariement sommet de rétrécissement \leftrightarrow sommet de rétrécissement	151
6.2.8. Appariement sommet de rétrécissement \leftrightarrow sommet principal	151
6.2.9. Appariement sommet de rétrécissement \leftrightarrow sommet de proéminence stricte	152
6.2.10. Appariement sommet de rétrécissement \leftrightarrow sommet de proéminence large.....	153
6.2.11. Appariement multiple sommet principal \leftrightarrow sommets principaux	154
6.2.12. Appariement multiple sommet principal \leftrightarrow sommets principaux, sommets de rétrécissement.....	156
6.3. Résultats de l'inférence de graphes de la ligne médiane.....	157
6.3.1. Stratégie globale.....	157
6.3.2. Résultats	157

7. Résultats.....	163
7.1. Squelettisation tridimensionnelle.....	163
7.2. Mise en correspondance de graphes.....	165
7.3. Inférence de formes et reconstruction tridimensionnelle	167
7.4. Conclusions	170
Conclusion.....	171
Références bibliographiques	173
Annexe	187
Table des figures.....	205

Introduction

L'être humain est plongé dans un monde tridimensionnel. Pour pouvoir y évoluer il doit être capable d'appréhender la forme des objets qui l'entourent, mais aussi de modéliser leur disposition les uns par rapport aux autres. Le principal outil lui permettant de décrire ces objets et de modéliser son environnement est la *vision*.

Deux systèmes de vision sont utilisés. Le premier s'attache à la vision de près, le second à la vision plus lointaine. Le processus de vision de près utilise en priorité les informations issues du *système stéréoscopique* d'acquisition des images que représentent l'œil droit et œil gauche reliés à notre système très performant de corrélation et de mise en correspondance. Ce système permet de modéliser facilement la structure tridimensionnelle des objets.

La vision plus éloignée utilise plus spécialement d'autres informations appelées *indices visuels* (ombres, perspectives, occlusions, ...) qui permettent d'aider à la construction du modèle tridimensionnel de la scène. Cette construction est faite en association avec une sorte de base de connaissance de la vision, issue de l'expérience acquise, et régie par des processus d'*apprentissage*, qui permet d'infirmer ou de confirmer les diverses sensations perçues.

Le système de vision tridimensionnel humain est très performant dans l'environnement naturel pour lequel il a été créé et dans lequel il a évolué. Par contre, pour étudier d'autres environnements (l'infiniment petit ou l'infiniment grand), pour remplacer l'homme dans des environnements hostiles et pour "voir l'invisible" (infrarouge, vision à travers les matériaux,

contrôle non destructif), il est de plus en plus souvent nécessaire de remplacer le système de vision humain par des systèmes de vision artificiels.

Dans le domaine biomédical les objets étudiés (os, tissus, vaisseaux, noyaux, chromosomes, ...) ont des structures tridimensionnelles. Malheureusement, les systèmes microscopiques et macroscopiques ne permettent en général pas de rendre cet aspect tridimensionnel ; c'est pourquoi il est nécessaire de développer des outils permettant de *représenter, modéliser, décrire* et *manipuler* ces structures. Les méthodes de reconstruction tridimensionnelle à partir de coupes sériées permettent de modéliser de telles structures, les méthodes de squelettisation tridimensionnelle étant utilisées, elles, pour aborder les problèmes de description et de manipulation de ces mêmes structures.

Les différentes méthodes d'acquisition et de reconstruction tridimensionnelle sont décrites dans le premier chapitre. Sur les bases des modèles d'analyse et de représentation de formes bidimensionnelles (chapitre 2) sont développés les modèles utilisés pour l'analyse des formes tridimensionnelles (chapitre 3). En particulier, une nouvelle méthode de squelettisation tridimensionnelle est présentée dans ce chapitre. Les méthodes de reconstruction tridimensionnelle par inférence de formes sont présentées dans le chapitre 4. Après avoir mis en œuvre (chapitre 5) une méthode de mise en correspondance de graphes utilisant des recherches heuristiques dans un graphe, l'une des deux méthodes d'inférence de formes, fondée sur une inférence structurelle de graphes de la ligne médiane, est développée (chapitre 6). Le chapitre 7 est consacré à la présentation des résultats ainsi qu'à la description des limites et améliorations pouvant être apportées aux différentes méthodes développées dans les chapitres précédents.

Chapitre 1

Problématique : la reconstruction tridimensionnelle

1.1. Reconstruction de formes tridimensionnelles

1.1.1. Notion d'objet

Dans la suite nous serons souvent amenés à parler de 2D et de 3D. Le 2D représente tout ce qui est bidimensionnel, le 3D tout ce qui est tridimensionnel. Ainsi on désignera par espace 2D le plan et par espace 3D l'espace au sens spatial du terme.

Avant toute chose il convient de préciser ce que dans la suite on appelle "objet". Un objet est un ensemble de points connexes de l'espace (plan ou espace) pouvant être distingués parmi les autres points de l'espace. Deux types d'objets peuvent être rencontrés. Les objets binaires sont constitués de points ayant tous la même valeur (différente de la valeur des autres points de l'espace). Les objets non binaires (ou objets en niveaux de gris) sont constitués, eux, de points de valeurs différentes, dans ce cas il est plus difficile de savoir si un point fait partie ou non de l'objet.

La saisie d'objets peut être effectuée de différentes manières. Ce peut être une saisie de la surface de l'objet, ou alors une saisie volumique de celui-ci. Dans le premier cas, c'est à l'aide de sa surface que chaque objet est modélisé. Dans le second cas, les informations sur la

structure interne de l'objet sont nécessaires à sa modélisation. Chacune des deux modélisations implique des techniques d'acquisition tout à fait différentes. La saisie des objets se fait à l'aide de données qui peuvent être directement tridimensionnelles ou alors bidimensionnelles avec des hypothèses permettant la reconstruction. Ce sont ces différentes méthodes de saisie qui sont abordées dans la suite.

1.1.2. Saisie de la surface

Dans le cas d'objets binaires, les données tridimensionnelles sont généralement obtenues à l'aide de procédés physiques de mesure de l'objet. Ceux-ci peuvent être des systèmes d'acquisition manuels, semi-automatiques ou automatiques.

Les systèmes manuels nécessitent un relevé de points sur la surface de l'objet à construire. La saisie étant dépendante de l'opérateur, des problèmes de précision et de reproductibilité de l'acquisition se posent.

Les systèmes semi-automatiques peuvent être :

- soit des systèmes acoustiques, dans ce cas le calcul du temps de propagation entre un émetteur et des récepteurs permet de calculer, par triangulation, les coordonnées de l'objet à construire,

- soit des systèmes mécaniques, dans ce cas l'acquisition est faite grâce à un capteur tactile déplacé sur la surface de l'objet.

Ces systèmes peuvent être automatisés en programmant le déplacement des capteurs (systèmes tactiles) ou des récepteurs (systèmes acoustiques).

1.1.3. Inférence : passage du plan à l'espace

Des acquisitions bidimensionnelles de données peuvent permettre d'obtenir des coordonnées tridimensionnelles d'un objet.

C'est le cas avec les méthodes qui utilisent la combinaison de projections telles que la stéréovision et la méthode des points caractéristiques.

La méthode de stéréovision s'inspire du schéma de vision binoculaire : la mise en correspondance des images de l'œil droit et de l'œil gauche permet, par triangulation, d'obtenir des coordonnées tridimensionnelles. La mise en correspondance est faite par calcul du maximum de corrélation entre les deux images. Cette méthode est utilisée pour la robotique [Ayache 88] [Ayache 89], la médecine ou la cartographie (calcul de modèles numériques de terrain [Mémier 91]).

La méthode utilisant des points caractéristiques procède par projections des points caractéristiques de l'objet afin de calculer les coordonnées de la surface de cet objet. Cette méthode est utilisée en Conception Assistée par Ordinateur, et chaque fois qu'un positionnement d'un objet rigide est requis (photogrammétrie, souris tridimensionnelle par exemple).

Les coordonnées tridimensionnelles peuvent aussi être obtenues par traitement d'images de l'objet.

On distingue, les systèmes passifs et les systèmes actifs.

Les systèmes passifs utilisent les propriétés, lumineuses, de la texture ou des ombres de l'objet [Ballard 82]. Ils sont en fait les problèmes inverses des problèmes consistant, à partir d'un objet et d'une texture, par exemple, à appliquer cette texture sur l'objet (problème de placage de texture) [Gagalowicz 86] [Rao 90]. La photométrie s'appuie sur le calcul de la variation de l'intensité lumineuse réfléchi par l'objet pour déterminer son allure ; elle n'est applicable que dans certaines conditions très précises (objet de couleur uniforme, mat et avec une loi de réflexion bien connue). Les méthodes utilisant la texture sont utilisées sur des objets de texture connue, dans ce cas le calcul du gradient de texture permet d'obtenir des informations sur la profondeur et sur l'orientation de la surface de l'objet.

Les systèmes actifs font les acquisitions avec une lumière structurée. On peut citer les techniques de moiré (exploitation des figures d'interférences produites par projection d'un réseau de raies lumineuses sur l'objet) et de projection simple (la scène est éclairée par un faisceau de lumière cohérente ayant différentes formes contenant l'information de distance).

1.1.4. Saisie volumique

Toutes les techniques citées précédemment ne permettent d'obtenir que la surface de l'objet à reconstruire. Si l'on souhaite connaître l'objet autrement que par sa surface il est nécessaire d'opérer des coupes dans l'objet à reconstruire ; c'est cette méthode qui est généralement employée en médecine et en biologie. Du procédé d'obtention des coupes vont dépendre les méthodes de reconstruction.

Les coupes peuvent être réelles ou virtuelles, sériées, concentriques ou quelconques et être de véritables coupes ou des projections de points de l'objet sur un plan.

Les coupes réelles sont obtenues en découpant physiquement l'objet, c'est le cas en cytologie où les cellules sont sectionnées par ultramicrotomie.

Les coupes virtuelles sont obtenues en utilisant des propriétés physiques, acoustiques ou magnétiques des différents matériaux (tissus, os). C'est le cas avec la tomographie

(scanner à rayons X), la radiologie (rayons X), l'échographie (ultrasons), l'imagerie par résonance magnétique, les microscopies électronique et confocale.

Dans le cas où les coupes sont sériées, il s'agit de coupes parallèles de l'objet, si elles sont concentriques, alors, ce sont des coupes radiales.

Dans le cas où ces coupes doivent être de véritables coupes, elles représentent l'intersection entre le plan de coupe et l'objet. Les coupes peuvent être des projections, dans ce cas il s'agit d'une accumulation des données prises sur les droites perpendiculaires au plan de coupe (mesure de l'intensité résultant de la traversée d'un tissu par un rayon X, par exemple). Ce sont ces genres de coupes qu'on retrouve en Résonance Magnétique Nucléaire, radiographie et scanner. Les procédés de reconstruction utilisés dans ces cas-là sont décrits par R. Gordon et G.T. Herman [Gordon 74].

1.1.5. Utilisation et nécessité de l'obtention d'un modèle tridimensionnel

Du résultat souhaité dépend le processus de reconstruction utilisé (cf. figure 1.1).

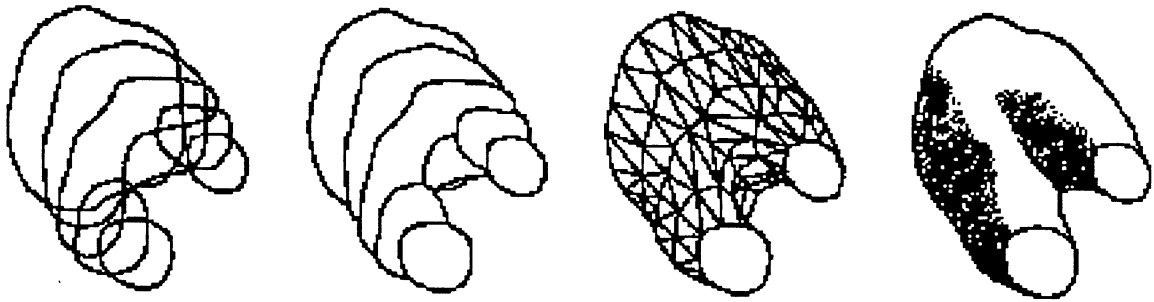


Figure 1.1. Différents résultats de reconstruction (extrait de [Russ 90]).

Le résultat de la reconstruction peut être de type surfacique, dans ce cas ce n'est que la surface de l'objet qui intéresse l'utilisateur ; il peut être de type volumique, dans ce cas c'est l'objet dans son ensemble qui est intéressant.

Dans certains cas seule une visualisation est entreprise, par contre dans d'autres une analyse et une description de l'objet sont nécessaires afin de mieux appréhender la structure tridimensionnelle de l'objet.

Le type de représentation dépend donc de l'utilisation pressentie de l'objet 3D. Si l'utilisateur souhaite une simple visualisation, la surface est suffisante ; c'est ce qui se produit généralement dans les applications de CAO où des modèles de type fil de fer sont utilisés pour visualiser l'objet. Si, par contre il veut avoir des notions de transparence il lui faudra tout l'objet, de même si l'objet doit constituer une donnée pour un processus de description, de modélisation

ou d'analyse d'objets 3D. Dans le cas général, le but à atteindre est beaucoup plus la description des structures tridimensionnelles rencontrées que le rendu visuel de celles-ci.

Les applications de la reconstruction tridimensionnelle sont nombreuses et variées : elles s'étendent de la médecine [Cook 80] à la géographie en passant par la construction navale.

En médecine, et plus généralement dans le domaine biomédical, la reconstruction touche aux domaines, de l'anatomie [Barillot 84], [Sequeira 87], de la microscopie électronique et confocale [Macias-Garza 89], de la radiologie [Pieroni 88], de la chirurgie osseuse et vasculaire mais aussi de l'aide au diagnostic, de l'analyse biologique, de la cytologie, de la biologie cellulaire, etc ...

La reconstruction 3D est particulièrement appréciée en microscopie. En effet, le microscope permet de voir des structures de très petite taille mais malheureusement il ne fournit qu'une image plane ; cependant, les tissus, cellules ou structures subcellulaires ont une architecture tridimensionnelle, le microscope ne peut donc la rendre. Le biologiste se trouve ainsi privé de certaines informations. Le microscope électronique à balayage permet de contourner ce problème mais il manque de souplesse et est limité dans certains cas (impossibilité de voir à travers les objets). Dans ce cas c'est la reconstruction 3D qui prend le relais.

La reconstruction tridimensionnelle est aussi utilisée :

- en Conception Assistée par Ordinateur : automobile, construction navale.
- en architecture et technique du bâtiment : aide à la construction, étude de la structure des matériaux (recherche de fissures, grains, bulles...).
- en géographie : cartographie (construction de modèles à partir de coupes effectuées à différentes altitudes).

1.2. Situation du problème

Afin d'éviter tout risque d'ambiguïté, il convient de préciser exactement dans quel cadre se situe le problème traité ici.

Il s'agit de : *la reconstruction tridimensionnelle à partir de coupes sériées.*

Les coupes utilisées sont des coupes obtenues dans des plans parallèles entre eux (cf. figure 1.2).

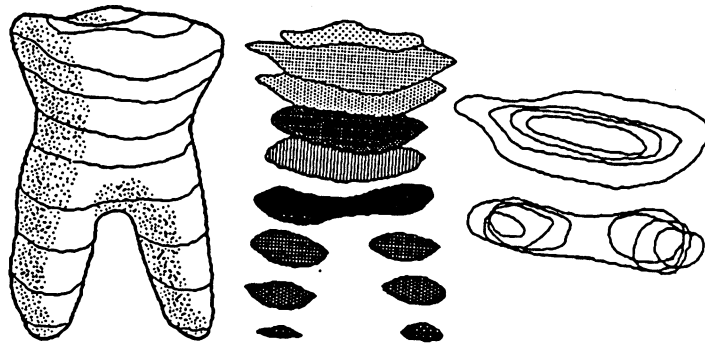


Figure 1.2. Un objet et les coupes qui en sont extraites (extrait de [Russ 90]).

Les coupes doivent être des images binaires, les coupes initialement acquises en niveaux de gris devront donc tout d'abord être segmentées avant de pouvoir être considérées comme des données de notre problème.

Ces coupes doivent aussi être significatives de l'objet à reconstruire. En effet, il faut que l'évolution de la forme dans l'inter-coupe soit directement conditionnée par les deux coupes données (une certaine continuité doit exister entre deux coupes successives) (cf. figure 1.3).

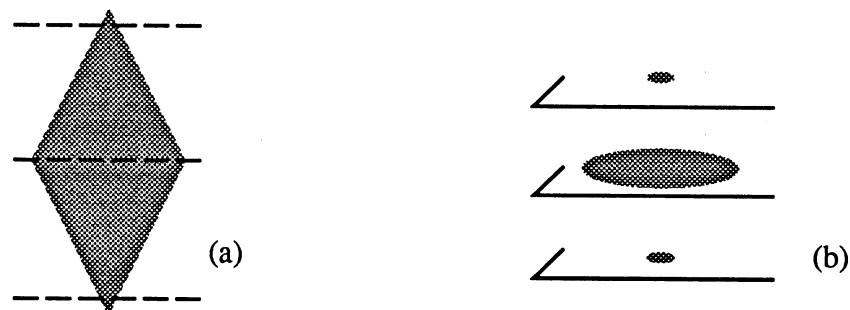


Figure 1.3. Les coupes doivent être significatives de l'objet : Pour reconstruire l'objet (a) il faut au minimum trois coupes sériées (b).

Les coupes doivent donc être assez similaires les unes des autres, pour permettre de savoir facilement quel est l'objet à reconstruire et pour qu'il ne subsiste que le moins d'ambiguïtés possibles. Les différentes coupes doivent permettre de deviner le volume qu'elles représentent tant au niveau du volume manquant qu'au niveau des associations entre les objets de deux coupes successives (cf. figure 1.4).

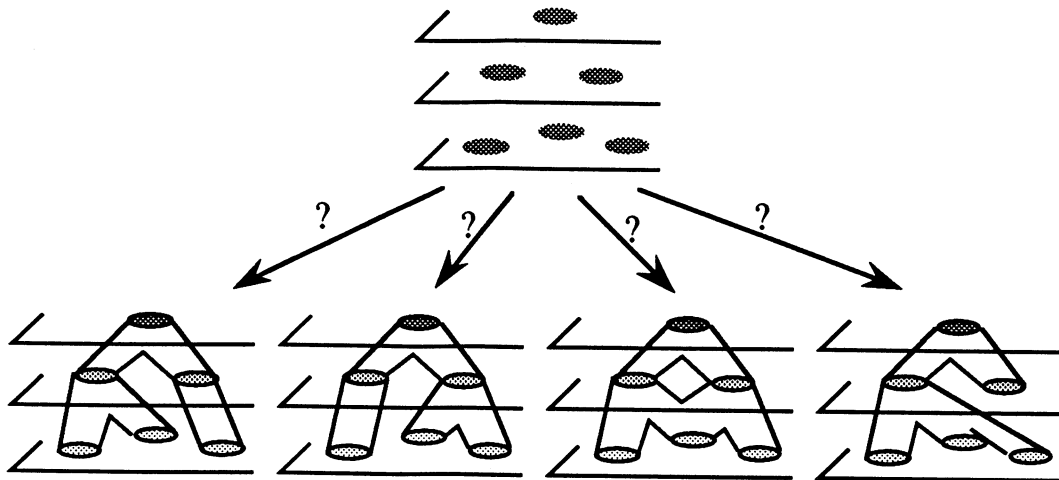


Figure 1.4. Dans certains cas des ambiguïtés sont possibles.

De plus les coupes doivent être recalées les unes par rapport aux autres c'est-à-dire liées à un même repère.

Les données sont donc des images binaires représentant des coupes parallèles significatives et recalées de l'objet à reconstruire.

Le but est d'obtenir un modèle tridimensionnel de l'objet reconstruit. Ce modèle doit contenir assez d'informations topologiques et structurelles pour définir précisément la forme 3D.

Le processus d'acquisition et de prétraitement des données, la reconstruction proprement dite, et la visualisation doivent, pour le système de reconstruction 3D idéal, réduire les interventions de l'utilisateur au strict minimum.

Un tel système doit être capable de faire une acquisition automatique des données, à l'aide d'une caméra vidéo couplée à un microscope, le tout piloté par un micro-ordinateur, par exemple. Il doit pouvoir ensuite effectuer la phase de reconstruction tridimensionnelle de la forme de manière totalement automatique en considérant que les contraintes au niveau des coupes données sont respectées (segmentées, significatives, recalées ...). Enfin le système doit permettre une visualisation de l'objet reconstruit.

De tels systèmes ont été imaginés [Veen 77] [Macias-Garza 89] mais semblent, pour l'instant, assez difficiles à mettre en œuvre. En effet, l'intervention humaine s'avère encore souvent nécessaire lors des phases de prétraitement des données telle que la segmentation, par exemple. Malgré cela, l'évolution des architectures des ordinateurs (diminution des temps de traitement, parallélisme, ...) et des techniques de l'intelligence artificielle au niveau des phases de traitement d'images (association des objets des coupes, segmentation, ...) permet à de tels systèmes de passer progressivement de l'état de projets à l'état de réalisations industrielles possibles.

1.3. Les moyens existants

1.3.1. Acquisition et prétraitement des données

Afin d'obtenir des données pour la reconstruction, il est nécessaire d'effectuer des coupes de l'objet à reconstruire. Ces coupes doivent être de véritables coupes, elles peuvent être obtenues de manière destructive ou non.

Dans le premier cas, il s'agit de découper l'objet en tranches afin d'obtenir les coupes. Cette méthode n'est évidemment pas utilisable dans tous les cas (médecine !). De plus, elle occasionne parfois des distorsions et des perturbations sur la structure même de l'objet ce qui provoque des erreurs lors de la reconstruction. Dans le cas où, par exemple, les coupes sont obtenues par ultramicrotomie celles-ci ne sont distantes que de quelques dizaines de nanomètres, ainsi le processus mécanique de la coupe provoque nécessairement une déformation de certaines structures de l'objet.

Les méthodes non destructives opèrent par focalisation sélective d'un microscope à différentes profondeurs pour obtenir les coupes ; chaque plan de focalisation donne une coupe de l'objet à une profondeur donnée. Ces méthodes sont celles de la microscopie confocale.

Les coupes ainsi obtenues sont des images en niveaux de gris.

Les coupes doivent aussi, si nécessaire, être recalées les unes par rapport aux autres. En effet, lors d'une acquisition par ultramicrotomie il est impossible de savoir si les coupes sont bien liées à un même repère, il faut donc les recalculer. Ceci est fait en effectuant des rotations et des translations sur l'une des deux coupes jusqu'à maximiser une fonction d'ajustement (cf. figure 1.5).

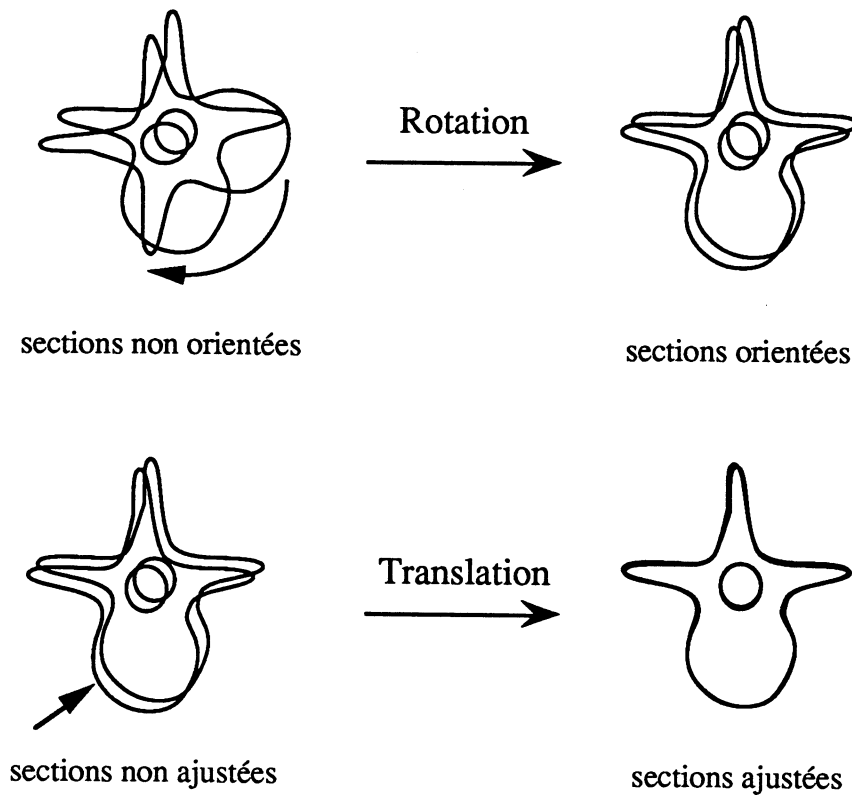


Figure 1.5. Recalage des coupes.

C'est ensuite une phase de segmentation qui est mise en œuvre. Celle-ci permet, à partir d'images en niveaux de gris d'obtenir des images binaires (cf. figure 1.6). Ce processus s'effectue par des méthodes classiques de segmentation [Rosenfeld 82] [Haralick 85] souvent complétées par l'utilisation d'outils de l'intelligence artificielle tels que les systèmes experts [Matsuyama 88] ou les systèmes multi-agents [Baujard 89].

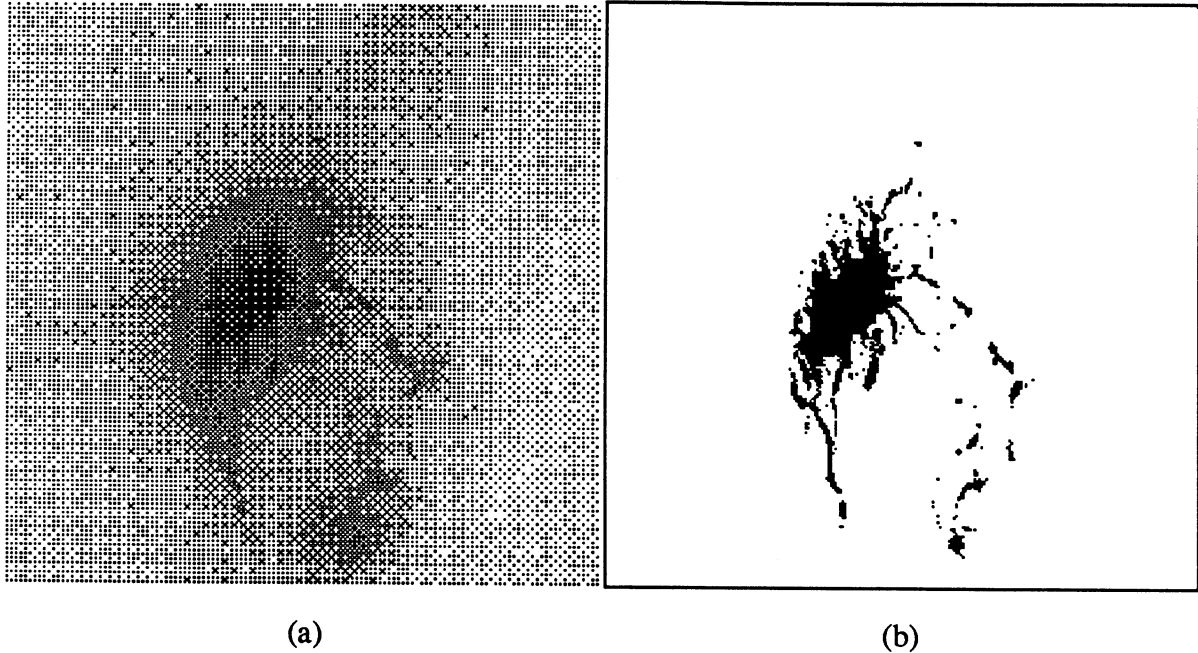


Figure 1.6. Exemple de segmentation :
L'image originale (a) et l'image segmentée (b).

Toutes ces techniques de segmentation sont très efficaces sur des images 2D, malheureusement dans notre cas il faut en plus qu'elle soit stable selon la troisième dimension. En effet, deux coupes successives doivent être segmentées en utilisant plus ou moins les mêmes paramètres sans quoi au moment de la reconstruction, surgiront des erreurs importantes sur la surface de l'objet.

Une segmentation stable selon la profondeur peut être obtenue en utilisant les techniques de détection de contours 3D [Liu 77] [Udupa 82] [Monga 90] et de segmentation tridimensionnelle [Artzy 81] [Marque 90]. En effet, en empilant les différentes coupes on obtient un volume en niveaux de gris, ensuite, en segmentant cet objet en 3D on obtient un volume binaire, en effectuant des coupes dans ce volume on obtient une "bonne" segmentation des différentes coupes initiales. Cette segmentation bidimensionnelle guidée par une segmentation tridimensionnelle est assez stable au voisinage de chacune des coupes (des problèmes pouvant apparaître sur les premières et les dernières coupes).

Si les coupes sont suffisamment nombreuses et suffisamment proches les unes des autres, on peut considérer qu'elles sont significatives. Un choix judicieux de l'axe des plans de coupe, un nombre important de coupes et l'assurance que l'objet à reconstruire est de forme "normale" sont autant de facteurs faisant décroître les risques d'ambiguïté au moment de la reconstruction.

Les coupes ainsi obtenues (véritables coupes, images binaires, recalées, significatives) sont dans la suite considérées comme les données du problème.

1.3.2. Méthodes de reconstruction tridimensionnelle

Les différentes méthodes de reconstruction tridimensionnelle à partir de coupes sériées se répartissent selon trois grandes classes :

- mise en œuvre d'une triangulation des contours des différentes coupes
- utilisation de la triangulation de Delaunay
- inférence de formes bidimensionnelles.

1.3.2.1. Méthodes par triangulation des contours

Ces méthodes, comme leur nom l'indique, n'utilisent que les contours des différentes coupes données. Pour la plupart d'entre elles, elles ne gèrent que certains points de ces mêmes contours.

Une phase d'extraction de contours et d'approximation polygonale est tout d'abord mise en œuvre.

La première solution consiste en une détermination manuelle de points de contrôle qui pour cette méthode sont considérés comme les points du contour. Cette méthode, comme toutes les méthodes utilisant l'expertise de l'utilisateur, n'est que très difficilement reproductible. Elle a malgré tout l'avantage d'être très simple et de permettre à l'utilisateur de répartir les points de manière astucieuse (plus de points dans les zones où le contour est tourmenté, moins dans celles où le contour est polygonal).

Les solutions plus automatiques utilisent les méthodes de codage du contour [Freeman 61] auxquelles sont adjointes des méthodes d'approximation polygonale [Rosenfeld 73] [Pavlidis 74] [Lux 85]. Ce sont ces méthodes qui sont généralement utilisées.

Les contours ainsi définis, la triangulation des contours est mise en place. Plusieurs méthodes de triangulation des contours ont été développées, elles sont toutes fondées sur le même principe.

Principe de base

Le principe de base de la triangulation des contours est le suivant [Shirai 87].

Soit P et Q deux polygones donnés (cf. figure 1.7) le problème se divise en deux sous-problèmes :

- détermination de la paire de sommets initiale
- pour une paire de sommets (P_i, Q_j) choisir le sommet suivant $(P_{i+1}$ ou $Q_{j+1})$ pour former une facette (triangle) de la surface de l'objet.

• détermination de la paire initiale :

Soit C le centre du polygone P , et D le centre du polygone Q ;

on note U_i le vecteur $C P_i$ et V_j le vecteur $D Q_j$ (cf. figure 1.7).

La paire de sommets (P_i, Q_j) sélectionnée est celle qui maximise la valeur du produit scalaire $U_i \cdot V_j$.

Si plusieurs choix sont possibles la paire (P_i, Q_j) la plus éloignée des centres de gravité est sélectionnée.

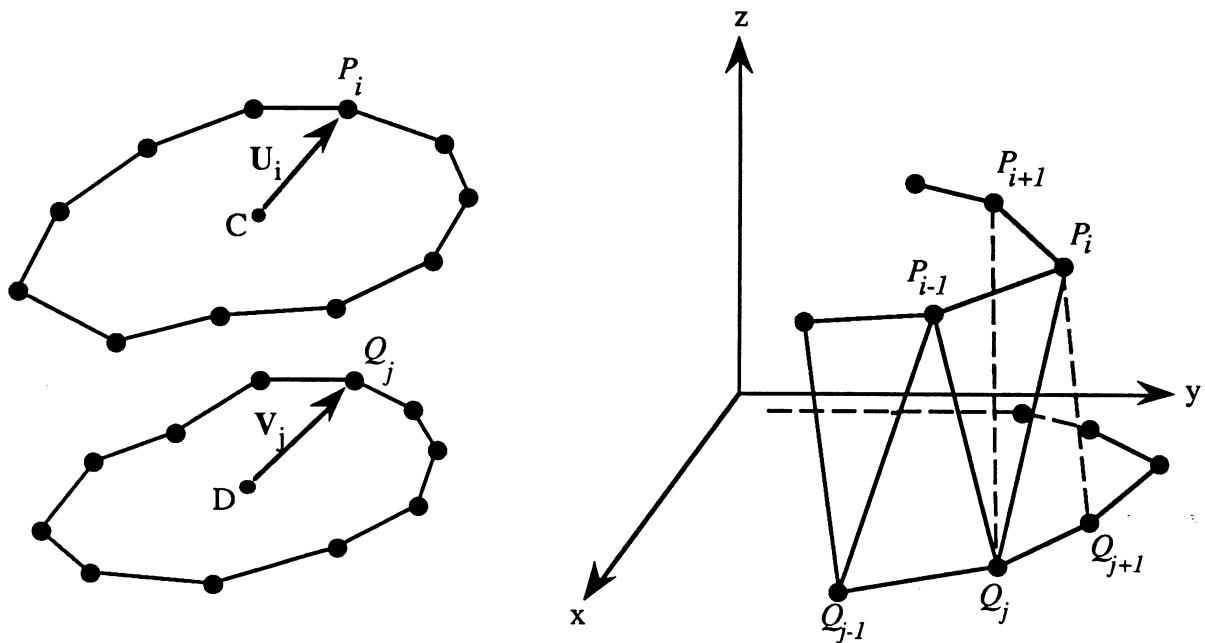


Figure 1.7. Triangulation des contours (extrait de [Shirai 87]).

• recherche du sommet suivant :

Soit la paire de sommets (P_i, Q_j) ;
 doit-on sélectionner le sommet P_{i+1} , afin de former la facette $P_i Q_j P_{i+1}$, ou bien le
 sommet Q_{j+1} , afin de former la facette $Q_j P_i Q_{j+1}$? (cf. figure 1.7).

De nombreux critères de choix peuvent être utilisés : aire des facettes, longueur des segments, direction par rapport au centre, etc ...

Une heuristique de construction fondée sur une construction intuitive est proposée. Elle consiste à minimiser l'angle entre deux facettes successives tout en évitant d'avoir des facettes de tailles trop différentes.

soit α_i l'angle entre les droites $P_i P_{i+1}$ et $Q_{j-1} Q_j$,
 soit β_i l'angle entre les droites $P_{i-1} P_i$ et $Q_j Q_{j+1}$.

L'algorithme est :

si $(P_{i+1} Q_j < P_i Q_{j+1})$ alors
 si $(\cos \alpha_i > t)$ alors sélectionner P_{i+1}
 sinon si $(\cos \beta_i > t)$ alors sélectionner Q_{j+1}
 sinon sélectionner P_{i+1}
 et inversement si $P_{i+1} Q_j \geq P_i Q_{j+1}$.
 où t est un seuil fixé par l'utilisateur.

Sur ce principe de base de nombreuses améliorations ont été proposées.

Améliorations

Ces améliorations permettent d'obtenir des résultats plus réalistes, en vérifiant, *a posteriori*, la véracité de la construction obtenue [Keppel 75].

Elles se composent de deux phases principales : la première consiste à construire les facettes, la seconde à vérifier si les facettes ainsi construites sont plausibles. Si ce n'est pas le cas, on modifie les paramètres qui permettent de construire les facettes et on recommence.

- **construction des facettes**

Soit P_0, P_1, \dots, P_{m-1} (respectivement Q_0, Q_1, \dots, Q_{n-1}) les sommets définissant le contour haut (respectivement bas). Il faut remarquer que P_0 (respectivement Q_0) suit P_{m-1} (respectivement Q_{n-1}) et que les indices de P (respectivement Q) sont donnés modulo m (respectivement n) (on notera $(x+y) [k]$ l'addition modulo k).

Chaque facette construite peut être décomposée en trois segments :

- un segment "contour" (faisant partie du contour de l'une ou de l'autre des coupes)
- deux segments "liaison" (chacune de ses extrémités faisant partie d'un contour différent).

Par exemple, pour la facette $\{P_i, Q_j, P_{i+1 [m]}\}$ on a :

- segment "contour" : $P_i P_{i+1 [m]}$
- segments "liaison" : $P_i Q_j$ et $Q_j P_{i+1 [m]}$.

Chaque facette est notée de manière à ce que les segments soient dans le sens direct. Par exemple, $\{P_i, Q_j, P_{i+1 [m]}\}$ est notée $\langle P_i, Q_j, P_{i+1 [m]} \rangle$; et $\{Q_j, P_i, Q_{j+1 [n]}\}$ est notée $\langle Q_{j+1 [n]}, P_i, Q_j \rangle$.

Pour la facette $\langle P_i, Q_j, P_{i+1 [m]} \rangle$, le segment $P_i Q_j$ est appelé *gauche* alors que le segment $Q_j P_{i+1 [m]}$ sera appelé *droit* ($P_i Q_j$ et $P_i Q_{j+1 [n]}$ sont respectivement les segments *gauche* et *droit* de la facette $\langle Q_{j+1 [n]}, P_i, Q_j \rangle$).

- **vérification de la véracité de l'ensemble des facettes construites**

Après construction, il faut vérifier que l'ensemble des facettes est "acceptable".

Pour qu'un ensemble de facettes soit "acceptable" il doit vérifier les deux conditions suivantes :

- tout segment du contour doit apparaître dans exactement *une* facette.
- si un segment apparaît comme segment *gauche* (respectivement segment *droit*) d'une facette, il doit apparaître au moins une fois comme segment *droit* (respectivement *gauche*) d'au moins une autre facette.

Ces conditions permettent de sélectionner les ensembles de facettes acceptables [Fuchs 77].

Pour évaluer ces deux conditions, la méthode la plus souvent utilisée s'appuie sur la théorie des graphes : le graphe construit est $G : \langle S, A \rangle$. S est l'ensemble des sommets et A l'ensemble des arcs du graphe.

Les sommets correspondent à l'ensemble des segments possibles entre les points P_0, P_1, \dots, P_{m-1} et les points Q_0, Q_1, \dots, Q_{n-1} ; les arcs correspondent à l'ensemble de toutes les facettes

définies par l'algorithme. Chaque arc part d'un sommet qui correspond au segment *gauche* de la facette et va sur le sommet qui correspond au segment "droit".

Ainsi, $S = \{ S_{ij} / i = 0, 1, \dots, m-1 ; j = 0, 1, \dots, n-1 \}$ dans lequel S_{ij} correspond au segment $P_i Q_j$ et $A = \{ \langle S_{kl}, S_{tu} \rangle$ avec soit $t = k$ et $u = l+1 [n]$ soit $t = k+1 [m]$ et $u = l \}$ et $\langle S_{kl}, S_{tu} \rangle$ correspond à la facette de segment *gauche* $P_k Q_l$ et de segment *droit* $P_t Q_u$.

Le graphe ainsi construit est un graphe toroïde (cf. figure 1.8).

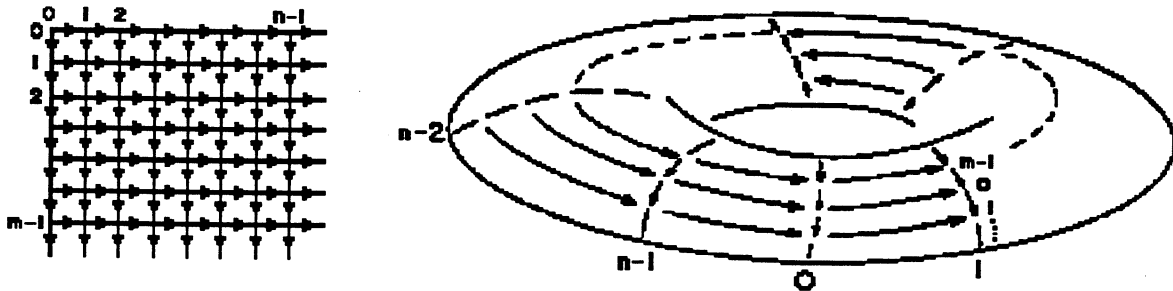


Figure 1.8. Graphe toroïde (extrait de [Fuchs 77]).

A partir de cette base théorique de nombreuses et différentes techniques s'affrontent [Keppel 75] [Fuchs 77] [Christiansen 78] [Ganapathy 82] [Pieroni 88]. Toutes sont fondées sur la surface de chaque facette et sur la surface totale de ces facettes. Le nombre de facettes acceptables est défini par une fonction récursive [Ganapathy 82] :

$$A[m,n] = A[m-1,n] + A[m,n-1] \text{ avec } A[m,1] = 1 \text{ pour tout } m \text{ et } A[1,n] = 1 \text{ pour tout } n.$$

Par récurrence on obtient le résultat sous la forme :

$$A[m,n] = \frac{[(m+1) + (n+1)]!}{(m-1)! (n-1)!}$$

Cette fonction donne très rapidement de très grandes valeurs.

Les différents auteurs ont donc proposé des conditions pour déterminer la surface la plus acceptable.

Soit $\phi : A \rightarrow \mathbb{R}$ (A ensemble des facettes élémentaires) définissant le "poids" de chaque facette (c'est en fait sa surface).

Etant donné que la surface construite doit être composée d'exactly $m+n$ facettes,

on définit $\phi = \sum_{k=1}^{m+n} \phi(a_k)$.

C'est sur la valeur de ϕ que les avis divergent :

- certains prétendent que la surface la plus acceptable minimise la valeur de ϕ

[Fuchs 77]

- d'autres qu'elle rend ϕ maximale [Keppel 75]

- d'autres, encore, disent qu'il ne faut ni maximiser, ni minimiser ϕ mais arriver à en

prévoir les valeurs sans que ceci ne soit trop prohibitif en temps de calcul ; ils proposent, donc, des méthodes heuristiques [Christiansen 78] [Ganapathy 82].

Les travaux plus récents [Pieroni 88] permettent de diminuer le nombre de graphes à traiter (en éliminant des graphes "inacceptables") et permettent ainsi de réduire les temps de calcul et, par la même, d'obtenir de meilleurs résultats.

On se rend compte que le problème de construction des facettes n'est pas un problème des plus simples ! Les différentes méthodes s'affrontent encore et dans la pratique ne donnent que peu d'améliorations par rapport à la méthode "classique" : temps de calcul trop importants par rapport aux améliorations apportées. En pratique, c'est encore la vérification "humaine" qui marche le mieux (triangulation manuelle dans les zones difficiles). De plus, ces méthodes échouent encore dans de nombreux cas [Christiansen 78] [Kaneda 88].

Les méthodes qui semblent les plus efficaces sont celles qui travaillent en s'appuyant localement, soit sur les surfaces des facettes (choix possible entre deux facettes), soit sur les diagonales (plus petite diagonale), soit sur des rapports entre la surface et le périmètre des facettes [Christiansen 78] [Pieroni 88].

Toutes ces méthodes de triangulation des contours sont largement utilisées dans la pratique. Malheureusement elles échouent souvent lorsqu'il apparaît, entre deux coupes successives, de trop grandes dissimilitudes. C'est le cas pour toutes les ruptures topologiques (apparition de branches ou de trous), quand le volume est presque parallèle au plan de coupe ou lorsqu'il y a une évolution trop rapide de la forme entre deux coupes successives (cf. figure 1.9). Dans ces cas un traitement *ad hoc* des problèmes doit être envisagé [Kaneda 88].

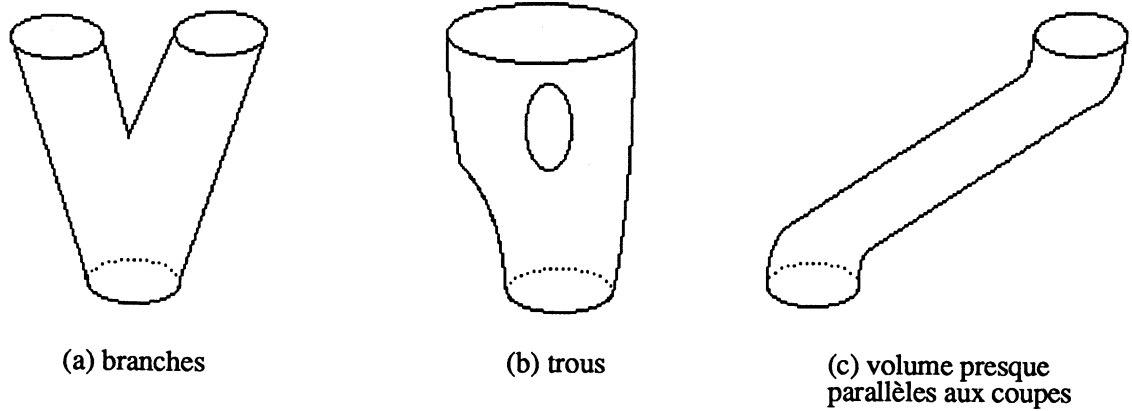


Figure 1.9. Trois exemples d'objets posant des problèmes.

Un traitement différent est effectué pour chacun de ces quatre cas :

- **apparition de branches :**

Un prétraitement des coupes doit être effectué afin de créer une structure arborescente déterminant à quel sous-objet chaque contour appartient (cf. figure 1.10).

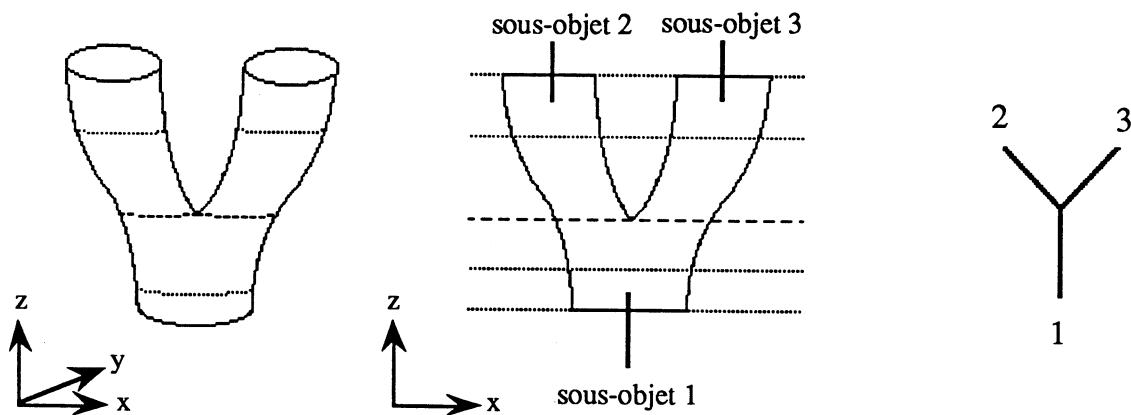


Figure 1.10. Exemple de sous-objets et connexions associées (extrait de [Kaneda 88]).

- **les trous :**

Seule une détection est proposée, aucun traitement, à ce jour, n'est encore effectif.

• **les objets presque parallèles au plan de coupe :**

Dans ce cas le calcul doit être effectué en trois étapes :

- recentrage des contours : chaque contour est approché par sa boîte englobante, le vecteur séparant les centres des deux boîtes est calculé, l'un des deux contours est déplacé selon ce vecteur.
- les facettes sont calculées (méthodes "classiques" de triangulation)
- le contour, qui avait été décalé, est replacé à sa position initiale, ceci induisant une déformation des facettes.

• **les changements rapides de forme entre deux coupes successives :**

C'est dans ce cas-là que la méthode est la plus nouvelle et la plus intéressante.

La détection se fait en remarquant que, lorsqu'on a un rapide changement entre deux coupes successives, la longueur des segments reliant les deux coupes devient vite grande par rapport à la distance intercoupe (cf. figure 1.11) ; dès que $l_{ij}^2 > k(h^2 + s_j^2)$ (k paramètre) on considère que l'on est dans ce cas.

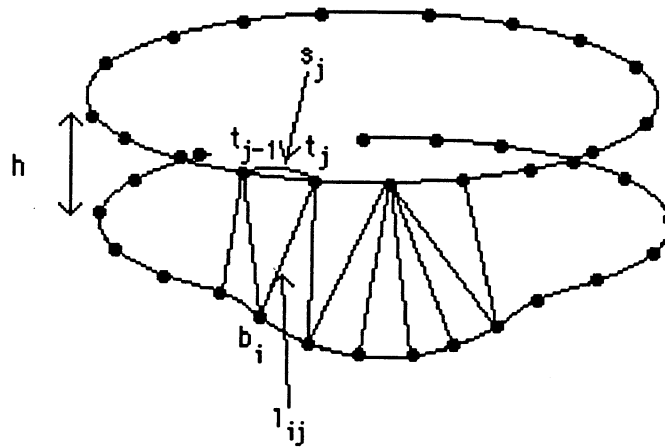


Figure 1.11. Exemple de rapide changement de forme
(Extrait de [Kaneda 88]).

Dès qu'une zone de rapide changement de forme est détectée entre deux coupes successives, une coupe intermédiaire locale est intercalée (pour des raisons de calculs, seule une portion de contour est inférée); puis un nouveau calcul de triangulation est lancé (cf. figure 1.12).

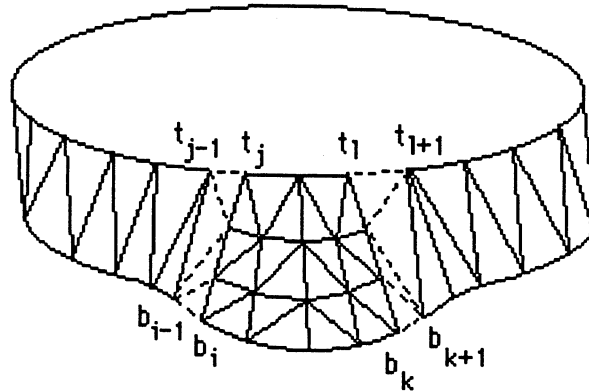


Figure 1.12. Reconstruction utilisant l'inférence de portions de contour (extrait de [Kaneda 88]).

Cette méthode permet d'obtenir de bien meilleurs résultats que les méthodes "classiques" mais ne propose qu'un traitement local des cas particuliers.

D'autres méthodes utilisant la triangulation, mais cette fois une triangulation très particulière, ont été développées ; il s'agit de la triangulation de Delaunay.

1.3.2.2. Méthode utilisant la triangulation de Delaunay

Cette méthode utilise un mode de triangulation selon la méthode de Delaunay [Boissonnat 84] [Boissonnat 85] [Boissonnat 88]. La triangulation de Delaunay [Preparata 85] (cf. figure 1.13) peut être définie comme la seule triangulation à cercles vides, c'est-à-dire que trois points forment un triangle de Delaunay si et seulement si le cercle (cercle de Delaunay) passant par ces trois points ne contient aucun autre point.

La triangulation de Delaunay est parfois simplement définie par la définition de son dual : le graphe de Voronoï.

Le graphe de Voronoï est l'ensemble des points du plan équidistants d'au moins deux "germes" (points donnés).

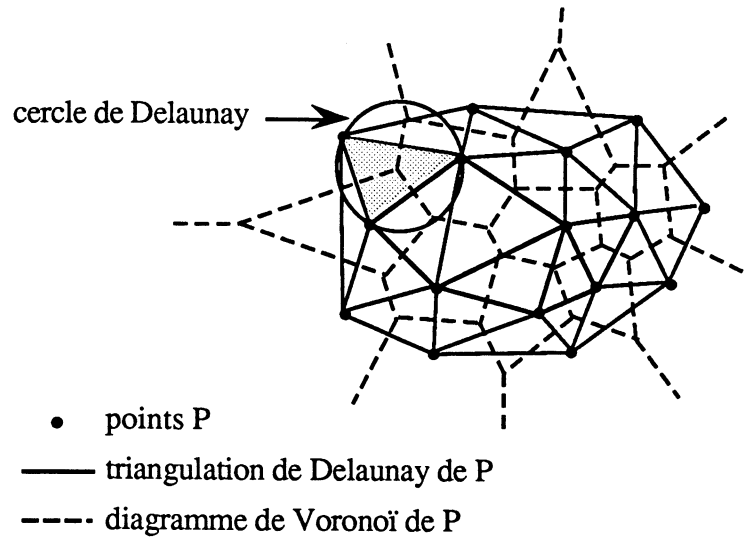


Figure 1.13. Diagramme de Voronoï et triangulation de Delaunay.

Cette méthode utilise les contours. Le prétraitement utilisé précédemment (§ 1.3.2.1) doit donc être mis en œuvre afin d'obtenir une approximation polygonale des contours de chacune des coupes.

Ensuite, il faut ajouter des points à ce contour afin que les segments de celui-ci appartiennent tous à la triangulation de Delaunay des points du contour (cf. figure 1.14).

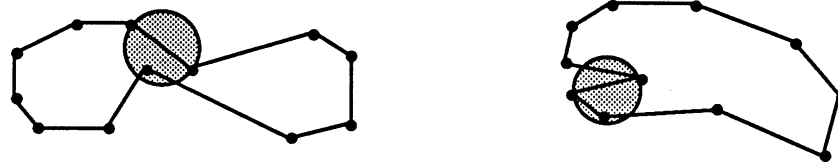


Figure 1.14. Exemples de polygones non contenus dans la triangulation de Delaunay de leurs sommets.

Ceci est fait en utilisant l'algorithme proposé par J.D. Boissonnat [Boissonnat 88] :

Initialisation

C := liste des arêtes du contour

M := liste des sommets du contour

Calculer DT triangulation de Delaunay de M

A := liste des arêtes de C qui ne sont pas des arêtes de DT

```

Boucle
  Tant que A est non vide faire
    début
      M := M ∪ les milieux des arêtes contenues dans A ;
      Ces milieux décomposent chaque arête de A en deux nouvelles arêtes qui
      sont ajoutées dans la liste B ;
      C := C - A ∪ B
    fin ;

```

Cet ajout de points de contours est effectué pour chacun des contours. Ensuite la triangulation proprement dite est effectuée. Celle-ci est une triangulation de Delaunay en 3D, mais il s'agit d'un cas particulier. En effet, les points donnés se situent sur deux plans parallèles ce qui simplifie grandement l'algorithme.

L'algorithme [Boissonnat 88] permet de calculer la triangulation tridimensionnelle de Delaunay DT à partir de deux triangulations bidimensionnelles de Delaunay DT_1 et DT_2 .

L'enveloppe convexe de cette triangulation tridimensionnelle peut être considérée comme une représentation par facettes de l'objet construit à partir de deux coupes successives.

Cette méthode est assez efficace, elle apporte une solution, sous certaines conditions, dans les cas de branchement.

D'autres méthodes que celles faisant appel à la triangulation ont été développées, il s'agit des méthodes d'inférence de formes 2D.

1.3.2.3. Méthodes par inférence de formes 2D

Au cours de certaines applications de la reconstruction tridimensionnelle à partir de coupes sériées, on s'aperçoit que si les coupes sont suffisamment nombreuses et assez proches les unes des autres un simple empilement des différentes coupes suffit pour obtenir un résultat satisfaisant. (cf. figure 1.15) [Calverley 88]

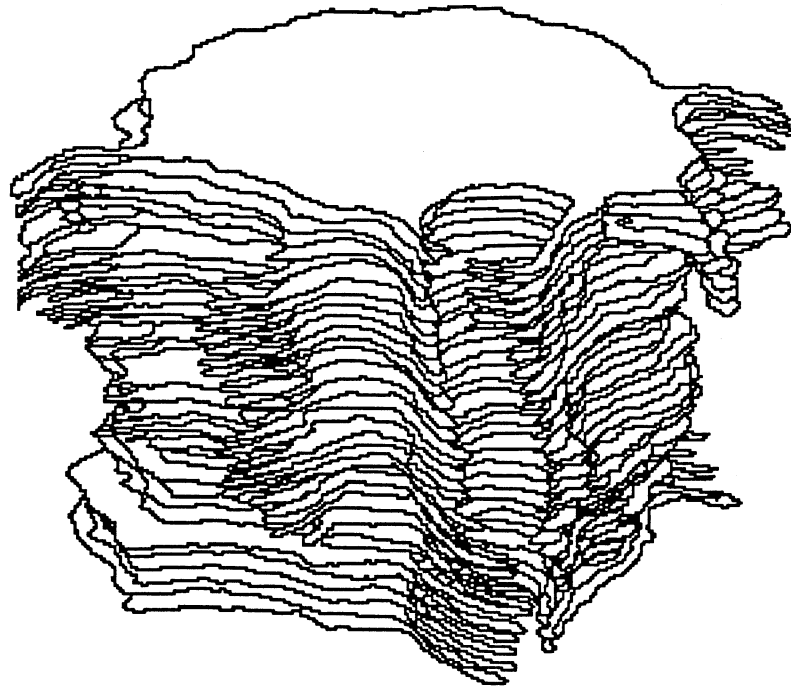


Figure 1.15. Exemple d'objet construit par simple empilement.

Malheureusement, lors de l'acquisition il n'est pas toujours possible d'obtenir de telles conditions. Dans de nombreux cas, la résolution des systèmes optiques d'acquisition n'est pas la même perpendiculairement aux plans de coupe que selon le plan de coupe. Ce rapport de résolution étant souvent de l'ordre de $1/4$ ou $1/3$, il est nécessaire de pouvoir calculer des coupes "intermédiaires" entre deux coupes successives afin d'avoir une résolution sensiblement équivalente selon les trois dimensions de l'espace.

C'est à partir de cette idée que les méthodes d'inférence de formes 2D ont été développées. Bien sûr il serait possible d'effectuer une triangulation entre deux coupes. Malheureusement ce processus n'est pas toujours très simple à mettre en œuvre, comme nous avons pu le voir précédemment, et, de plus, il ne fournit pas de modèle de l'objet.

Les méthodes d'inférence de formes nécessitent l'utilisation, tant en 2D qu'en 3D, de modèles de représentation de formes. Ce sont ceux-ci que nous allons aborder dans les chapitres 2 et 3 avant de présenter, chapitres 4 et suivants, les méthodes d'inférence de formes proprement dites (méthodes de "contours élastiques" et d'inférence structurelle).

Avant cela, il convient de préciser les différentes possibilités de visualisation du résultat.

1.3.3. Visualisation du résultat

Le résultat obtenu est un volume. Celui-ci est décrit soit par sa surface (méthode de triangulation), soit par le volume complet (méthode par inférence de formes). De cette description vont dépendre les méthodes de visualisation.

Le problème consiste, dans tous les cas, à visualiser un objet tridimensionnel sur un écran plan (les méthodes de type holographique étant encore à l'état d'études). La résolution de ce problème passe par l'utilisation de projections. Avant de pouvoir projeter l'objet sur l'écran il est nécessaire de l'orienter dans l'espace et de définir un point de vue. Ces mouvements dans l'espace sont effectués par des compositions de translations, rotations et affinités opérant sur les coordonnées homogènes de l'objet.

Ensuite différentes méthodes de rendu doivent être mises en œuvre. Les principales d'entre elles sont les lissages, l'élimination des parties cachées, les couleurs, transparences, ombrages et éclairages. Pour cette mise en œuvre le type de volume utilisé est important. En effet, on n'utilisera pas les mêmes méthodes d'élimination de parties cachées si l'on connaît le volume à l'aide de sa surface décrite sous forme d'ensemble de facettes que si l'objet est décrit par tous les points de l'espace le constituant (représentation par voxels).

Toutes ces techniques étant très classiques et ayant fait l'objet de nombreuses thèses et ouvrages [Martinez 82] [Rogers 85] [Martinez 86] [Péroche 88], elles ne seront pas développées ici. Elles sont en fait un post-traitement de notre problème.

Par contre il est important de préciser ici que, quel que soit le mode de visualisation choisi et quelles que soient les techniques mises en œuvre, la visualisation, sur un écran, de données tridimensionnelles, implique une perte d'informations non négligeable ; une description plus formelle bien que moins agréable, est beaucoup plus puissante au niveau descriptif.

C'est à ces aspects de représentation et de modèles de descriptions dans les espaces 2D et 3D que sont consacrés les deux chapitres suivants.

Chapitre 2

Modèles en analyse de formes : aspect bidimensionnel

Ce chapitre est consacré au rappel de notions indispensables au travail dans le plan discret \mathbb{Z}^2 , à la précision de quelques notations ainsi qu'à la description des différents modèles de représentation d'images.

Nous précisons les significations attribuées à certains termes, afin de limiter les ambiguïtés sémantiques.

2.1. Définitions et notions de base

2.1.1. Images discrètes binaires

L'analyse d'images par ordinateur nécessite l'utilisation de données à support fini. La représentation de base est l'image discrète, elle est le résultat d'un échantillonnage spatial et d'une quantification d'amplitude à partir de l'image analogique. Le support de ces images est associé à un maillage (issu d'un pavage) précisant l'arrangement des points entre eux. Différents maillages réguliers sont possibles (cf. figure 2.1).

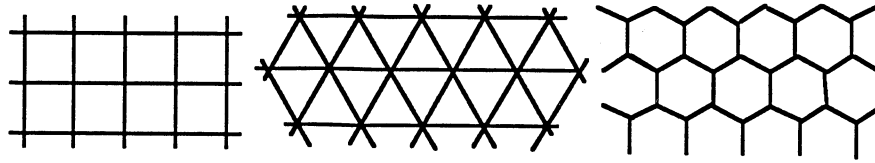


Figure 2.1. Maillages carré, triangulaire et hexagonal.

En raison de sa plus grande facilité d'obtention, de la simplicité de son calcul et de sa proximité de la réalité physique des capteurs d'images, le maillage carré est le plus fréquemment utilisé. C'est celui qui est employé dans la suite.

Chaque point du maillage définit un point de l'image, appelé pixel, auquel est associé une valeur (niveaux de gris, couleur, ...).

L'image discrète peut donc être définie par une fonction f (appelée fonction caractéristique) :

$$\begin{aligned} f: \quad \mathbb{Z}^2 &\rightarrow \mathbb{I} \\ (i, j) &\rightarrow f(i, j) \end{aligned}$$

Les données devant être finies, le domaine de la fonction est par conséquent réduit au produit cartésien de deux intervalles : $[1 \dots N] \times [1 \dots M]$.

L'image est alors dite de dimension $N \times M$.

Chaque image peut donc être codée à l'aide d'une matrice $N \times M$ à coefficients dans \mathbb{I} (cf. figure 2.2).

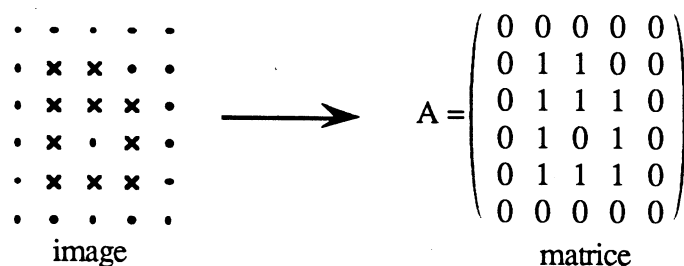


Figure 2.2. Codage matriciel d'une image.

Dans le cas où l'ensemble \mathbb{I} comprend 256 valeurs, on parle d'images à 256 niveaux.

Dans notre cas, l'ensemble \mathbb{I} est réduit à deux valeurs (0 et 1), on parle alors d'images binaires.

Les points de valeur 0 représentent le fond, les points de valeur 1 représentent la forme (les objets de l'image).

Afin de pouvoir traiter les objets, il est nécessaire de définir des distances sur \mathbb{Z}^2 .

2.1.2. Distances dans \mathbb{Z}^2

Plusieurs distances ont été développées sur \mathbb{Z}^2 . Seules celles utilisant le maillage carré sont précisées ici.

Quatre métriques sont généralement utilisées dans \mathbb{Z}^2 , à la définition de chacune d'elles correspondent différents modes de raisonnement [Chassery 91].

On définit aussi l'image de distance comme étant l'image dans laquelle chaque pixel prend pour valeur la distance minimale qui le sépare de l'objet

2.1.2.1. Distance euclidienne

C'est la distance euclidienne classique de l'espace continu \mathbb{R}^2 ramenée au plan discret \mathbb{Z}^2 . Elle est définie par :

$$\begin{aligned} d_E (P , Q) &= d_E ((x_P, y_P) , (x_Q, y_Q)) \\ &= \sqrt{(x_Q - x_P)^2 + (y_Q - y_P)^2} . \end{aligned}$$

Malheureusement les valeurs de $d_E (P , Q)$ ne sont pas entières ; or il est préférable de faire des calculs entiers (contraintes de stockage, problèmes de précision et d'erreurs de calcul avec les réels) ; et leur approximation entière n'est pas une fonction de distance. De plus, le calcul de l'image de distance nécessite le calcul et le stockage de deux images (distances selon les abscisses et les ordonnées), il s'agit donc d'un procédé de calcul coûteux.

D'où la nécessité de définir d'autres distances entièrement discrètes.

2.1.2.2. Distance de Manhattan

C'est la distance la plus simple, elle correspond à des déplacements horizontaux ou verticaux sur le maillage carré. Elle est souvent notée d_4 car chaque pixel est à distance 1 de quatre autres. Sa définition est :

$$d_4 (P , Q) = |x_P - x_Q| + |y_P - y_Q|.$$

De cette définition découle la notion de boule discrète selon d_4 :

$$B_4 (P , 1) = \{ Q \in \mathbb{Z}^2 : d_4 (P , Q) = 1 \}$$

Cette boule a une forme de diamant, fort différente du disque défini par d_E (cf. figure 2.3).

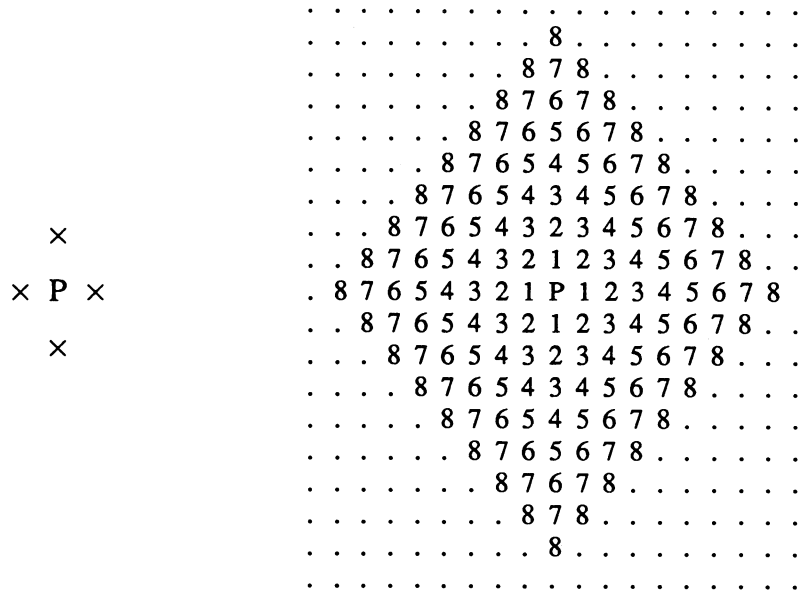


Figure 2.3. $B_4 (P , 1)$ et propagation de la distance d_4 .

On démontre facilement que :

$$\forall P, Q \in \mathbb{Z}^2, d_4 (P , Q) \geq d_E (P , Q) .$$

Les erreurs relatives par rapport à la distance euclidienne varient entre 0% et 41,4%.

Pour la distance de Manhattan, le calcul de l'image de distance se fait très facilement en deux passages sur l'image à l'aide de masques [Rosenfeld 68].

2.1.2.3. Distance de l'échiquier

Elle correspond à des déplacements horizontaux, verticaux ou diagonaux sur le maillage. Elle est notée d_8 car chaque pixel à huit voisins à distance 1.

Elle est définie par :

$$d_8 (P , Q) = \max (|(x_P - x_Q)| , |(y_P - y_Q)|) .$$

Cette distance définit des boules carrées qui elles aussi provoquent des approximations importantes par rapport à la distance euclidienne (cf. figure 2.4).

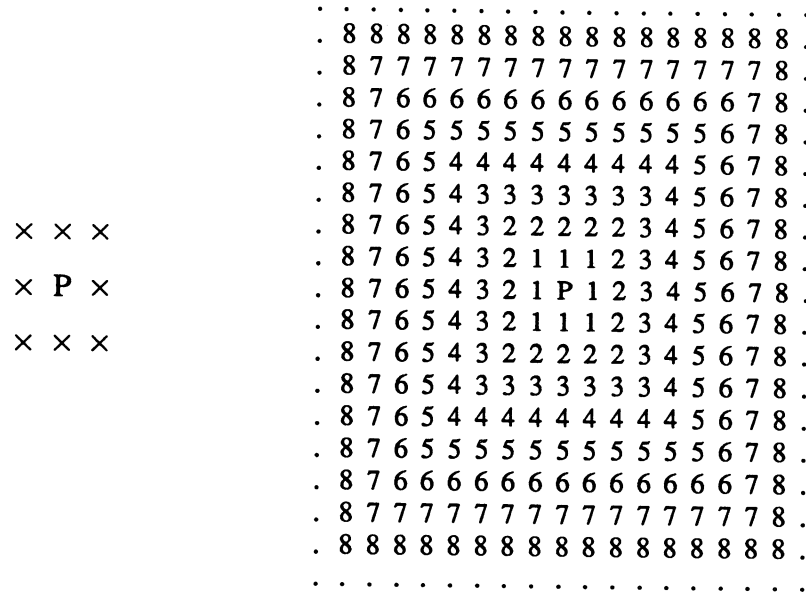


Figure 2.4. $B_8 (P , 1)$ et propagation de la distance d_8 .

Ici, par contre, on démontre facilement que :

$$\forall P, Q \in \mathbb{Z}^2, d_8 (P , Q) \leq d_E (P , Q)$$

Les erreurs relatives par rapport à la distance euclidienne sont inférieures à 29,3%.

Là encore le calcul de l'image de distance se fait très facilement en deux passages sur l'image à l'aide de masques.

En remarquant que $d_4 \geq d_E$ et que $d_8 \leq d_E$ il est possible de définir une distance "octogonale" [Rosenfeld 68] en utilisant alternativement d_4 et d_8 . Cette distance donne une meilleure approximation de la distance euclidienne que les distances de Manhattan et de l'échiquier.

2.1.2.4. Distances du chanfrein

Ces distances tentent de faire une approximation un peu plus précise de la distance euclidienne. Elles ont été construites en essayant tout d'abord de minimiser l'erreur commise sur le support image par rapport à la distance euclidienne et ensuite en ajoutant une contrainte d'intégrité aux différents coefficients calculés. Ainsi la distance du chanfrein 3-4 est construite en pondérant par une valeur de 3 les déplacements horizontaux et verticaux (cf. figure 2.5) et par une valeur de 4 les déplacements diagonaux, on obtient une meilleure approximation de la distance euclidienne (masque 3 x 3) [Borgefors 86]. Les boules sont ainsi un peu plus rondes (cf. figure 2.5).



Figure 2.5. Masque du chanfrein 3-4 et une boule associée.

L'algorithme séquentiel de calcul de l'image de distance pour la distance du chanfrein 3-4 nécessite deux passages (aller et retour) sur l'image à l'aide de masques.

Les deux masques utilisés sont deux masques 3 x 2.

Il s'agit de $\begin{pmatrix} 4 & 3 & 4 \\ 3 & 0 & \end{pmatrix}$ et $\begin{pmatrix} 0 & 3 \\ 4 & 3 & 4 \end{pmatrix}$. Ils sont appelés m_{aller} et m_{retour} .

Ces masques sont appliqués sur la représentation matricielle A de l'image (de taille $N \times M$), où les points de l'objet sont initialisés à ∞ et les autres à 0, à l'aide de l'algorithme suivant :

Passage aller :

pour $i := 1$ à N faire

pour $j := 1$ à M faire

$$A(i,j) = \text{minimum}_{(h,k) \in \{(-1,-1), (-1,0), (-1,1), (0,-1)\}} (A(i+h, j+k) + m_{\text{aller}}(h,k));$$

Passage retour :

pour $i := N$ à 1 faire

pour $j := M$ à 1 faire

$$A(i,j) = \text{minimum}_{(h,k) \in \{(0,1), (1,-1), (1,0), (1,1)\}} (A(i+h, j+k) + m_{\text{retour}}(h,k));$$

En passant d'un masque 3 x 3 à un masque 5 x 5 (chanfrein 5-7-11), 7 x 7 (chanfrein 14-20-31-44) ou plus encore, on peut se permettre d'augmenter la valeur des coefficients (cf. figure 2.6), on obtient donc des approximations de la distance euclidienne de plus en plus précises [Borgefors 86] [Thiel 91a].

						1300
					1015	1184
			732			1093
	287	454	642	837	1035	
0	203					

Figure 2.6. Premier octant du masque du chanfrein 11 x 11.

Les boules, ainsi définies, sont de plus en plus proches des disques de la distance euclidienne (cf. figure 2.7). Malheureusement, plus la taille du masque augmente plus les coefficients sont grands et plus les temps de calcul et le volume des données à stocker deviennent importants, c'est pourquoi, souvent, on se contente d'utiliser les distances discrètes d_4 et d_8 .

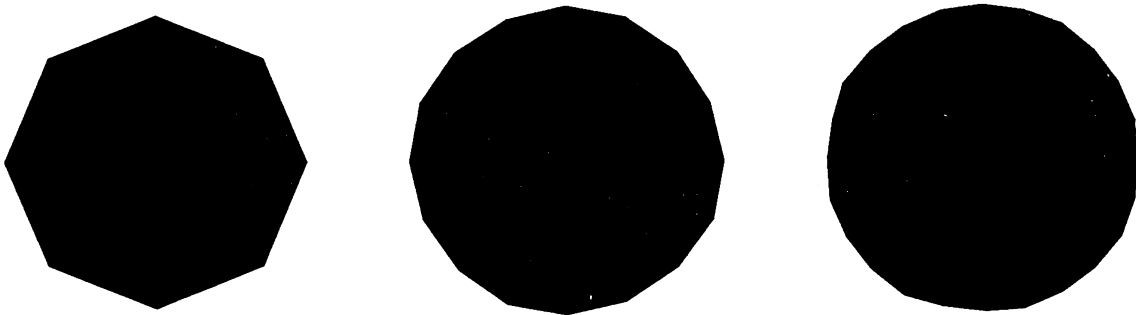


Figure 2.7. Exemples des boules pour les distances du chanfrein. (a) chanfrein 3-4, (b) chanfrein 5-7-11, (c) chanfrein 14-20-31-44

2.1.3. Voisinage et connexité

Les notions de voisinage et de connexité viennent des métriques d_4 et d_8 .

Un pixel Q est dit *voisin* du pixel P si et seulement si $d(P, Q) = 1$.

On obtient ainsi deux notions de voisinage différentes pour d_4 et pour d_8 correspondant pour chacune d'elles à la boule unité centrée en P .

Dès lors, le concept topologique de points voisins doit être compris en terme de points consécutifs potentiels d'un point donné.

Pour les définitions suivantes les notions de voisinage s'entendent par rapport à une métrique donnée.

On appelle *chemin connexe* une suite de points $P_0 \dots P_n$ telle que :
 $\forall i \in [1 .. n] P_i$ est voisin de P_{i-1} (cf. figure 2.8).



Figure 2.8. Chemin connexe entre P_0 et P_n selon d_4 (a) et selon d_8 (b).

On appelle *courbe* une suite de points $P_0 \dots P_n$ telle que chacun des points possède exactement deux voisins (cf. figure 2.9).

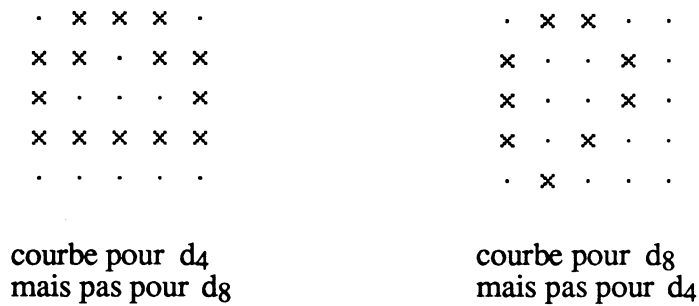


Figure 2.9. Courbes.

On appelle *composante connexe* tout sous-ensemble S de \mathbb{Z}^2 tel que :
 $\forall (A, B) \in S \times S$, il existe un chemin de A à B entièrement inclus dans S (cf. figure 2.10).

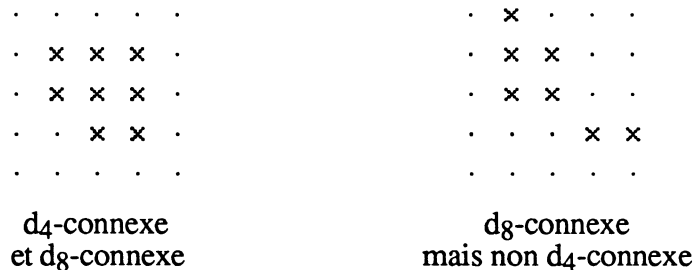


Figure 2.10. Composantes connexes.

On appelle *objet* une composante connexe de pixels de valeur 1 (4- ou 8-connexe selon la métrique utilisée).

Ainsi, une image binaire, initialement partitionnée en classe objets (points de valeur 1) et classe fond (points de valeur 0), est composée :

- d'objets : les composantes connexes de points à 1
- du fond : la composante connexe de points à 0 qui contient le bord de l'image
- des trous éventuels dans les objets : les autres composantes connexes de la classe fond

(cf. figure 2.11).

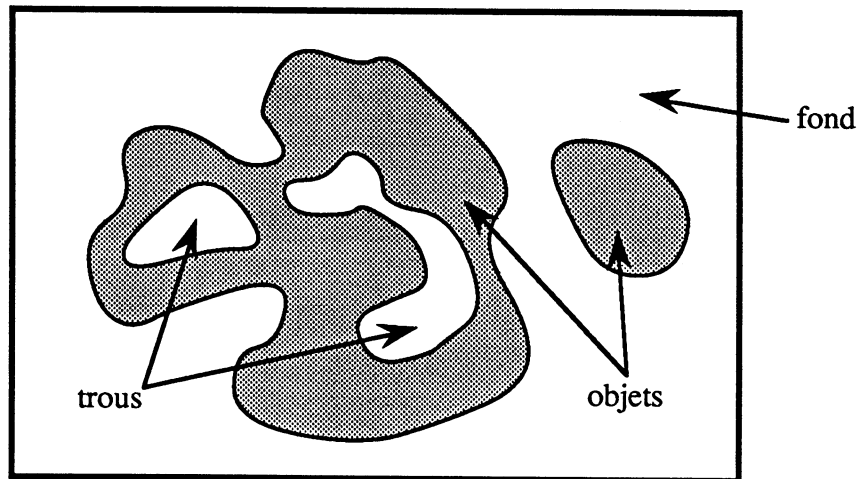


Figure 2.11. Une image est composée d'objets, du fond et de trous.

Il convient, ici, de présenter le théorème de Jordan dans sa version utilisable dans le plan discret :

Le complémentaire de toute courbe discrète 4-connexe (respectivement 8-connexe) est formé de deux composantes 8-connexes (respectivement 4-connexes). L'une d'entre elles est l'intérieur de la courbe discrète, l'autre est l'extérieur (cf. figure 2.12).

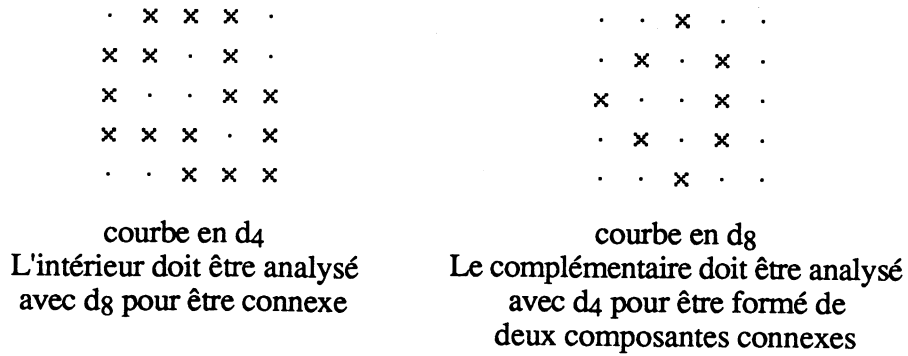


Figure 2.12. Illustration du théorème de Jordan.

Ce théorème est très important pour la compréhension des différentes notions de connexité. Il permet entre autre de considérer les distances d_4 et d_8 comme duales l'une de l'autre.

On appelle point *bord* d'un objet tout point P de valeur 1 (objet) dont l'un des 4 voisins a pour valeur 0 (dans le fond ou dans un trou).

On appelle *contour* d'un objet S l'ensemble des points de S qui sont à une distance 1 du complémentaire de S (selon la distance qui caractérise S) (cf. figure 2.13).

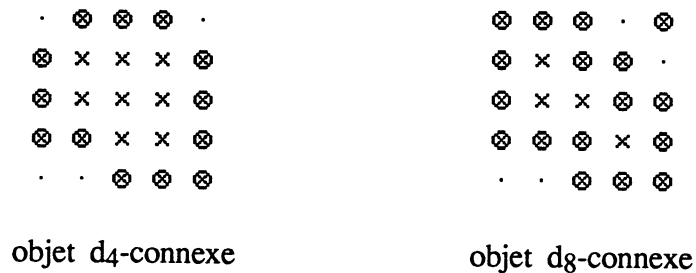


Figure 2.13. Contours.

On peut distinguer contours internes et contours externes (cf. figure 2.14).

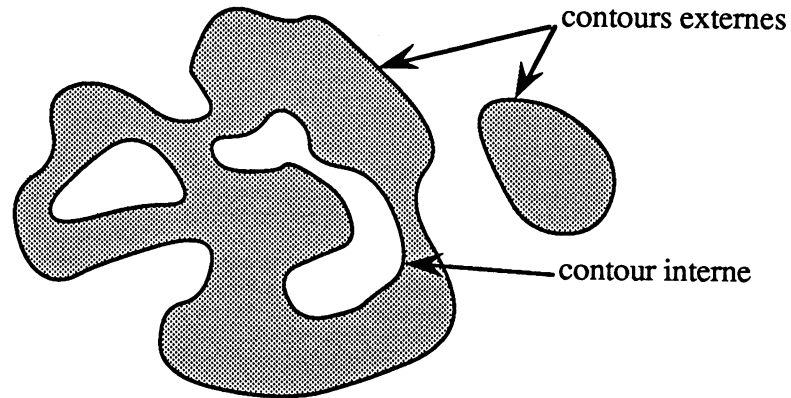


Figure 2.14. Différents contours.

On appelle *type de connexité* d'une image la valeur ε définie par :

$$\varepsilon = K - H$$

où K représente le nombre de composantes de l'image et H le nombre de trous.

Ce type de connexité est un invariant topologique [Duda 73].

Un calcul plus local permet de définir le nombre de connexité [Yokoi 75].

On appelle *nombre de connexité* N_4 (pour la 4-connexité) et N_8 (pour la 8-connexité) les valeurs définies par :

$$N_4 = \sum_{k=0}^7 (v_k - v_k \cdot v_{k+1} \cdot v_{k+2})$$

$$N_8 = \sum_{k=0}^7 (\bar{v}_k - \bar{v}_k \cdot \bar{v}_{k+1} \cdot \bar{v}_{k+2})$$

où v_k représente la valeur du point de l'image dans la direction de Freeman k (cf Figure 2.15).

Cette notion de nombre de connexité est utilisée lors de calcul de modèles de représentation par squelettes, il permet de classifier les points dans leur contexte (point isolé, intérieur, de croisement, de branchement, ...).

2.2. Modèles de représentation

Divers modèles de représentation de formes 2D existent, ils sont principalement fondés sur deux approches différentes de la forme : soit une approche de type contour, soit une approche de type région.

Les approches de type contour sont, généralement, soit des codages du contour de l'objet, soit une approximation polygonale de ce même contour.

Les approches de type région utilisent l'ensemble de la forme (et pas seulement le contour).

2.2.1. Contour et approximations polygonales

Les contours sont stockés à l'aide d'un codage séquentiel [Freeman 61]. On mémorise ainsi pour chaque contour un point de départ, puis la suite des codes des directions suivant lesquelles sont obtenus successivement les points de contour (cf. figure 2.15).

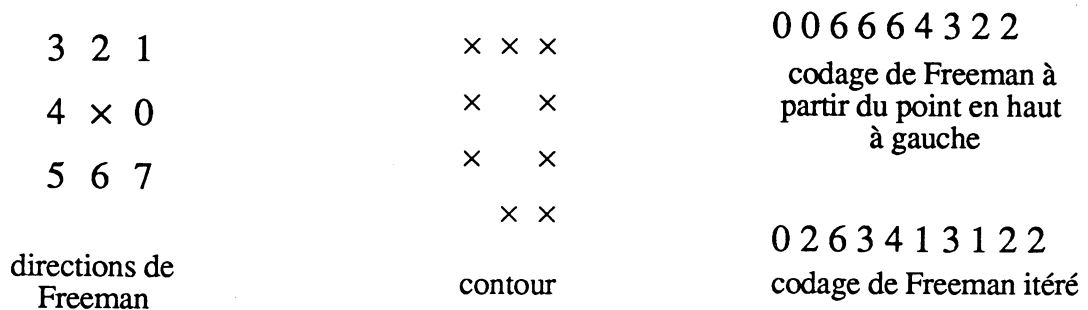


Figure 2.15. Codage des contours par la méthode de Freeman.

Ce codage fournit une bonne compression des données. Par contre, il n'autorise qu'assez difficilement les opérations ensemblistes sur les objets (union, intersection, ...).

D'autres codages (codage de Freeman itéré par exemple) réduisent encore la taille des données. Le codage de Freeman itéré donne à chaque fois la direction et le nombre de déplacements élémentaires effectués dans cette direction, ce qui réduit grandement les répétitions.

Les méthodes d'approximations polygonales tentent, en codant l'objet par les extrémités des arêtes d'un polygone l'approchant, de réduire encore la taille des données, tout en essayant de minimiser l'erreur d'approximation commise. Elles procèdent soit par analyse des changements de direction (par les codes de Freeman), soit par analyse des courbures discrètes [Lux 85], soit par découpage récursif [Pavlidis 74] ou encore par approximations itératives.

Ces approximations polygonales permettent, en outre, d'accéder à une description de l'objet codé, tant pour effectuer des travaux de reconnaissance [Stein 90] que pour effectuer des inférences de formes [Burr 81], comme cela est expliqué au chapitre 4.

Les autres modèles de représentation sont tous fondés sur une approche par régions de la forme.

2.2.2. Les arbres

Différents types d'arbres peuvent être élaborés. Les plus couramment utilisés sont les arbres quaternaires (*quadrees*) [Samet 80]. Il s'agit d'une méthode de codage fondé sur un partitionnement des objets à coder. L'image à traiter doit être de dimension $2^n \times 2^n$. Elle est découpée de manière récursive en quatre blocs carrés et ce jusqu'à ce que chacun des blocs soit homogène (tous ses points à 1 ou tous ses points à 0).

Les blocs carrés sont organisés de manière arborescente. Chaque bloc non homogène est découpé en quatre blocs (ses fils). Ainsi, chaque feuille est un bloc homogène (le plus petit est le pixel) ; chaque nœud est un bloc non homogène (cf. figure 2.16)

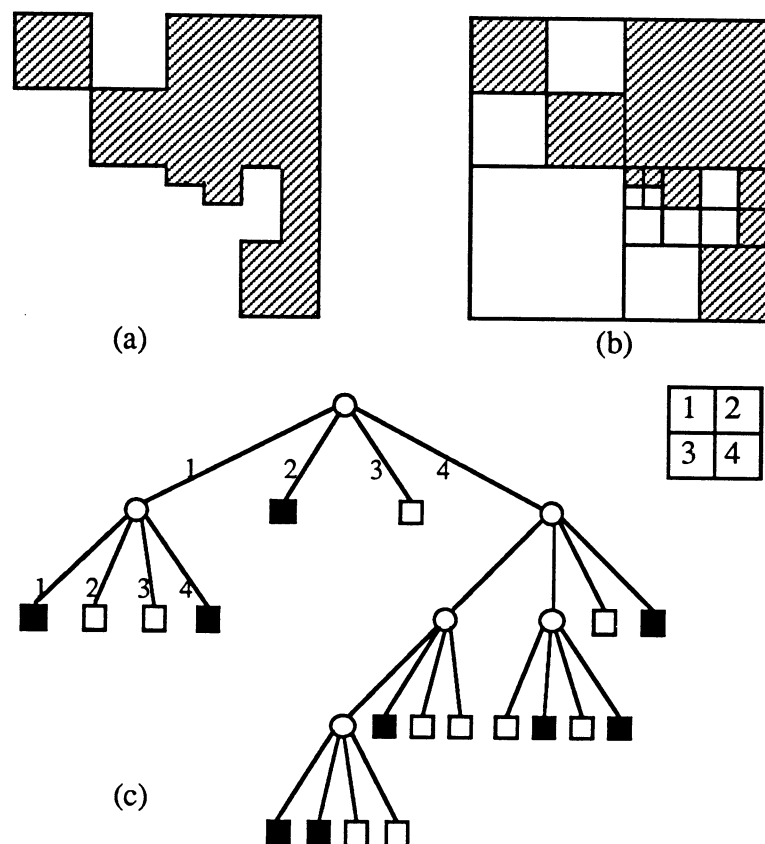


Figure 2.16. Un objet (a), son partitionnement (b) et le *quadtree* correspondant (c).

Cette structure de données hiérarchique est très performante dans la plupart des cas. Elle permet des traitements de l'objet sans pour autant être obligé de manipuler des concepts de compréhension de forme ; pour cela il suffit de tronquer l'arbre à un certain niveau ; on obtient alors une représentation approchée des objets.

Le *quadtree* permet entre autre en travaillant directement sur sa structure, d'effectuer des opérations telles que des translations, rotations, changements d'échelle, intersections, réunions ou parcours [Samet 82] [Laurini 88]. Il permet aussi de retrouver le codage de Freeman de l'objet représenté dans l'image [Dyer 80].

Deux reproches lui sont cependant adressés.

D'une part, le *quadtree* n'est pas invariant en translation. En effet, un objet décalé dans l'image induit un *quadtree* totalement différent de l'original (cf. figure 2.17).

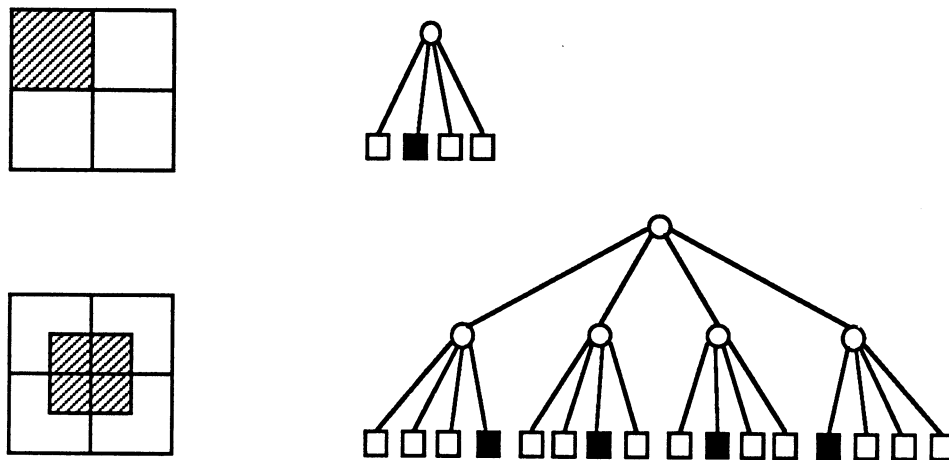


Figure 2.17. Non invariance en translation du *quadtree*.

D'autre part, malgré l'efficacité des algorithmes de recherche de voisins, il est plus difficile d'accéder aux surfaces voisines d'une région donnée que lorsqu'on utilise une représentation matricielle des images. Ceci est dû au fait que les feuilles représentant des surfaces voisines peuvent être très éloignées les unes des autres dans l'arbre.

J'ajouterais même un troisième reproche : le pouvoir descriptif du *quadtree* est assez faible. En effet le codage n'étant pas du tout guidé par la forme des objets, il ne permet que très difficilement de repérer les déformations ou les trous d'un objet décrit par son *quadtree*.

Pour remédier en partie à la non invariance en translation, une extension des *quadtrees* a été proposée : le *Quadtree Medial Axis* [Samet 82] [Samet 83] [Ahuja 84].

Leur principe d'obtention consiste à dilater les blocs du *quadtree* tant qu'ils restent inclus dans la forme. Chaque feuille du *quadtree medial axis* est ainsi étiquetée avec la taille du bloc

maximal inclus (cf. figure 2.18). Ce *quadtree medial axis* est un recouvrement de la forme, contrairement au *quadtree* qui en est une partition.

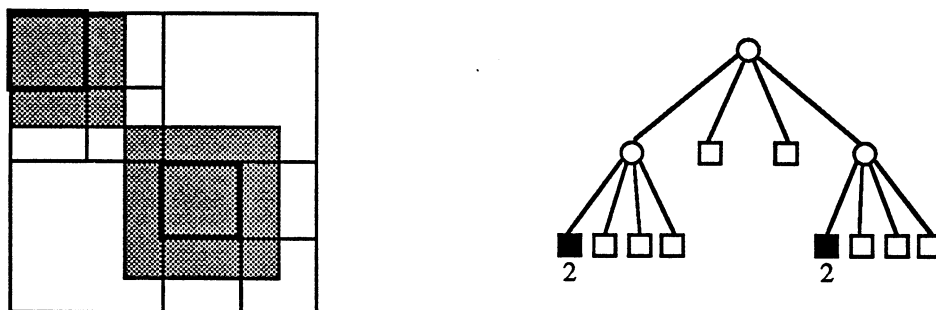


Figure 2.18. *Quadtree Medial Axis*.

D'autres types d'arbres ont été développés. Ils procèdent par partitionnement du plan et hiérarchisation des données [Samet 90a] [Samet 90b].

Parmi eux on peut citer les arbres binaires linéaires (*linear bintrees*) [Samet 88]. Il s'agit en fait d'une modification de la notion de *quadtree*. En effet, au lieu de découper directement l'image en quatre blocs (d'où quatre fils), elle est découpée en deux (deux fils : "est" et "ouest") puis encore en deux (deux fils : "nord" et "sud") (cf. figure 2.19)

Chaque niveau pair de l'arbre correspond donc à un découpage parallèlement à l'axe y tandis que les niveaux impairs opèrent des découpages parallèlement à l'axe des x. Cette méthode semble intéressante dans certains cas surtout si on lui adjoint une architecture matérielle adéquate (parallélisme, *transputers*, ...).

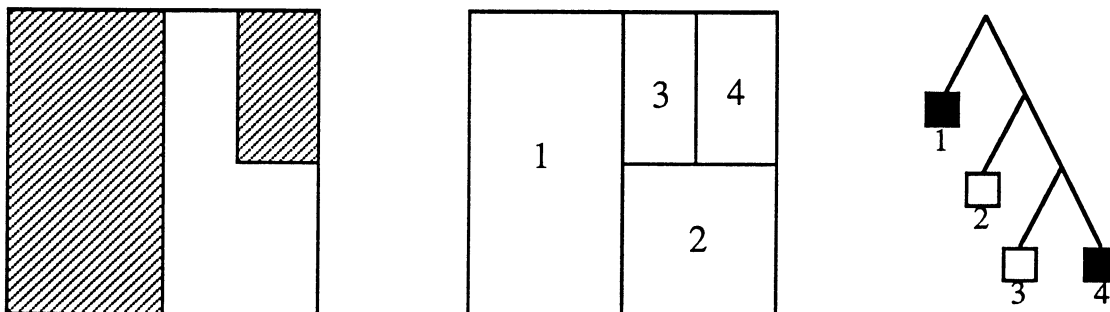


Figure 2.19. Arbre binaire.

2.2.3. Les squelettes

L'idée du codage par squelette est assez ancienne [Blum 64] [Pfaltz 67] ; elle est, bien entendu, inspirée par les modèles animaux et végétaux.

Il s'agit de modéliser un objet à l'aide d'une représentation filiforme, connexe et centrée dans celui-ci, cette représentation devant être significative des élongations et des déformations de l'objet modélisé.

Ce squelette est obtenu, dans le domaine continu, en mettant en œuvre le principe suivant (dit principe du feu de brousse) :

- le contour de la forme sert d'initialisation à une onde qui va se propager de manière isotrope vers l'intérieur de l'objet.
- lorsque deux fronts d'onde, issus de zones d'initialisation distinctes, se rencontrent, la progression est stoppée en un point qui est un point du squelette.

L'ensemble de ces points définit le squelette continu (cf.figure 2.20)

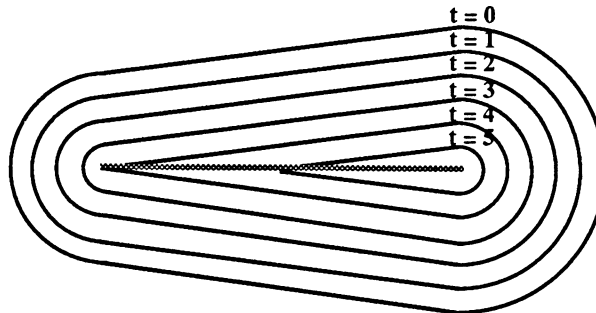


Figure 2.20. Squelette par propagation d'un feu de brousse (extrait de [Blum 64].

Si, de plus, pour chacun des points du squelette on mémorise l'instant auquel il a été défini on obtient une représentation réversible de la forme. En effet, il est possible de régénérer l'objet en excitant les points du squelette dans l'ordre inverse de leur obtention.

Le procédé de régénération est le suivant :

- | Soit B la valeur maximale des t (temps) obtenue sur le squelette.
- | Initialiser une onde aux points de poids β à l'instant 0.
- | Initialiser une onde sur les points de poids $\beta-i$ à l'instant i.
- | A l'instant β le front de l'onde est égal au contour de l'objet initial.

Le squelette ainsi défini n'est affecté ni par les translations, ni par les rotations, ni par les homothéties. Par contre il est très sensible au bruit puisqu'une petite déformation de la forme telle que l'apparition d'une proéminence ou d'un trou le modifie complètement. (cf. figure 2.21)

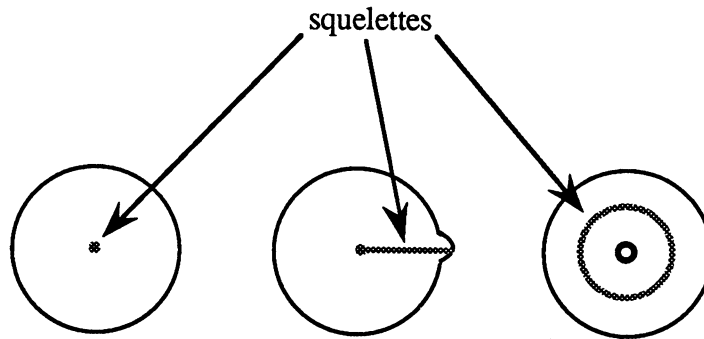


Figure 2.21. Sensibilité au bruit du squelette.

La notion de squelette binaire (squelette discret puisque le support de l'image est \mathbb{Z}^2) est un peu plus complexe à aborder.

Tout d'abord il faut signaler que le squelette binaire ne peut pas faire l'objet d'une définition formelle. En effet, la notion simple de centrage n'est pas définie dans \mathbb{Z}^2 si la largeur est paire. Ainsi, les squelettes binaires ne peuvent être définis que comme le résultat d'un algorithme opéré sur un objet binaire.

Des algorithmes nombreux et variés ont été développés à ce sujet.

Certains d'entre eux procèdent par érosions successives d'autres par symétrie.

Les méthodes procédant par érosions successives à partir du contour de l'objet le font, soit en vérifiant à chaque fois que l'érosion n'a pas modifié la topologie de l'objet (type de connexité), soit en vérifiant des conditions de connexités locales sur le point à éroder (non modification du *nombre de connexité*). Certaines encore procèdent par érosion à l'aide d'automates morphologiques [Nguyen 87].

Les méthodes qui utilisent les symétries procèdent, soit par calcul d'axes de symétrie à partir d'une approximation des contours, soit en effectuant un suivi de la ligne de crête de l'image des lignes de niveau de l'objet [Rosenfeld 68] [Montanari 68] [Dorst 86] [Borgefors 88].

Certaines de ces méthodes pouvant être utilisées avec les différentes distances discrètes (d_4 , d_8 , chanfrein, euclidienne, ...) avec plus ou moins de parallélisme plus ou moins massif, il est facile de recenser plus d'une trentaine de méthodes et d'implémentations différentes pour le calcul des squelettes binaires [Leymarie 89].

Toutes les méthodes donnent des résultats assez proches les uns des autres mais toujours différents. Ceci est un inconvénient mais c'est aussi la richesse qui fait que les squelettes sont largement utilisés pour la représentation des formes.

2.2.4. Axe médian et ligne médiane

L'axe médian utilise les centres des boules incluses [Pfaltz 67]. De manière plus formelle, l'axe médian est le lieu des centres des boules maximales entièrement contenues dans l'objet qui ne sont pas incluses dans d'autres boules [Pizer 87].

Pour chaque point de l'axe médian, on stocke sa "distance au fond", c'est-à-dire le rayon de la boule qu'il code.

Le codage par axe médian définit un recouvrement par des boules de taille maximale (cf. figure 2.22).

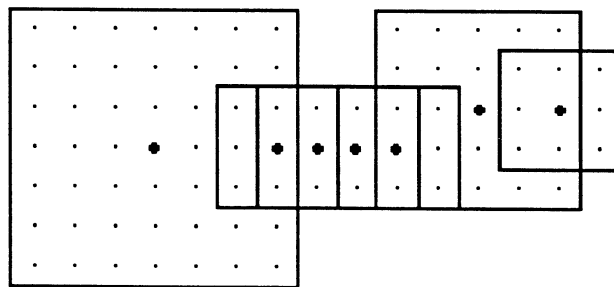


Figure 2.22. Exemple d'axe médian en d_g .

La transformation en axe médian est réversible ; de plus, elle est obtenue assez facilement [Pfaltz 67]. L'axe médian permet aussi d'effectuer de nombreux traitements (appartenance de points à une forme, trace de contours, union, intersection, calcul de surfaces, ...).

Les principaux reproches faits à l'encontre de l'axe médian sont :

- sa sensibilité au bruit
- sa non invariance en rotation
- son épaisseur dans certains cas
- sa non connexité.

La non invariance en rotation de l'axe médian est due à l'utilisation de métriques pour lesquelles les boules ne sont pas rondes donc sensibles aux rotations. Certains travaux permettent de résoudre ces problèmes en utilisant des distances autres que d_4 et d_8 [Arcelli 88].

Lorsque l'objet est de largeur paire, l'axe médian est épais (deux points d'épaisseur).

Ce problème peut être résolu en effectuant le calcul dans un espace fictif de sur-échantillonnage [Bertrand 84].

Comme on peut le remarquer facilement l'axe médian est déconnecté (cf. figure 2.23). Ceci est un problème important car cette non connexité rend complexe la définition de parcours de l'objet à partir de l'axe médian. C'est à partir de là qu'est apparu le concept de ligne médiane (axe médian connecté).

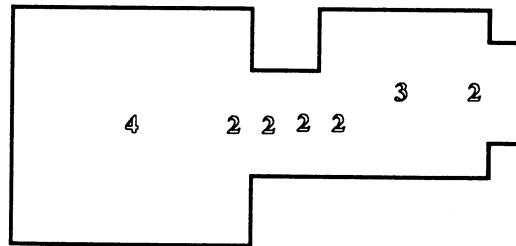


Figure 2.23. Non connexité de l'axe médian (en relief).

Le calcul de la ligne médiane peut se faire soit par pelage soit par reconnexion des points de l'axe médian.

Les méthodes par pelage itératif procèdent par érosions successives à partir des points de contours de l'objet [Pavlidis 80] [Arcelli 81].

Les méthodes consistant à reconnecter les points de l'axe médian sont plus efficaces. Elles procèdent en deux passages sur l'image de distance en prenant garde à ne pas supprimer pas les maximums locaux lors du calcul de l'axe médian [Arcelli 85] [Montanvert 85] (cf. figure 2.24).

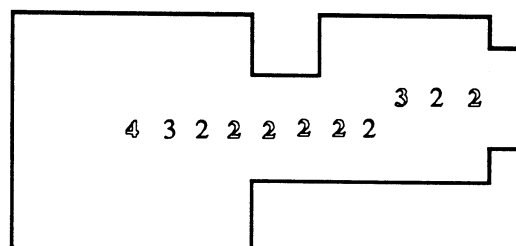


Figure 2.24. Ligne médiane.

Le codage par ligne médiane est réversible mais il reste sensible au bruit. Il fournit un moyen rapide de squelettisation de formes.

Malheureusement, il n'y a pas toujours unicité de la ligne médiane. En effet, lors du calcul, il est parfois nécessaire de faire un choix entre plusieurs points candidats pour effectuer une connexion.

Certaines extensions des notions de ligne médiane avec d'autres distances sont actuellement en cours l'élaboration

2.2.5. Représentations hiérarchiques

On regroupe sous le terme de représentations hiérarchiques toutes les méthodes mettant en œuvre une hiérarchie entre les différentes composantes de l'objet.

Il s'agit en fait d'organiser des primitives de manière à aboutir à une description de la forme codée.

2.2.6. Le graphe de la ligne médiane

C'est un codage de la ligne médiane aboutissant à une représentation hiérarchique et permettant de résoudre le problème du choix des chemins de connexion.

Les primitives (points de l'axe médian) sont organisées à l'aide d'un graphe.

Le graphe de la ligne médiane est un graphe planaire orienté. Chaque sommet représente un ensemble de points de l'axe médian (centre de boules) de même poids (même distance au fond).

Les arcs représentent les chemins de connexion ; ils sont orientés des poids faibles vers les poids forts [Montanvert 87a] (cf. figure 2.25).

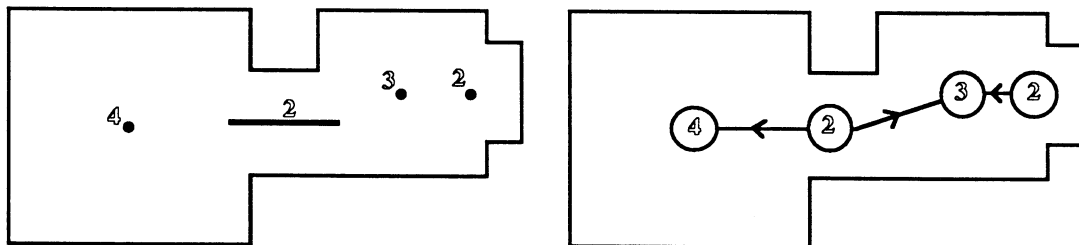


Figure 2.25. Graphe de la ligne médiane.

Pour chacun des sommets du graphe des informations topologiques, topographiques et structurelles sont mémorisées :

- la distance au fond (aussi appelée poids)
- la liste des points qu'il représente (coordonnées des points de l'axe médian qu'il code)
- un type.

Le type correspond à une réalité structurelle de la composante de la forme codée par rapport à la forme globale. Il est déterminé à l'aide de la configuration locale du graphe.

Trois types sont définis :

- principal : attribué aux sommets ne possédant pas d'arc sortant (arc dont ils sont l'extrémité initiale).
- rétrécissement : attribué aux sommets possédant au moins deux arcs sortants.
- déformation : attribué aux sommets possédant un arc sortant.

Les sommets de déformation se répartissent en :

- sommets de proéminence stricte : sommets ne possédant pas d'arc rentrant (arc dont ils sont l'extrémité terminale).
- sommets de proéminence large : sommets possédant au moins un arc rentrant (cf. figure 2.26).

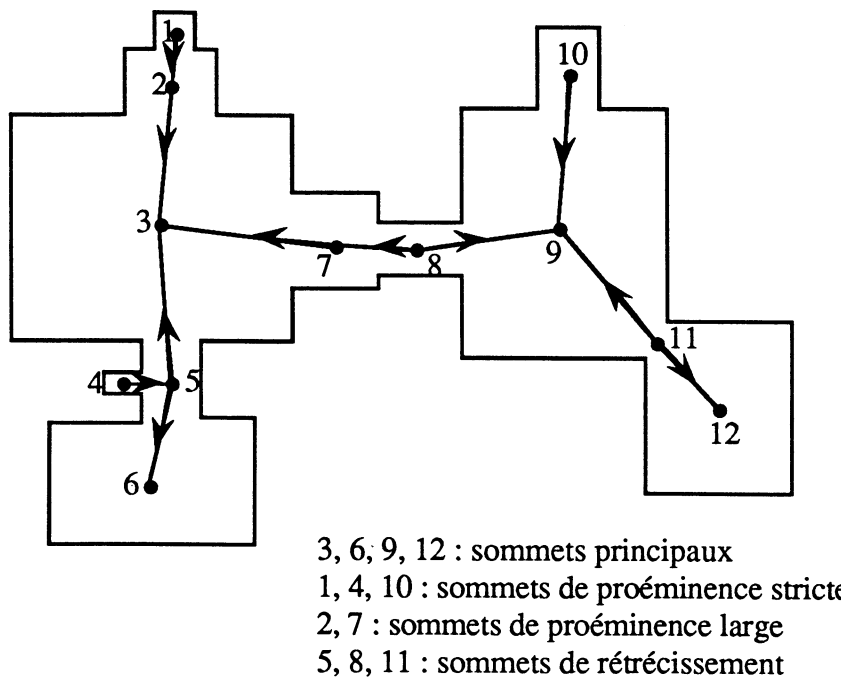


Figure 2.26. Graphe de la ligne médiane.

C'est à partir du type des différents sommets que peut être définie une structure hiérarchique [Montanvert 87b].

On distingue trois niveaux dans cette hiérarchie :

- principal
- rétrécissement
- déformation.

Il faut noter que le graphe de la ligne médiane n'est pas exactement un graphe au sens formel du terme ; ceci parce que chaque sommet comporte une information de localisation nécessaire à la

réversibilité de la représentation et parce qu'il est défini uniquement en tant que projection sur l'image qu'il représente.

Le graphe de la ligne médiane constitue un mode de codage des images binaires efficace (réversibilité, hiérarchisation). C'est aussi un excellent descripteur de formes : il permet de repérer les déformations et rétrécissements de la forme mais aussi, par exemple, de compter et de localiser les trous d'un objet [Rolland 89a]. Le graphe de la ligne médiane est aussi un très bon outil de traitement d'images ; il permet d'effectuer des filtrages (filtrages relatifs ou avec contrôle de la connexité), des décompositions, des partitionnements de formes [Montanvert 87d], des manipulations orientées objet [Montanvert 88], ou même de définir des méthodes de parcours de surfaces [Montanvert 87c].

Cette représentation par graphe de la ligne médiane est une représentation symbolique de l'objet. Les manipulations de l'objet sont donc faites par son intermédiaire [Montanvert 89].

Dans la suite c'est à l'aide de cette structure de graphe de la ligne médiane que seront codées les images. Ce graphe de la ligne médiane sera en outre utilisé à des fins de description et de représentation de l'objet dans le chapitre 5.

Les différents modèles en analyse de forme ayant été abordés sous leur aspect bidimensionnel il convient maintenant de voir quels sont les modèles utilisés dans le domaine tridimensionnel.

Chapitre 3

Modèles en analyse de formes : aspect tridimensionnel

Comme cela a été fait pour le plan discret dans le chapitre précédent, il convient de préciser quelques définitions et notions indispensables lorsqu'on souhaite travailler dans l'espace discret \mathbb{Z}^3 . Ce chapitre sera aussi consacré à la description des différents modèles utilisés pour la représentation de volumes.

3.1. Définitions et notions de base

Avant toute chose il faut noter qu'il est difficile de représenter des volumes dans le plan ; c'est pourquoi, contrairement aux définitions données dans le plan, les définitions seront, seulement dans la mesure du possible, étayées par un exemple.

Il serait logique de penser que toutes les définitions données pour \mathbb{Z}^2 sont généralisables dans \mathbb{Z}^3 , malheureusement ce n'est pas toujours le cas, l'espace offrant beaucoup plus de possibilités que le plan.

3.1.1. Espace discret

La représentation de volumes 3D par ordinateur requiert un échantillonnage du volume afin d'en extraire des valeurs discrètes. Cet échantillonnage de l'espace continu est effectué à l'aide d'un

maillage, tableau tridimensionnel de points régulièrement espacés de $(\delta x, \delta y, \delta z)$. Chaque point de ce maillage correspond à un point de l'espace discret.

Ainsi, à chaque point $P = (x_P, y_P, z_P)$ de l'espace discret est associé l'ensemble des points (x, y, z) tels que :

$$x_P - \delta x/2 \leq x < x_P + \delta x/2$$

$$y_P - \delta y/2 \leq y < y_P + \delta y/2$$

$$z_P - \delta z/2 \leq z < z_P + \delta z/2$$

Dès lors, chaque point de l'espace discret référence un volume élémentaire cubique appelé *voxel* (le terme voxel abrège l'expression "*volume element*" comme le fait pixel pour "*picture element*" en deux dimensions).

Cet échantillonnage de l'espace nécessite un pavage polyédrique fermé compact [Srihari 81]. Parmi les cinq polyèdres réguliers que sont le tétraèdre (quatre faces), le cube (six faces), l'octaèdre (huit faces triangulaires), le dodécaèdre pentagonal (douze faces pentagonales) et l'isocaèdre (vingt faces qui sont des triangles équilatéraux égaux), seul le cube permet d'effectuer un échantillonnage régulier. Cela est différent du cas 2D où trois maillages réguliers sont possibles.

En dehors des polyèdres réguliers d'autres motifs sont utilisables pour échantillonner l'espace (modèle "cubique face centré" par exemple) mais leur champ d'utilisation est très réduit à cause des difficultés de raisonnement qu'ils occasionnent [Toriwaki 82] [Hafford 84]. Dans la suite nous n'utiliserons que le maillage cubique.

Ainsi échantillonné, tout volume discret peut être défini par sa fonction caractéristique f :

$$\begin{array}{lcl} f : & \mathbb{Z}^3 & \rightarrow \mathbb{I} \\ & (i, j, k) & \rightarrow f(i, j, k) \end{array}$$

Si \mathbb{I} se réduit à l'ensemble $\{0, 1\}$, le volume est dit binaire.

Comme pour les images 2D, les volumes peuvent être codés à l'aide de matrices ; cette fois elles doivent être tridimensionnelles.

Pour les mêmes raisons que dans le plan discret, il est nécessaire de définir des distances pour l'espace discret.

3.1.2. Distances dans \mathbb{Z}^3

3.1.2.1. Distance euclidienne

La distance euclidienne entre deux voxels U et V est définie par :

$$\begin{aligned} d_E(U, V) &= d_E((u_1, u_2, u_3), (v_1, v_2, v_3)) \\ &= \sqrt{\sum_{i=1}^3 (v_i - u_i)^2} \end{aligned}$$

La distance entre les coins opposés d'un cube $N \times N \times N$ est $N\sqrt{3}$ et celle entre deux des coins opposés de l'une de ses faces est $N\sqrt{2}$. Elle permet de définir des boules euclidiennes (cf. figure 3.1).

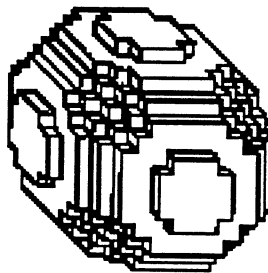


Figure 3.1. Boule euclidienne discrète.

Ces valeurs n'étant pas entières et les fonctions pour les rendre entières telles que $\lceil d_E \rceil$ ou $\lfloor d_E \rfloor$ n'étant pas compatibles avec les propriétés nécessaires à la définition d'une distance, il est nécessaire, là aussi, de définir des métriques à valeurs entières.

3.1.2.2. Distance d_6

Elle est la généralisation de la distance de Manhattan.

Sa définition est :

$$d_6(U, V) = \sum_{i=1}^3 |u_i - v_i|$$

Elle permet de définir la notion de *6-voisinage* d'un point :

$$N_6(U) = \{ V(v_1, v_2, v_3) \text{ tel que } |u_1 - v_1| + |u_2 - v_2| + |u_3 - v_3| = 1 \}$$

En fait deux voxels sont *6-voisins* s'ils partagent l'une de leurs faces (ils sont six) (cf. figure 3.2).

La distance entre les coins opposés d'un cube $N \times N \times N$ est $3N$, celle entre les coins opposés de l'une de ses faces est $2N$.

Les boules discrètes définies par d_6 sont des octaèdres discrets dont les faces sont inclinées de plus ou moins 45° par rapport à chacun des axes ; c'est l'équivalent du diamant en 2D (cf. figure 3.2).



Figure 3.2. 6-voisinage d'un point et boule discrète selon d_6 .

On peut démontrer que :

$$\forall U, V \in \mathbb{Z}^3, d_6(U, V) \geq d_E(U, V).$$

Pour la distance d_6 , l'image de distance (on devrait dire "volume de distance") peut être calculée en deux passages sur le volume à l'aide de masques [Borgefors 84]. Le principe d'obtention est le même que celui utilisé en 2D, le premier passage se fait selon l'une des diagonales et le second dans le sens inverse. Les masques utilisés sont cette fois des masques tridimensionnels (cf. figure 3.3).

Passage Aller

∞	∞	∞
∞	1	∞
∞	∞	∞

∞	1	∞
1	0	

Passage Retour

	0	1
∞	1	∞

∞	∞	∞
∞	1	∞
∞	∞	∞

Figure 3.3. Masques aller et retour pour d_6 .

3.1.2.3. Distance d_{26}

Elle est la généralisation dans l'espace discret de la distance de l'échiquier définie dans le plan. Elle est donnée par :

$$d_{26}(U, V) = \text{Max}_{i=1}^3 |u_i - v_i|$$

Dans ce cas, le *26-voisinage* d'un voxel est constitué de tous les voxels partageant avec lui, soit l'une de leurs faces (ils sont six), soit uniquement une de leurs arêtes (ils sont douze), soit uniquement un de leurs sommets (ils sont huit). Le *26-voisinage* d'un voxel comprend donc vingt-six voxels (cf. figure 3.4).

La distance entre deux coins opposés d'un cube $N \times N \times N$ est N , de même que la distance entre deux coins opposés de l'une des faces de ce même cube.

Ainsi, la boule $B_{26}(V, r)$ forme un cube centré en V et ayant sur une arête $2r+1$ voxels (cf. figure 3.4).

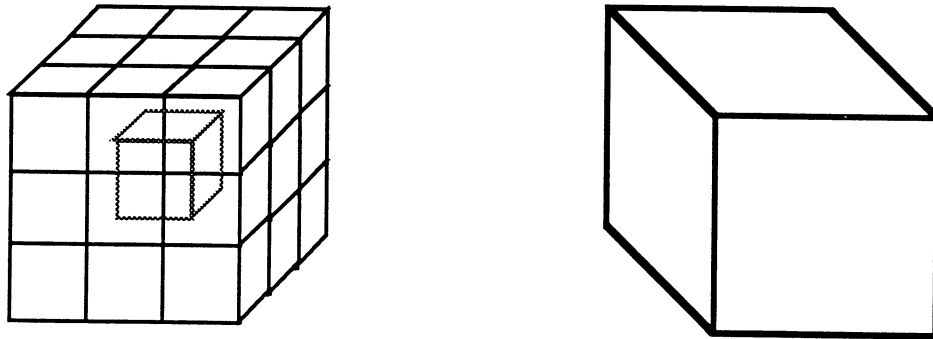


Figure 3.4. 26-voisinage d'un point et boule discrète selon d_{26} .

On peut démontrer que :

$$\forall U, V \in \mathbb{Z}^3, d_{26}(U, V) \leq d_E(U, V).$$

Là encore l'image de distance peut être calculée en deux passages à l'aide de deux masques (cf. figure 3.5)

Passage Aller

1	1	1
1	1	1
1	1	1

1	1	1
1	0	

Passage Retour

	0	1
1	1	1

1	1	1
1	1	1
1	1	1

Figure 3.5. Masques aller et retour pour d_{26} .

La distance d_{26} est la distance duale de d_6 .

3.1.2.4. Distance d_{18}

Il n'en existe pas de définition formelle simple. Elle est en fait définie par l'intermédiaire de la notion de voisinage. On a :

$$N_{18}(U) = \left\{ V(v_1, v_2, v_3) \text{ tel que } 0 < \sum_{i=1}^3 |u_i - v_i|^2 \leq 2 \right\}.$$

Deux voxels sont voisins (*18-voisins*) s'ils ont, soit l'une de leurs faces, soit uniquement une de leurs arêtes en commun. Chaque voxel a ainsi dix-huit voisins (cf. figure 3.6).

La distance entre les coins opposés d'un cube $N \times N \times N$ est $2N$, celle entre deux coins opposés de l'une de ses faces est N .

Les boules définies par d_{18} sont des polyèdres discrets à quatorze faces ; ce sont en fait des cubes (boules de d_{26}) qui ont été tronqués parallèlement aux faces de l'octaèdre (boule de d_6) (cf. figure 3.6).

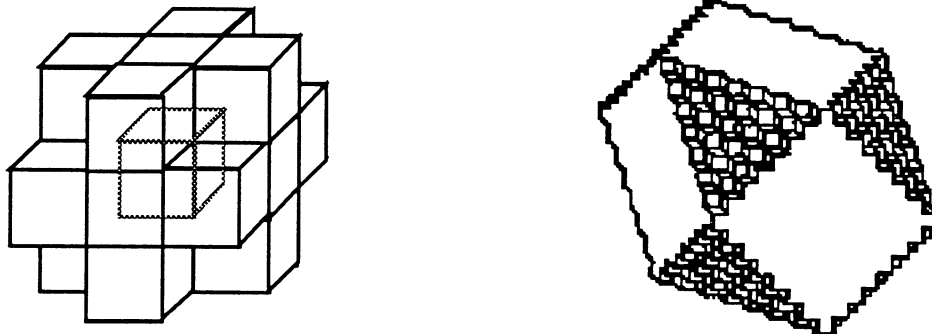


Figure 3.6. 18-voisinage d'un point et boule discrète selon d_{18} .

L'image de distance se calcule là encore à l'aide de deux masques (cf. figure 3.7)

Passage Aller

∞	1	∞
1	1	1
∞	1	∞

1	1	1
1	0	

Passage Retour

	0	1
1	1	1

∞	1	∞
1	1	1
∞	1	∞

Figure 3.7. Masques aller et retour pour d_{18} .

On démontre que, pour tous les points de l'espace à l'exception des huit points $(\pm 1, \pm 1, \pm 1)$ [Das 88], on a :

$$\forall U, V \in \mathbb{Z}^3, d_{18}(U, V) \leq d_E(U, V).$$

Cette propriété, adjointe à ses semblables pour les distances d_6 et d_{26} , permet de définir d'autres distances telles que les distances (hyper-)octogonales 6/18, 6/26 ou même 18/26, en combinant les distances correspondantes.

3.1.2.5. Distances du chanfrein

Comme dans le plan, il semble intéressant d'approcher un peu mieux la distance euclidienne dans l'espace [Borgefors 84].

Les distances du chanfrein le font. Elles sont basées sur la même idée qu'en 2D. Ainsi, la distance du chanfrein 3-4-5 permet d'approcher $\sqrt{2}$ par $4/3$ et $\sqrt{3}$ par $5/3$.

Le calcul de l'image de distance se fait là aussi en deux passages (aller et retour) sur le volume à l'aide de masques (cf. figure 3.8).

Passage Aller

5	4	5
4	3	4
5	4	5

4	3	4
3	0	

Passage Retour

	0	3
4	3	4

5	4	5
4	3	4
5	4	5

Figure 3.8. Masques aller et retour pour la distance du chanfrein 3-4-5.

Ces distances, comme les distances octogonales, bien qu'étant de meilleures approximations que d_6 , d_{18} et d_{26} ne sont pas beaucoup utilisées. Ceci est en partie dû aux formes bizarres des

boules qu'elles définissent (cf. figure 3.9), mais surtout parce que l'utilisation de telles distances en 3D est relativement récente.

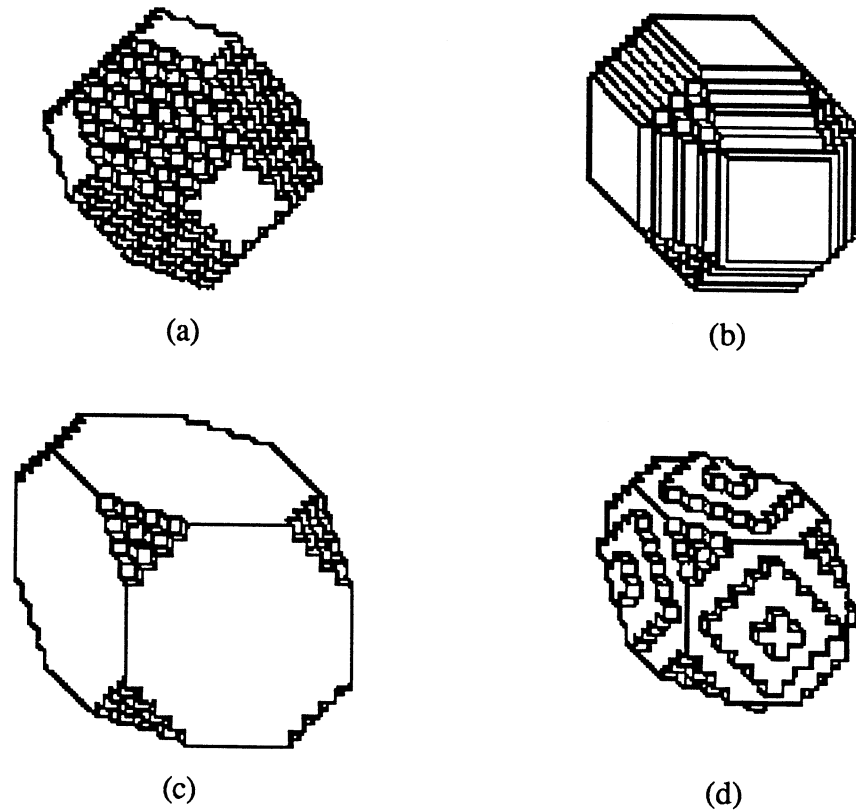


Figure 3.9. Différentes boules : pour les distances octogonales 6/18 (a), 6/26 (b), 18/28 (c) et pour la distance du chanfrein 3-4-5 (d) (extrait de [Borgefors 84]).

3.1.3. Connexité, nombre d'Euler

Il est important de compléter ici certaines définitions données dans le plan discret.

La notion de *chemin connexe* est la même que dans le plan. Elle est directement liée à la notion de voisinage utilisée qui détermine les points potentiellement consécutifs d'un point donné. Un chemin 26-connexe peut donc être décomposé en trois parties potentiellement vides : une partie où les voxels sont 6-voisins, une partie où les voxels sont 18-voisins et non 6-voisins, et une partie où les voxels sont 26-voisins sans être ni 6- ni 18-voisins [Okabe 83].

Les notions de *courbe* et de *composante connexe* sont les généralisations de celles données dans le plan.

De même qu'en 2D, le complémentaire d'un volume doit être analysé avec la distance duale de celle utilisée pour l'objet. Par exemple si on utilise la 6-connexité pour l'objet il faut utiliser la 26-connexité pour le fond, et *vice versa*. Dans le cas où on utilise la 18-connexité il est nécessaire de définir sa distance duale, parfois notée d'₁₈.

A l'aide des notions de voisinage, on définit la notion de point *bord* :

On appelle point *bord* d'un objet tout point dont l'un des 6-voisins est dans le complémentaire de cet objet.

On appelle point de type "Ouest" (respectivement Est, Nord, Sud, Haut et Bas) tout point bord dont le voisin dans la direction Ouest (respectivement Est, Nord, Sud, Haut et Bas) est dans le complémentaire de l'objet considéré.

Les directions Nord, Sud, Est, Ouest, Haut et Bas sont définies selon le schéma de la figure 3.10.

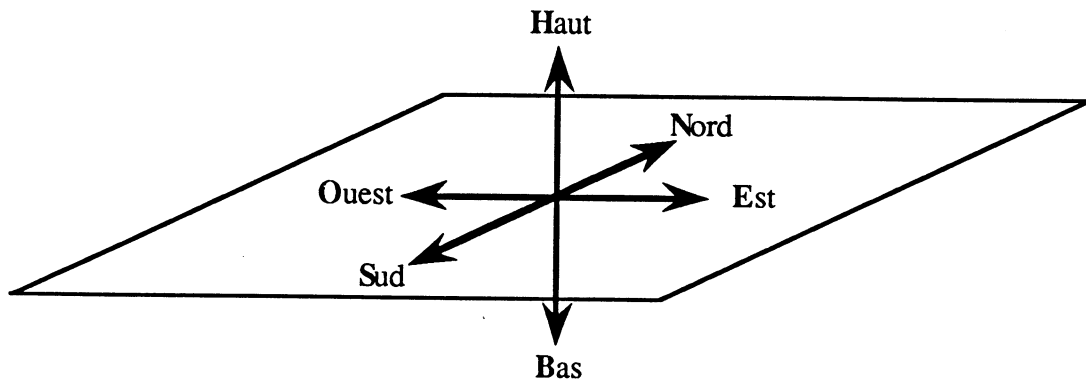


Figure 3.10. Les 6 directions de l'espace.

On appelle *trou* de l'objet toute composante connexe finie du complémentaire de l'objet.

On définit ainsi deux sortes de trous.

Les *cavités* sont définies comme les composantes connexes finies (au sens de la connexité du complémentaire de l'objet) du complémentaire de l'objet.

Le *fond* est donc simplement la composante connexe infinie du complémentaire.

On définit aussi la notion d'*anse*, elle se forme quand, à partir d'un solide, on perce un tunnel le traversant de part en part. Ces objets comportant des anses peuvent être mis en bijection avec des tores.

C'est à partir du nombre d'anses qu'est défini le "genre" de l'objet, c'est un invariant topologique.

On montre [Hilbert 52] que :

$$S - a + f = 2 - 2 T$$

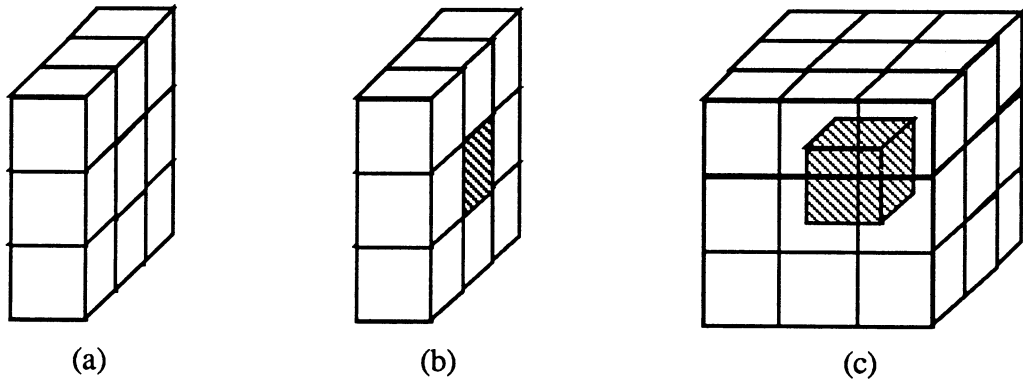
où S est le nombre de sommets de l'objet, a le nombre d'arêtes et f le nombre de faces. T représente le nombre d'anses (tunnels) (cf. figure 3.11).

On peut généraliser la notion de nombre de connexité défini dans le plan en définissant le *nombre de Euler* ϵ :

$$\epsilon = K - T + C$$

où K est le nombre de composantes, T le nombre d'anses et C le nombre de cavités.

Ce nombre d'Euler sera utilisé pour calculer les squelettes 3D.



<i>Objet</i>	K	C	T	s	a	f	ε
a	1	0	0	32	60	30	1
b	1	0	1	32	64	32	0
c	1	1	0	56	108	54	2

K : nombre de composantes ; **C** : nombre de cavités ; **T** : nombre d'anses
s : nombre de sommets ; **a** : nombre d'arêtes ; **f** : nombre de faces
ε : nombre d'Euler

(d)

Figure 3.11. Objets 3D : (a) objet fermé ; (b) objet avec anse ; (c) objets avec cavité ; (d) invariants topologiques.

3.2. Modèles de représentation tridimensionnels

Etant donné la taille des données nécessaires pour mémoriser un volume sous forme matricielle, divers modèles de représentations des volumes ont été développés. On retrouve les modes de représentation utilisés en 2D ; deux approches sont généralement utilisées : soit une approche de type surface (contour), soit une approche de type volume (région) [Péroche 88].

Les approches par la surface tentent de modéliser le volume par l'intermédiaire de sa surface. Les approches par volume modélisent l'ensemble du volume par partition ou recouvrement de l'objet à l'aide de volumes élémentaires.

Avant de présenter la modélisation par la surface et la modélisation volumique il convient de noter ici l'existence de la modélisation par fil de fer. Ce modèle est le modèle le plus simple. Il ne conserve que les coordonnées des sommets ainsi que les arêtes les joignant. La simplicité de représentation rend la visualisation très rapide. Par contre elle ne permet pas de distinguer le plein du vide ce qui rend difficile et même impossible certains calculs tels que le calcul de volume, de masse ou l'élimination de parties cachées. Cette modélisation ne permet d'ailleurs pas toujours de donner un sens physique aux objets construits (possibilité de créer des "illusions d'optique") (cf. figure 3.12) [Hofstadter 85].

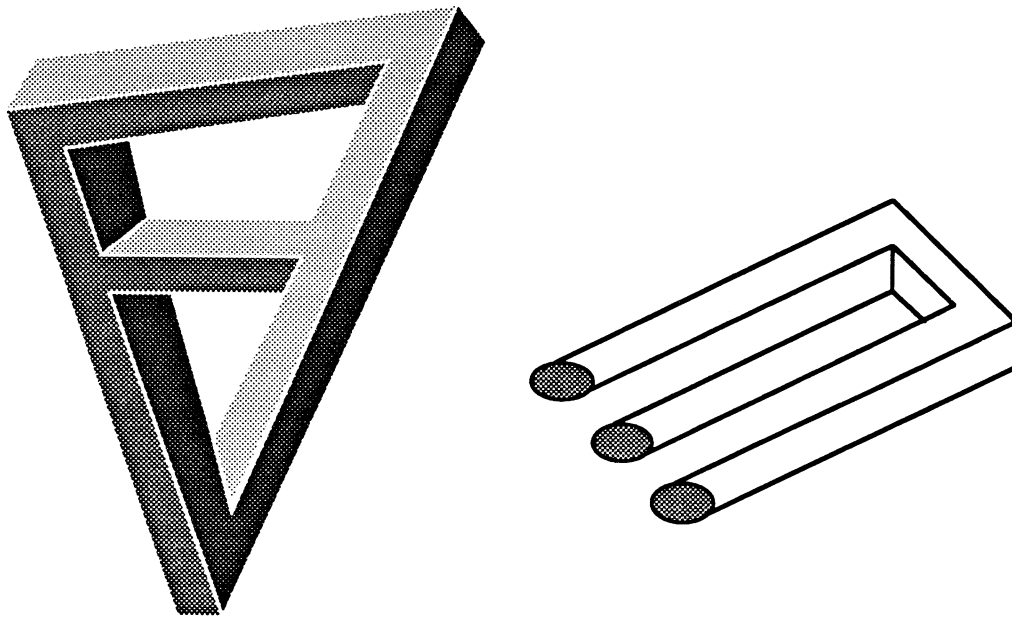


Figure 3.12. Illusions.

3.2.1. Représentation par modélisation de la surface

Ces méthodes utilisent une modélisation de la surface de l'objet, c'est cette surface qui sert ensuite de représentation.

Plusieurs types de surfaces peuvent être utilisées pour les représentations surfaciques. La modélisation se fait généralement par facettes, chacune de ces facettes pouvant être une surface plane (triangle ou carreau), une courbe paramétrique [Grimson 83] ou une surface gauche (spline [Wu 77], carreau de Coons, etc).

Ces différentes méthodes permettent, en ne stockant que le contour, d'obtenir des représentations dont le volume de données n'est pas trop important. Elles fournissent une description assez naturelle des volumes. Ces descriptions facilitent les transformations géométriques telles que rotations et translations ; ce sont d'ailleurs ces méthodes de représentation qui sont largement utilisées en infographie et en Conception Assistée par Ordinateur [Bartels 87].

Par contre, ces descriptions par les surfaces ne permettent qu'avec difficulté la réalisation d'opérations logiques entre volumes telles que la réunion, l'intersection ou la différence.

De plus, les surfaces ne définissent pas toujours des volumes, la validation des différents algorithmes est donc rendue difficile.

3.2.2. Les arbres

De nombreuses structures arborescentes sont utilisées pour la modélisation des volumes. Les principales sont les arbres CSG, les *octrees* et les arbres multiples.

- Les arbres CSG

Les arbres CSG (*Constructive Solid Geometry*) sont des arbres binaires pour lesquels les nœuds sont des opérateurs logiques et les feuilles des volumes élémentaires (polyèdres, cylindres, cônes, sphères, ...) (cf. figure 3.13).

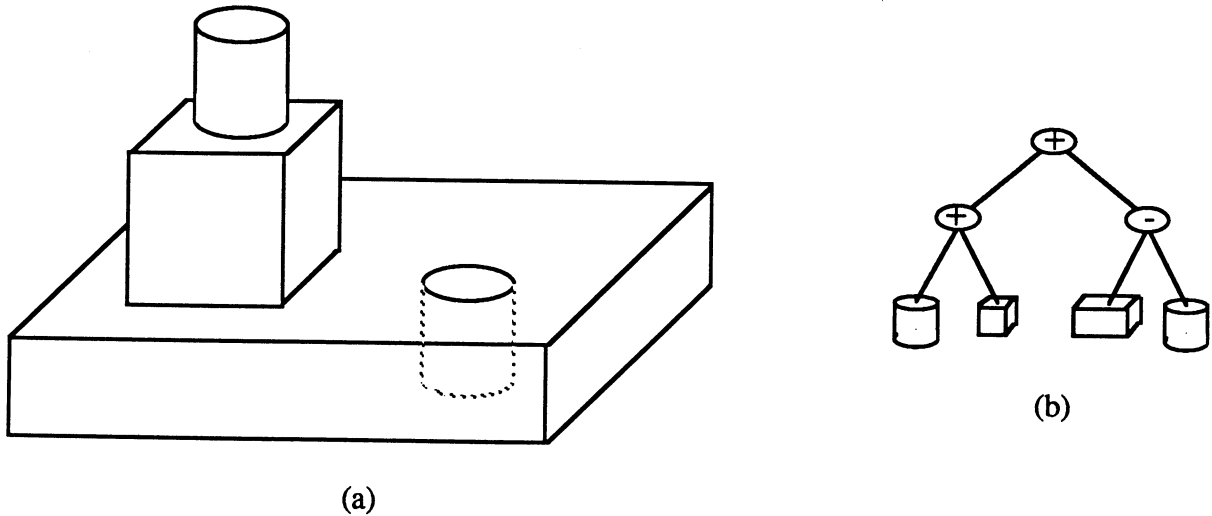


Figure 3.13. Un objet (a) et l'arbre CSG (b) correspondant.

Ils sont construits à partir d'une famille de volumes élémentaires donnée qui leur permet de modéliser un plus ou moins grand nombre de volumes [Requicha 80].

Ce mode de représentation est non ambigu (un arbre représente un seul volume) mais n'est pas unique (un volume peut être modéliser par plusieurs arbres). Ceci rend d'ailleurs délicat le choix des primitives utilisées.

L'inconvénient principal d'une telle méthode est que son domaine de représentation est clos pour une famille de volumes élémentaires donnée. Elle ne permet que difficilement de prévoir si tel ou tel volume va ou ne va pas pouvoir être modéliser. D'autre part il faut noter que ce modèle de représentation ne permet que très difficilement de savoir si un point appartient ou non au volume. C'est pourquoi cette méthode de représentation est plus utilisée pour construire des volumes (Conception Assistée par Ordinateur) que pour représenter des volumes obtenus par d'autres moyens.

• Les *octrees*

Ce sont des arbres octaux construits sur le même principe que les *quadrees*. Leur construction est obtenue en découpant le volume initial, qui doit être de dimension $2^n \times 2^n \times 2^n$, en huit cubes et cela itérativement jusqu'à ce que chaque cube soit homogène [Jackins 80] [Chen H.H. 88] (cf. figure 3.14).

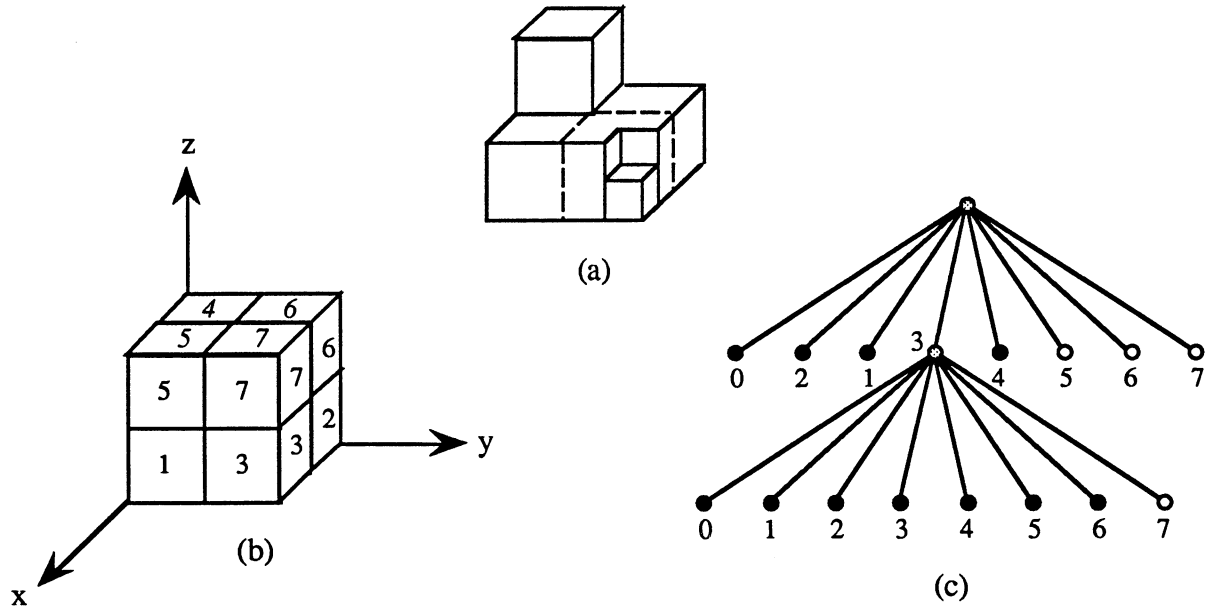


Figure 3.14. (a) un objet, (b) les codes pour l'obtention de l'*octree* correspondant (c).

Les *octrees* ont les mêmes propriétés que les *quadrees*.

L'*octree* permet, en travaillant sur sa structure, d'effectuer des rotations, translations [Jackins 80], d'extraire le contour de l'objet codé [Kunii 85], de mesurer des volumes et des surfaces, de calculer des centres de gravités et d'effectuer des opérations logiques (intersection, réunion, ...).

Par contre les *octrees* ne sont pas invariants en translation. Pour remédier à ce problème, la notion d'*octree* généralisé a été développée.

La structure d'*octree* est, parmi les autres structures arborescentes, la plus couramment employée.

On peut noter que, comme pour le *quadtree*, il est possible, au lieu d'effectuer un découpage complet de l'espace en une seule étape, de le faire en plusieurs pour obtenir des arbres binaires. C'est le cas avec les arbres binaires 3D où l'on effectue, à la place d'un découpage en huit cubes, successivement trois découpages en deux volumes parallélépipédiques [Samet 88] (cf. figure 3.15).

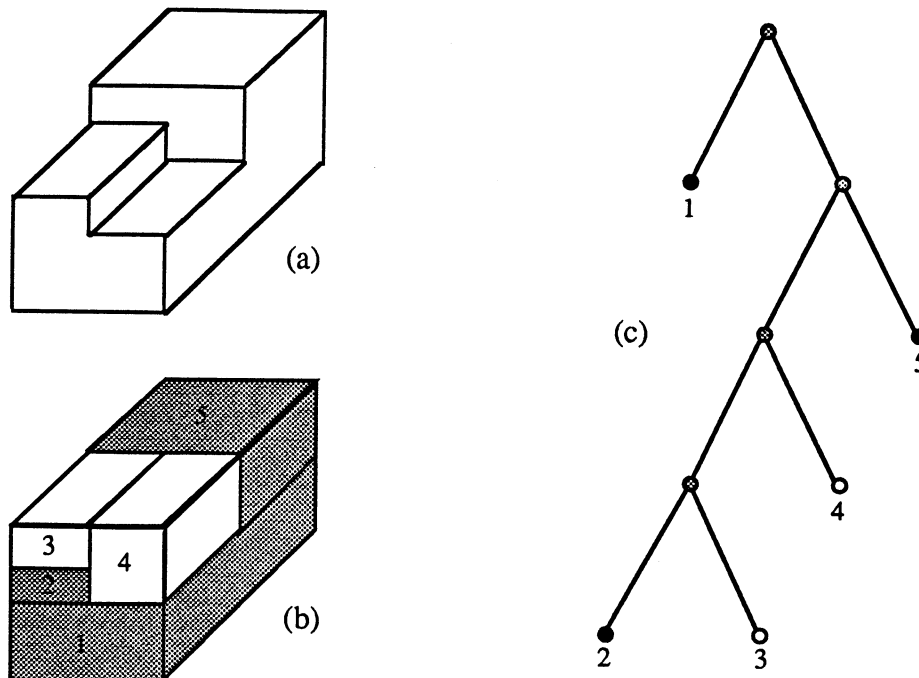


Figure 3.15. Un objet (a) et sa décomposition (b) pour construire l'arbre binaire correspondant (c).

• Les arbres multiples

Ce sont là encore des arbres construits en décomposant l'espace jusqu'à ce que chaque composante soit homogène [Carlbom 85].

Dans ce cas cinq types de cellules homogènes sont autorisés (cf. figure 3.16) :

- les cellules vides
- les cellules pleines
- les cellules *coin*
- les cellules *angle*
- les cellules *surface*.

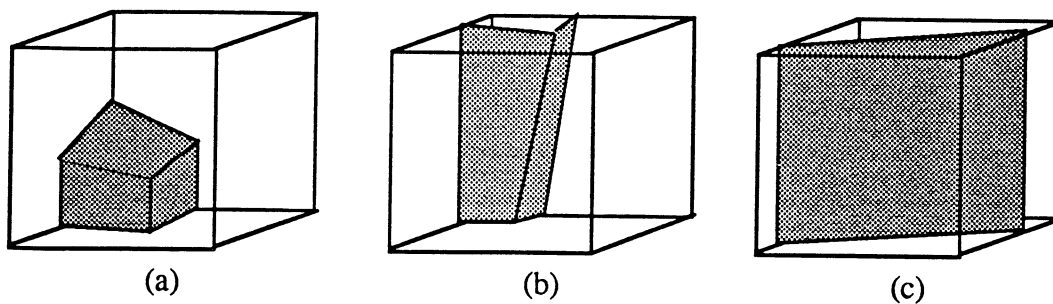


Figure 3.16. Cellule coin (a), cellule angle (b) et cellule surface (c).

Ces arbres permettent, plus facilement que les *octrees*, de passer à une structure de type facette pour représenter la surface de l'objet codé, par contre ils sont plus difficiles à calculer.

- **Les arbres syntaxiques**

D'autres structures arborescentes faisant intervenir des primitives tridimensionnelles plus ou moins complexes sont aussi utilisées [Dickinson 91]. Le large choix de primitives et les nombreuses possibilités d'organisation de celles-ci permettent à ces différentes structures de s'adapter aux applications pour lesquelles elles sont construites. Les *geons* [Levine 91], par exemple, sont tout particulièrement adaptés à la reconnaissance d'objets 3D à partir de vues 2D de ceux-ci, cela grâce à l'utilisation de primitives d'aspects dans la représentation des objets tridimensionnels.

3.2.3. Axe médian

Il est défini, de même qu'en 2D, comme étant le lieu des centres de boules maximales incluses dans l'objet . L'ensemble de ces centres forme une surface (cf. figure 3.17).

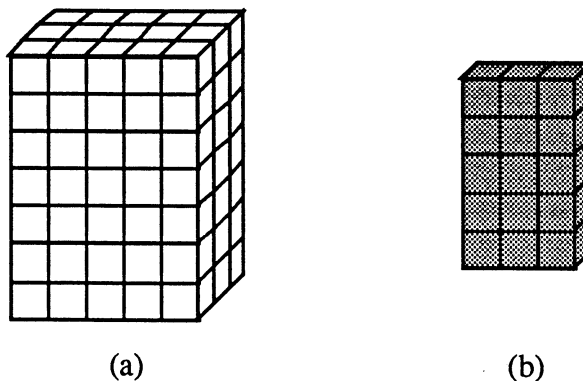


Figure 3.17. Axe médian (b) d'un parallélépipède (a).

Le codage par un axe médian définit un recouvrement de l'objet par des boules de taille maximale [O'Rourke 79].

Si, lors du calcul de l'axe médian, les tailles des boules dont les centres constituent l'axe médian sont mémorisées, la transformation est alors réversible.

Le processus de calcul de l'axe médian 3D est très simple, il est fondé sur un calcul de distance au fond suivi d'une extraction de maximums locaux. L'axe médian peut ainsi être calculé en opérant deux passages sur le volume à l'aide de masques. Les passages s'effectuant l'un après l'autre selon deux directions diagonales opposées de l'espace (*Nord Est Haut* suivi de *Sud*

Ouest Bas, par exemple). Le principe est en fait le même que celui utilisé pour le calcul de l'axe médian 2D.

Un exemple d'axe médian pour la 26-connexité est donné en annexe.

Comme nous l'avons déjà remarqué en 2D, l'axe médian 3D est déconnecté (cf. figure 3.18).

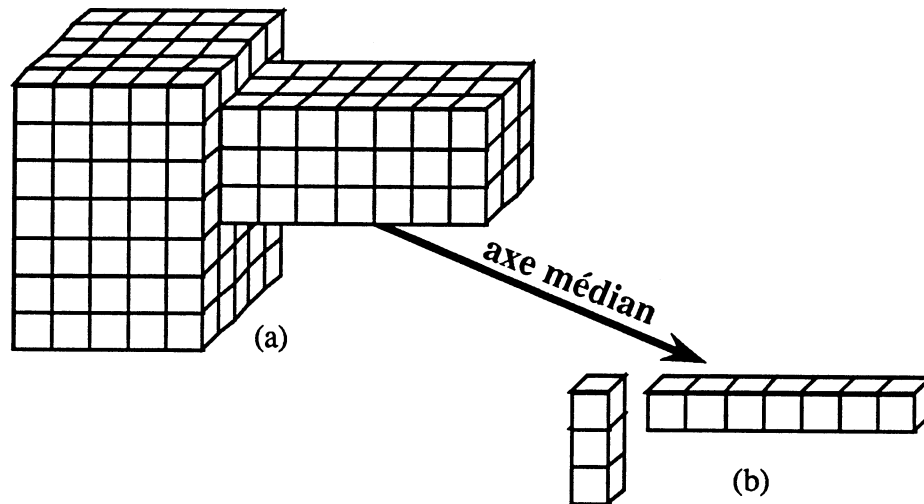


Figure 3.18. L'axe médian est déconnecté.

Une reconnexion de l'axe médian pour obtenir une ligne médiane est tentant. Une solution, fondée sur le calcul d'un squelette 3D filaire, sera proposée dans la suite de cette thèse pour calculer, à partir d'un objet, sa ligne médiane 3D (axe médian connecté).

L'axe médian est généralement mince selon au moins l'une des trois directions de l'espace. Comme on l'a déjà noté dans le plan, l'axe médian dans l'espace est parfois épais. Une méthode par sur-échantillonnage fictif est envisageable.

Malgré ces quelques inconvénients, l'axe médian est un bon moyen de représentation des volumes discrets. En effet, la réversibilité de la transformation permet, si le besoin s'en fait sentir, de revenir à une description *in extenso* du volume. Par contre, lors de manipulations, la compacité du codage est encore un atout pour l'utilisation de l'axe médian comme représentation de formes tridimensionnelles.

3.2.4. Les squelettes

3.2.4.1. Principe

L'idée générale de squelette est la même que celle donnée en 2D. Il s'agit de représenter un volume à l'aide d'une structure "mince" mais ayant, autant que possible, les mêmes propriétés topologiques. Ce modèle de représentation doit être mince selon au moins l'une des trois directions de l'espace (une épaisseur de deux voxels est permise dans le cas où l'objet est d'épaisseur paire) et centré (à un voxel près) dans l'objet.

Des méthodes semblables à celles utilisées pour le calcul de squelettes dans le plan sont employées. La plupart d'entre elles tentent de modéliser le phénomène du feu de brousse en opérant par érosions successives à partir des points bords de l'objet.

Ces méthodes diffèrent ensuite au niveau des conditions choisies pour arrêter le processus d'érosion afin que l'objet érodé possède les mêmes propriétés topologiques que l'objet initial. Les premières méthodes développées [Lobregt 80] [Toriwaki 82] procèdent par conservation du nombre d'Euler du volume considéré. Ils calculent le nombre d'Euler du volume donné et ensuite, avant l'érosion de chaque voxel, s'assurent que cette érosion ne va pas modifier le nombre d'Euler initial, donc qu'elle ne va créer ni anse ni cavité. L'objet ainsi obtenu a alors les mêmes propriétés topologiques que l'objet dont il est issu.

Le nombre d'Euler n'étant pas très facile à calculer, les algorithmes se sont peu à peu orientés vers une évaluation locale de la contribution de tel ou tel point au nombre d'Euler de l'ensemble de l'objet. A certaines de ces méthodes s'adjoignent des maillages tétradécaédriques qui semblent faciliter les choses [Hafford 84].

Finalement, la dernière génération d'algorithmes s'est développée en essayant de généraliser certains principes de squelettisation dans le plan. Ils opèrent en décidant si tel ou tel point peut ou non être érodé en examinant ses voisins et en prévoyant tous les cas où la topologie de l'objet va être modifiée [Arcelli 72] [Tsao 81] [Srisuresh 83] [Mukherjee 89] [Gong 90a] [Gong 90b] [Mukherjee 90]. C'est donc un processus strictement local qui est mis en œuvre.

Avant de donner plus précisément un algorithme de calcul, il convient de préciser exactement ce que l'on attend d'un algorithme de squelettisation 3D.

En effet, deux types de squelettes peuvent être obtenus ayant des propriétés très différentes pour l'un et pour l'autre, mais qui peuvent être complémentaires comme cela sera présenté plus

loin. A partir d'un parallélépipède veut-on obtenir une portion de plan ou un segment de droite ? (cf. figure 3.19)

En fait veut-on obtenir une surface médiane ou un squelette filaire ? L'un est mince selon l'un des axes de l'espace alors que l'autre l'est selon deux.

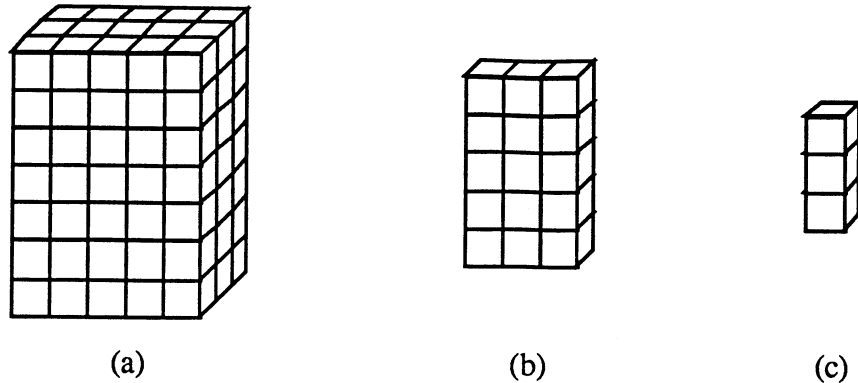


Figure 3.19. A partir d'un parallélépipède (a) veut-on obtenir un squelette surfacique (b) ou un squelette filaire (c).

Tout dépend en fait de l'utilisation envisagée du squelette.

Le squelette surfacique (*medial surface*) permet d'avoir une bonne vision de la forme ; en effet ses épaisseurs dans les différentes directions rendent exactement compte des proportions entre les différentes dimensions de l'objet initial.

De plus, une certaine réversibilité peut être obtenue si on adjoint au squelette surfacique les pondérations de l'axe médian 3D (distance au bord).

Par contre, le squelette surfacique est volumineux ; le gain de place au niveau des données est correct mais semble encore trop faible étant donnée la taille des données initiales.

Le squelette filaire permet d'avoir une bonne idée de la forme initiale au niveau topologique. Par contre, il ne permet pas de connaître les proportions de l'objet initial.

En ce qui concerne la taille, il est très économe en place mémoire étant donné qu'il ne s'agit que d'une suite d'arcs discrets alors que le squelette surfacique est composé de portions de plan (un exemple est présenté figure 3.25).

De plus le squelette filaire étant composé d'arcs un parcours de celui-ci est très facile à obtenir.

Chacun de ces deux squelettes a ses avantages ; nous verrons plus loin qu'ils ne sont pas incompatibles mais plutôt complémentaires dans certains cas.

3.2.4.2. Squelette surfacique

Je vais maintenant détailler un algorithme de calcul de squelette surfacique [Gong 90a] ; puis, à partir d'une étude des différentes conditions qu'il manipule, je présenterai les modifications que j'y ai apportées pour obtenir un algorithme de calcul de squelette filaire.

L'algorithme est fondé sur une définition et un théorème :

Définition

Un point $p \in S$ est un point *simple* si et seulement si son effacement ne modifie pas les propriétés topologiques de S .

Ce sont donc ces points simples qui vont être effacés.

Cette définition est la même que celle utilisée en 2D. Malheureusement elle ne permet pas de construire le squelette, c'est pourquoi il faut trouver des conditions locales équivalentes.

W.X. Gong et G. Bertrand démontrent que :

Théorème

Le point p est simple si et seulement si il vérifie les quatre conditions suivantes :

- (1a) • $\overline{C}_6 = 1$, où \overline{C}_6 est le nombre de composantes connexes de $\overline{S} \cap N_{26}(p)$ adjacentes (selon la 6-connexité) à p .
- (1b) • $\overline{C}_{26} = 1$, où \overline{C}_{26} est le nombre de composantes connexes de $\overline{S} \cap N_{26}(p)$ adjacentes (selon la 26-connexité) à p .
- (1c) • $NH(S \cap N'_{26}(p)) = NH(S \cap N_{26}(p))$, où $NH(A)$ est le nombre de trous dans A et $N'_{26}(p) = N_{26}(p) \cup p$.
- (1d) • $NH((\overline{S} \cap N_{26}(p)) \cup p) = NH(\overline{S} \cap N_{26}(p))$.

A partir de ce théorème l'algorithme construit est le suivant :

Algorithme

A chaque itération, appliquer l'opération T2 aux points *bord* selon la séquence de directions : H, N, E, B, S, O (correspondant au type de point bord considéré).

L'opération T2 est définie par :

T2

Effacer simultanément tous les points bords d'un type donné d qui satisfont aux conditions 2 et 3 suivantes :

- (2)
- $p_d = 0$, où p_d est le 6-voisin de p dans la direction d .
 - $P_{\bar{d}} = 1$, où $P_{\bar{d}}$ est le 6-voisin de p dans la direction opposée à d .
 - si $p_i = 0$ alors $p_l = 0$, pour tout $p_i \in N_d(p)$, et pour tout p_l 6-adjacent à p_i et à p_d , où $N_d(p) = (N_6(p) - \{p_d\} - \{P_{\bar{d}}\})$.
 - si p_i, p_j et $p_k = 0$ alors $p_m = 0$, pour tout $p_i \in N_d(p), p_j \in N_d(p), p_j$ 12-adjacent à p_i, p_k 6-adjacent à p_i et à p_j, p_m 12-adjacent à p_d et 6-adjacent à p_k .
- (3) $n \geq 8 \vee (4 \leq n \leq 7 \wedge (\exists i \in \{1, \dots, 8\}, n_i = 3))$, où n est le nombre de 1 dans $N_{26}(p)$, et n_i le nombre de 1 6-adjacents à p dans le i ème octant de $N_{26}(p)$ (un octant est un cube $2 \times 2 \times 2$, huit octants sont donc inclus dans $N_{26}(p)$).

L'algorithme ainsi décrit permet d'obtenir le squelette surfacique d'un volume (cf. figure 3.20).

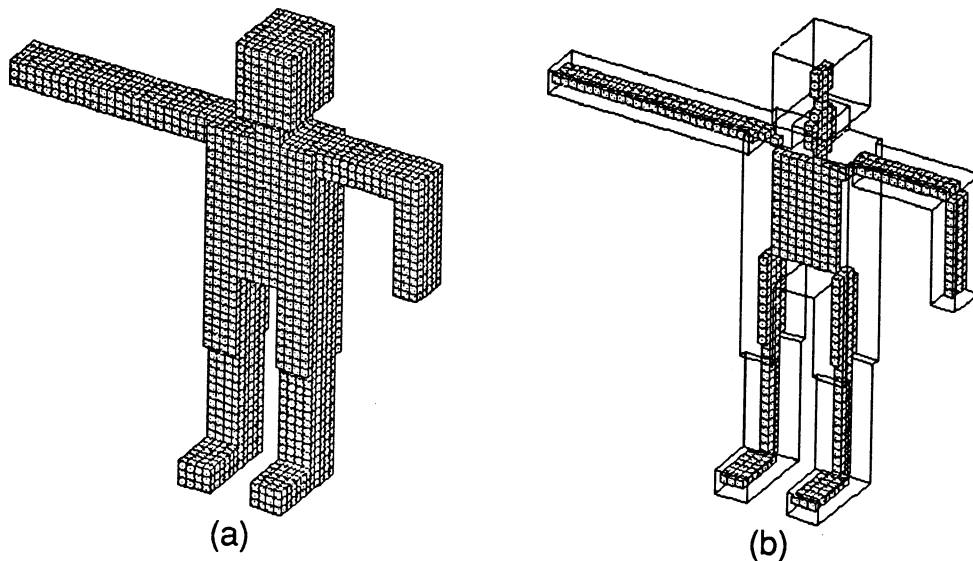


Figure 3.20. L'objet initial (a) et le squelette surfacique (b).

Une étude détaillée des conditions (2) et (3) permet de séparer les propriétés indépendamment garanties par chacune d'elles.

W. X. Gong et G. Bertrand démontrent que si un point vérifie la condition (2) alors il est un point simple. La condition (2) garantit donc la connexité du squelette obtenu et la préservation des propriétés topologiques [Gong 90a]. La condition (3) sert à limiter l'érosion.

L'exemple suivant met en évidence le rôle cette condition (3) (cf. figure 3.21).

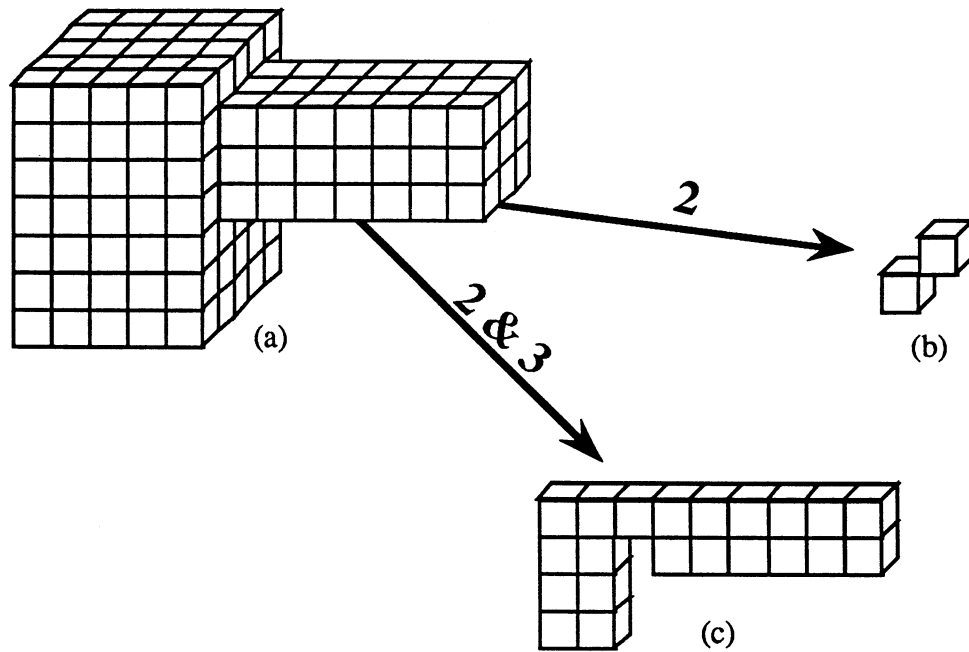


Figure 3.21. Evaluation des conditions (2) et (3).

Elle empêche que certains points ne soient considérés comme des points bords de la surface. Ainsi, n'étant pas des points bords ils ne peuvent être érodés.

Cette méthode est très satisfaisante, elle permet de conserver tous les points dès que l'épaisseur (selon l'un des trois axes de l'espace) est égale à 1. Elle préserve ainsi les proportions des épaisseurs du volume initial.

La condition (2) seule (cf. figure 3.22) se rapproche d'une érosion, procédure utilisée pour le comptage des composantes connexes.

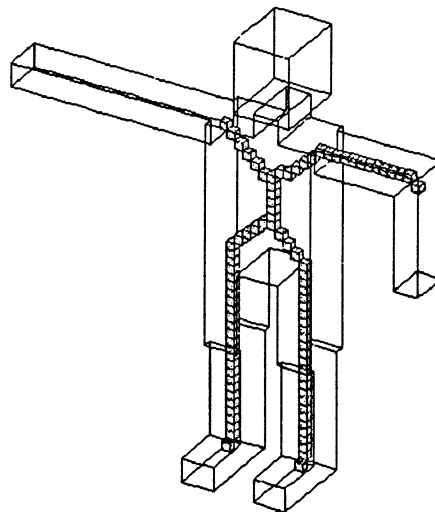


Figure 3.22. Squelette obtenu avec la condition (2) seule.

Si, par contre, on souhaite laisser se poursuivre le processus d'érosion pour amincir un peu plus (selon une direction en plus), c'est la condition (3) qu'il faut modifier.

C'est ainsi que l'on va accéder à un squelette filaire.

3.2.4.3. Squelette filaire

Le squelette filaire pourrait être obtenu à partir du squelette surfacique en opérant sur chaque portion de plan une squelettisation bidimensionnelle. Ce n'est pas cette méthode que j'ai choisie, j'ai préféré modifier la condition (3).

Si l'on veut que le processus d'érosion se poursuive il faut que la condition remplaçant (3) soit plus faible que la condition (3) elle-même. Elle doit malgré tout empêcher que le processus ne devienne trop destructif, ce qui est le cas lorsqu'on la supprime purement et simplement. Dans ce cas les extrémités se trouvent rongées (cf. figure 3.22).

La condition que j'ai proposée pour remplacer la condition (3) est la suivante [Rolland 91a] :

$$(4) \quad n > 1, \text{ où } n \text{ est le nombre de } 1 \text{ dans } N_{26}(p).$$

Cette condition (4) remplit exactement la fonction qui lui a été fixée. Elle permet en effet de conserver les points extrémités comme cela est fait en 2D.

L'algorithme ainsi construit (condition (2) et (4)) permet d'obtenir de bons résultats (cf. figure 3.23).

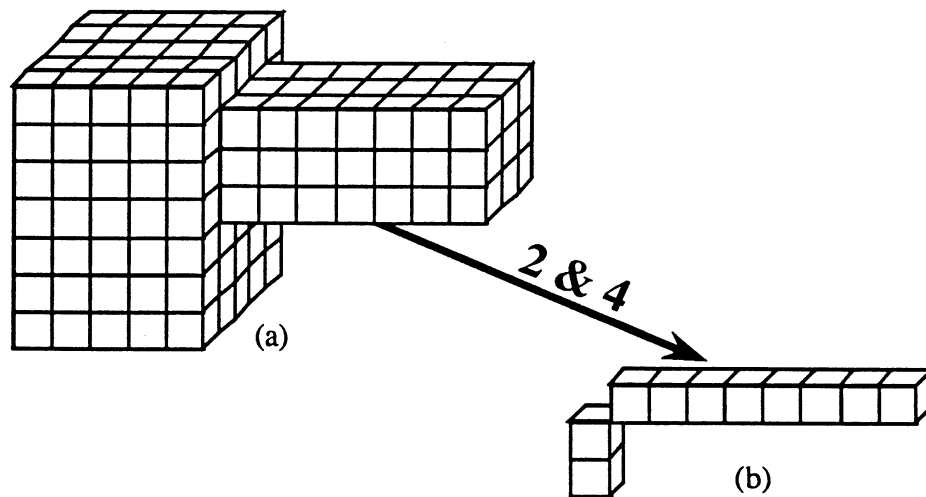


Figure 3.23. Objet tridimensionnel (a) et le squelette obtenu (b) en utilisant les conditions (2) et (4).

Les résultats obtenus sur différents exemples (cf. figure 3.24 et annexe) montrent que le squelette filaire est un bon moyen pour décrire les formes tridimensionnelles ; particulièrement bien adapté à la représentation de formes à structure tubulaire.

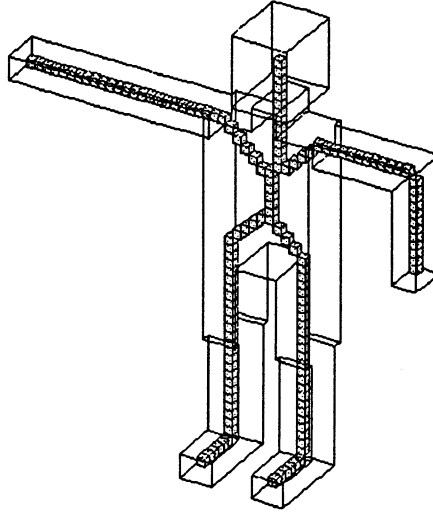


Figure 3.24. Squelette filaire

Diverses utilisations de ce squelette filaire peuvent être envisagées.

L'une des principales est la spécification d'une méthode de parcours d'un objet tridimensionnel connexe.

3.2.4.4. Parcours d'un objet 3D

Dans le plan, l'ordre de parcours d'un objet peut être défini à l'aide de son squelette. En effet, le squelette 2D est composé d'arcs, il est donc facile d'en définir un parcours. Le parcours de l'objet est défini par rapport au parcours du squelette. L'objet est ainsi parcouru perpendiculairement au squelette.

Cette méthode de parcours est d'ailleurs aussi utilisée en synthèse d'images pour créer des objets à partir d'un squelette et d'un générateur [Péroche 88].

Dans l'espace la définition du parcours nécessite l'utilisation à la fois du squelette filaire et du squelette surfacique.

Le squelette filaire est inclus dans le squelette surfacique. Ce squelette surfacique est constitué de portions de plans et d'arcs de l'espace. Le squelette filaire, lui, est un ensemble d'arcs de l'espace. Dans les portions où le squelette surfacique est une portion de plan, il contient un arc appartenant au squelette filaire. Le parcours du squelette surfacique se fait donc, en parcourant le squelette filaire, de manière perpendiculaire à ce dernier (cf. figure 3.25).

Ensuite le parcours de l'objet se fait de manière analogue, et ce, perpendiculairement au squelette surfacique.

Le parcours de l'objet se fait donc portion de plan par portion de plan, chacune de ces portions de plan est définie par un arc du squelette surfacique et par la perpendiculaire au squelette surfacique.

Chaque arc du squelette surfacique est défini par un point du squelette filaire et la perpendiculaire au squelette filaire en ce point (cf. figure 3.25).

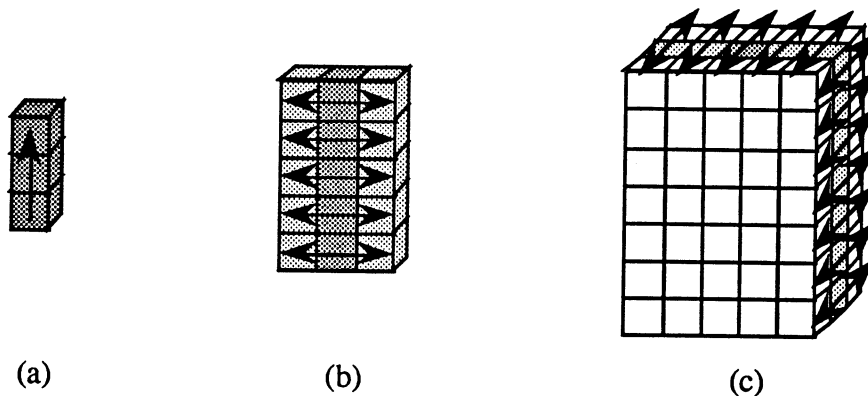


Figure 3.25. Parcours d'un volume (c) à l'aide du parcours de son squelette surfacique (b) défini lui-même par le parcours du squelette filaire (a).

Le squelette filaire peut aussi être utilisé pour faire une description qualitative de la forme de l'objet.

Il est parfois plus facile de se rendre compte de la forme d'un objet à partir de son squelette qu'à partir de l'objet dans son ensemble. En effet, lorsque le volume est visualisé dans sa totalité certaines parties peuvent en cacher d'autres ce qui rend la compréhension difficile ; par contre lorsque seul le squelette filaire est affiché sur l'écran, de par le fait qu'il est filaire, il y a peu de chance qu'une partie de celui-ci soit dissimulée par une autre (cf. annexe).

3.2.4.5. Reconnexion de l'axe médian

L'axe médian, comme cela a été déjà remarqué, n'est pas nécessairement connexe ni dans le plan ni dans l'espace. Dans le plan plusieurs méthodes de reconnexion de l'axe médian existent [Pavlidis 80] [Arcelli 81] [Arcelli 85] [Montanvert 85]. Malheureusement elles ne sont que très difficilement transposables dans l'espace. Dans le plan, cette reconnexion s'effectue généralement en recherchant des chemins de connexion entre les différents points de l'axe médian. Dans l'espace plusieurs problèmes se posent. Tout d'abord, il faut arriver à déterminer

quels points des différentes composantes connexes de l'axe médian doivent être reliés ; ensuite, il faut arriver à orienter correctement ces chemins de connexion.

Ces deux problèmes semblent assez compliqués à résoudre dans certains cas (quels points, à relier, choisir lorsque deux composantes sont parallèles, quel chemin choisir parmi les nombreuses possibilités offertes).

Les squelettes filaires apportent une solution au problème de la connexion de l'axe médian.

En effet, en appliquant, à un volume initial donné, l'algorithme de squelettisation (conditions (2) et (4)) et en garantissant que les points de l'axe médian ne sont pas érodés, on obtient une ligne médiane.

Pour cela, il suffit d'ajouter aux conditions (2) et (4) de l'algorithme la condition suivante :

$$(5) \quad p \notin \text{AXM}(S), \text{ où } \text{AXM}(S) \text{ est l'axe médian de } S.$$

L'algorithme ainsi construit permet de connecter l'axe médian (cf. figure 3.26).

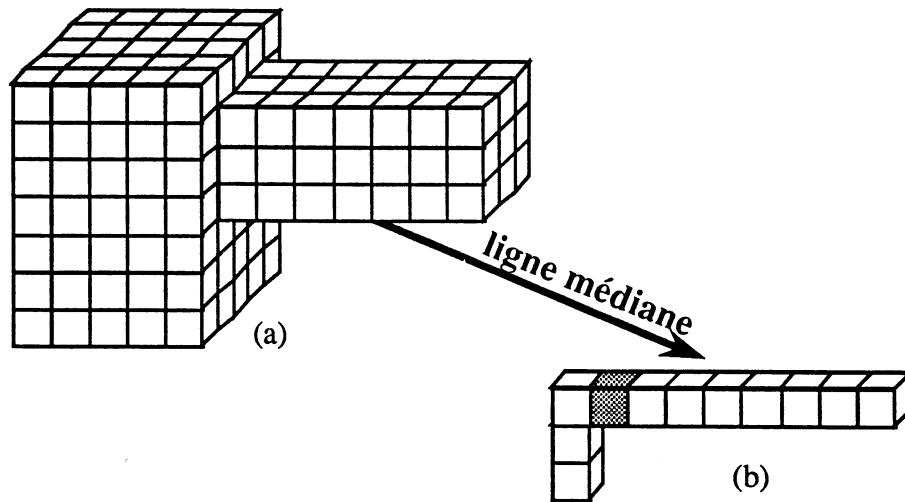


Figure 3.26. Connexion de l'axe médian pour obtenir une ligne médiane.

En étudiant les différentes conditions utilisées pour le calcul de la ligne médiane ((2), (4) et (5)) on se rend compte que les conditions (2) et (5) suffisent pour obtenir la connexion de l'axe médian.

En effet, la condition (4) avait été ajoutée afin que les extrémités des "branches" ne soient pas rongées. Elle est donc inutile ici puisque, l'axe médian assurant la réversibilité et les points de l'axe médian ne pouvant être érodés, les extrémités ne pourront pas être abusivement érodées.

La condition (4) est plus qu'inutile puisqu'elle génère parfois des lignes médianes un peu trop longues (cf. figure 3.27).

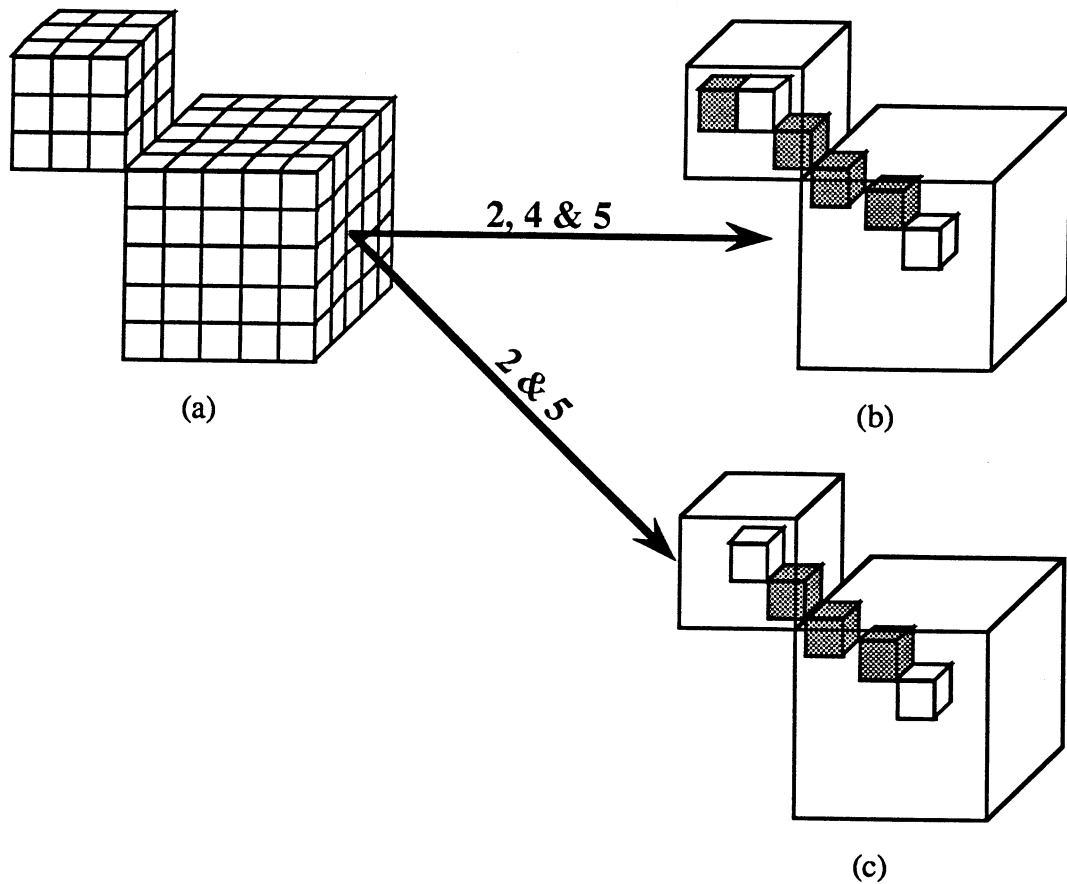


Figure 3.27. Exemple de ligne médiane un peu trop longue. L'objet 3D initial (a), la ligne médiane calculée avec les conditions 2, 4 & 5 (b) (elle est trop longue) et la ligne médiane calculée avec les conditions 2 & 5 (c).

L'algorithme finalement retenu pour le calcul de la ligne médiane utilise donc les conditions (2) et (5). La condition (5) assure la conservation de l'axe médian et la condition (2) garantit la connexion des différentes parties de l'axe médian.

Les résultats obtenus par cette méthode sont très satisfaisants (cf. figure 3.28. et annexe).

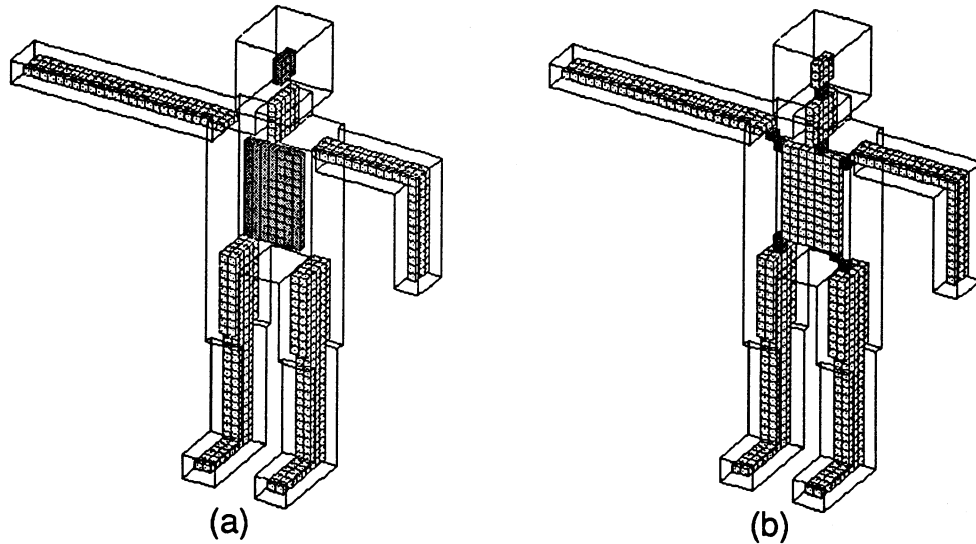


Figure 3.28. Axe médian (a) et ligne médiane (b).

3.2.4.6. Les squelettes en tant que moyen de codage

Les squelettes filaires peuvent donc être considérés comme de bons descripteurs de formes 3D. Il convient cependant de prendre garde au fait que la réversibilité du procédé de codage par squelette simple (sans mémorisation des poids de l'axe médian) n'est pas assurée.

Il faut en fait bien distinguer deux choses : la signification visuelle et la réversibilité.

La signification visuelle du squelette permet, à elle seule, d'appréhender la structure spatiale de l'objet 3D. La réversibilité ne peut, elle, être obtenue qu'en utilisant à la fois le squelette et l'axe médian (muni des distances au fond).

C'est à partir d'un exemple que les pouvoirs de codage et de réversibilité sont analysés.

La figure 3.29 présente une synthèse des performances des différentes méthodes de codage ainsi que certaines valeurs représentant les pouvoirs de codages et de réversibilité de ceux-ci.

L'objet codé est le volume présenté en annexe ("bonhomme").

On remarque que le meilleur taux de réduction de données est obtenu par le squelette dont les extrémités sont rongées (condition (2) seule), par contre dans ce cas le taux de réversibilité est très faible ; on perd donc une bonne partie de l'information véhiculée par le volume.

Dans le cas où on utilise le squelette surfacique (conditions (2) et (3)), la réversibilité est presque totale (seuls certains points de la surface ont disparu), la réduction du volume des données est correcte, sans plus.

En ce qui concerne le squelette filaire (conditions (2) et (4)) le taux de réduction est très bon, le taux de réversibilité n'est pas très élevé mais on se rend compte malgré tout que la forme est

bien conservée. Les points ayant disparus sont des points de la surface, ils sont donc nombreux (ce qui explique le faible taux) mais pas très importants en ce qui concerne l'allure générale de l'objet.

Bien sûr, si l'on souhaite une réversibilité totale il faut s'orienter vers des codages par axe médian ou ligne médiane. La ligne médiane ne comporte qu'un peu plus de points que l'axe médian et a le grand avantage d'être connexe ce qui implique une plus grande facilité de description et de parcours.

Les deux codages les plus intéressants sont donc le squelette filaire et la ligne médiane.

<i>codage</i>	nombre de voxels	taux de réduction	nombre de voxels de l'objet reconstruit	taux de réversibilité
squelette surfacique (conditions 2 & 3)	475	89 %	3913	90 %
squelette en n'utilisant que la condition 2	90	98 %	2034	47 %
squelette filaire (conditions 2 & 4)	148	96 %	2907	67 %
axe médian 3D	672	84 %	4322	100 %
ligne médiane 3D (conditions 2 & 5)	684	84 %	4322	100 %

Figure 3.29. Différentes méthodes de codage appliquées à un objet composé de 4322 voxels.

3.2.5. Liens entre modèles bidimensionnels et modèles tridimensionnels

Après avoir donné un aperçu des différents modèles de représentation tant en 2D qu'en 3D, il convient de s'interroger sur les relations existantes entre les modèles bidimensionnels et les modèles tridimensionnels.

L'analyse de ces relations permet de bien situer les problèmes rencontrés lorsqu'on souhaite passer de formes bidimensionnelles à une forme tridimensionnelle (c'est exactement ce que l'on souhaite faire dans le cadre de la reconstruction tridimensionnelle à partir de coupes sériées).

Bien entendu ces relations ne peuvent s'exprimer qu'entre des modèles relativement semblables.

Les premières structures qui semblent permettre le développement de telles relations sont les contours en 2D et les surfaces en 3D. Il y a une analogie certaine entre ces deux modes de représentation. Malgré cela il semble assez difficile de passer de l'une à l'autre. Comment peut-on passer d'un ensemble de contours 2D à une surface 3D ? Dans l'esprit ce passage semble simple. Il "suffit", en effet, de faire passer une surface par tous les contours donnés.

Le principe est simple mais sa réalisation est ardue.

En effet, si par exemple les contours 2D sont donnés par des approximations polygonales on a envie de construire la surface à l'aide de facettes. Malheureusement le procédé de triangulation n'est pas très facile à mettre en œuvre et en plus il ne donne pas toujours le résultat escompté.

Les seules méthodes semblant permettre le passage de courbes 2D à des surfaces 3D de manière assez facile sont les méthodes utilisant des approximations des contours et de la surface par des splines [Wu 77]. Malheureusement ces méthodes étant très calculatoires, elles ne peuvent être mise en œuvre, d'après leurs auteurs, qu'à l'aide de calculs considérables qui ne peuvent être effectués, dans des délais acceptables, que dans des cas relativement simples. Par contre, une fois les résultats obtenus ceux-ci sont vraiment très satisfaisants.

Le passage inverse, de la surface 3D à des coupes 2D, bien que plus simple, ne l'est pas toujours autant qu'on veut bien le dire. En effet, pour effectuer, par exemple, des coupes dans un objet représenté par facettes, il faut déterminer tout d'abord l'ensemble des facettes qui vont être coupées, ensuite il faut calculer les intersections entre les différentes facettes et le plan de coupe ; le tout en essayant de rester dans le domaine discret pour ne pas avoir d'approximations, et en tentant d'obtenir, si possible directement, une représentation correcte du contour (codage de Freeman ou approximation polygonale par exemple).

Le deuxième type de données permettant d'analyser les relations entre les représentations 2D et 3D sont les arbres. Bien sûr ces arbres doivent être construits suivant le même principe. C'est le cas avec les *quadrees* et les *octrees*.

Le passage d'un ensemble de *quadrees* représentant des coupes à un *octree* représentant l'objet est assez complexe à mettre en œuvre. En effet, les notions de voisinage n'étant que très

difficiles à obtenir, le procédé est très difficile à réaliser. Aucune méthode ne semble à ce jour avoir résolu le problème.

Par contre une méthode existe lorsqu'il s'agit de calculer le *quadtree* de coupes d'un *octree*, que les coupes soient perpendiculaires à l'un des axes utilisés ou non [Yau 84] [Chen H.H. 88]. Cette méthode procède en déterminant, lors d'un parcours de l'*octree*, les intersections entre les octants et les quadrants pour permettre la construction du *quadtree*.

En ce qui concerne les méthodes utilisant une représentation par squelette, là non plus le problème n'est pas simple à résoudre (cf. figure 3.30).

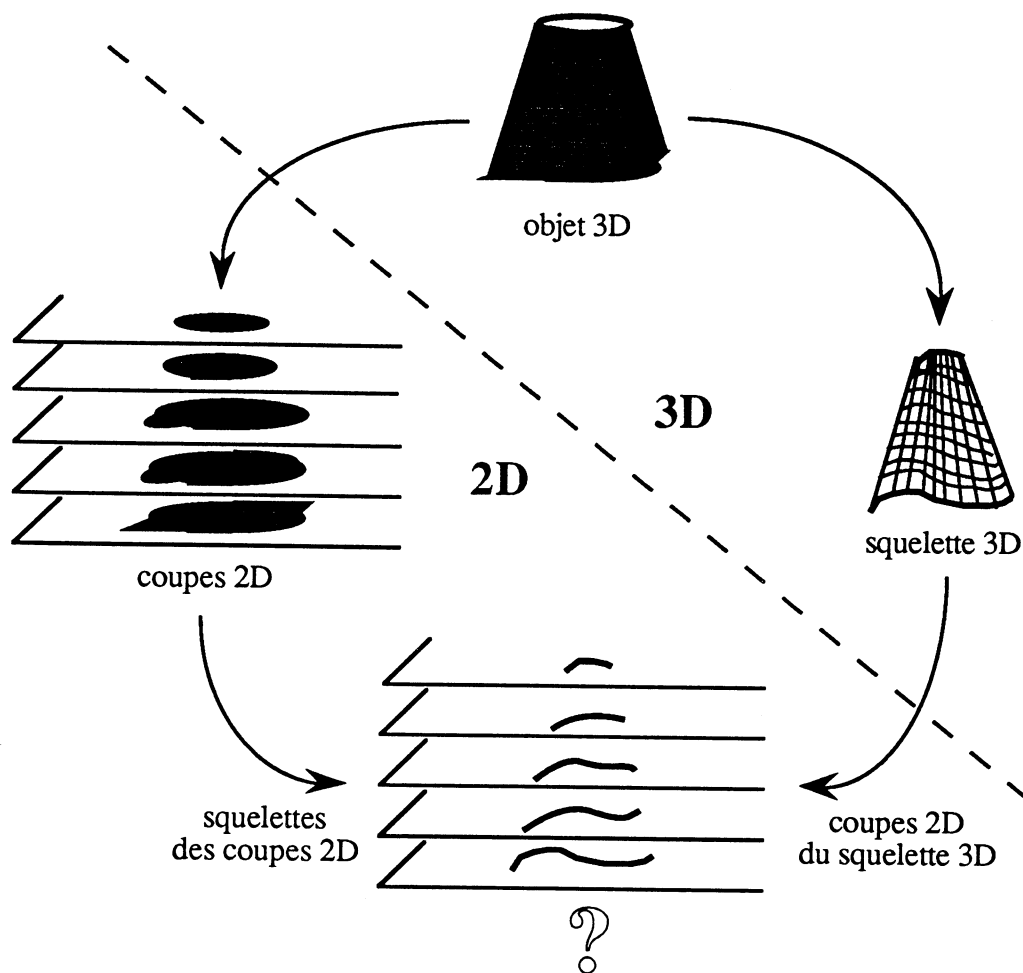


Figure 3.30. Squelettes de coupes 2D et coupes 2D de squelette 3D.

Ceci est en partie lié au fait que la structure même du squelette n'est pas conservée lorsqu'on passe du plan à l'espace ou de l'espace au plan. En effet, il n'y a que très peu de ressemblances entre le squelette 2D d'une coupe d'un objet 3D et une coupe 2D du squelette 3D de ce même

objet. En particulier, la coupe d'un squelette 3D n'est pas nécessairement connexe alors que le squelette 2D d'une coupe l'est par définition (certains points peuvent ne pas être nécessaires à la connexion dans l'espace alors qu'ils le deviennent dans le plan) (cf. figure 3.31).

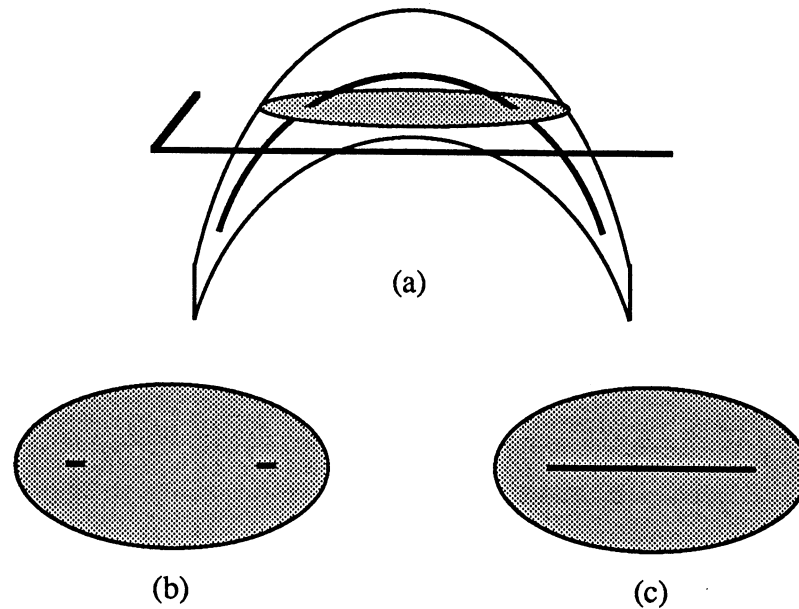


Figure 3.31. La coupe 2D (b) du squelette d'un objet (a) peut être différente du squelette 2D (c) de cette même coupe.

Dans le cas où l'étude de ses représentations par squelettes 2D et 3D aurait permis d'obtenir des résultats intéressants, ceux-ci auraient pu être mis à profit pour réduire l'inférence de coupes intermédiaires à celle des squelettes. Des propriétés qui seraient exploitables pour la reconstruction tridimensionnelle à partir de coupes sériées ne semblent pouvoir être trouvées que dans les cas d'objets convexes, ce qui réduit considérablement leur champ d'application.

Pour ce qui est des modèles par axe médian, des problèmes semblables à ceux rencontrés pour les squelettes apparaissent.

Les seules méthodes qui devraient en fait permettre de passer des modèles de représentations 2D à des modèles de représentations 3D sont les méthodes procédant par inférence sur tout l'objet. Ce sont ces méthodes que je développe dans la suite de cette thèse. Certaines seront juste présentées, celle que j'ai développée étant beaucoup plus détaillée.

Chapitre 4

Inférence de formes pour la reconstruction tridimensionnelle

Je présente dans ce chapitre les différentes méthodes de reconstruction tridimensionnelle utilisant l'inférence de formes.

Ces méthodes sont fondées sur une idée commune : un volume 3D peut être considéré comme un empilement de coupes 2D.

Ainsi, si le nombre de coupes données est suffisamment grand, le processus de reconstruction consiste en un simple empilement de ces coupes.

Malheureusement, si deux coupes successives ne sont pas très ressemblantes l'une avec l'autre, lors de l'empilement une forte discontinuité va apparaître dans le volume ; une "marche d'escalier" va rompre la "continuité" de l'objet sur son pourtour.

Ce cas se produit très souvent, entre autres parce que les systèmes optiques d'acquisition ne permettent pas d'avoir la même résolution dans le plan de coupe et perpendiculairement à celui-ci. Dans le meilleur des cas, c'est-à-dire en microscopie confocale à balayage laser en fluorescence, le rapport entre ces deux résolutions est de l'ordre de trois. Pour la plupart des autres systèmes d'acquisition le rapport est plus souvent de l'ordre de quinze ou vingt (30 coupes pour des images 512 x 512). Il est donc très important de pouvoir intercaler un nombre suffisant de coupes, entre deux coupes données, pour que les différentes coupes soient

suffisamment semblables les unes avec les autres et qu'ainsi aucune discontinuité ne persiste. Les coupes à intercaler doivent donc être calculées à partir des coupes données.

Dans la suite, ces coupes sont appelées coupes intermédiaires. Les différentes méthodes de reconstruction tridimensionnelle procédant par inférence de formes ont pour objectif de calculer les coupes intermédiaires à partir de deux coupes extrémales.

Il s'agit bien d'un processus d'inférence, puisque c'est à partir des analyses des coupes initiales que sont calculées les coupes intermédiaires.

Dans la suite de cette thèse, on préférera employer le terme d'inférence plutôt que le terme d'interpolation. En effet, dans certains cas, on peut être amené à inférer des coupes à l'extérieur des coupes données ; dans ce cas on peut encore parler d'inférence alors qu'il faudrait déjà parler d'extrapolation. Ce cas se reproduit chaque fois que, à partir de deux coupes dont l'une est soit la première soit la dernière des coupes données, les descriptions des coupes sont telles qu'on peut considérer que le volume se ferme après la coupe extrémale considérée.

La première méthode qui vient à l'esprit pour effectuer l'inférence de formes consiste à faire des sections dans un objet construit par facettisation. Comme on l'a vu cette méthode de facettisation n'est pas très simple et ne donne pas toujours de très bons résultats. De plus, si on résout le problème de la reconstruction tridimensionnelle en utilisant une facettisation il est inutile de développer une méthode d'inférence de formes. C'est pourquoi cette méthode peut être qualifiée de "pseudo-méthode" d'inférence de formes.

En fait, deux véritables méthodes d'inférence de formes s'affrontent dans le domaine de la reconstruction tridimensionnelle à partir de coupes sériées. L'une utilise une approche par les contours des coupes bidimensionnelles données, l'autre effectue une approche par les régions (par blocs) de ces mêmes coupes. Les représentations manipulées par ces deux méthodes étant fondamentalement différentes, leur mise en œuvre l'est également.

4.1. Méthode des modèles élastiques

Cette méthode utilise une représentation des différentes coupes bidimensionnelles par leurs contours. Elle fait partie des techniques employant des "contours actifs", largement utilisées dans les différents domaines de l'analyse d'images.

Il s'agit ici, dans un premier temps au moins, d'effectuer une interpolation de contours.

La méthode présentée ici [Chen J.S. 88] [Lin 89] consiste à interpoler des contours à partir des contours des coupes données. Le principe a été proposé par D. J. Burr [Burr 81].

La méthode se propose de générer des coupes intermédiaires à l'aide d'une technique d'assortiment élastique.

Tout au long de cette phase, le travail consiste à calculer des coupes intermédiaires entre un contour source et contour but. Chacun de ces contours est représenté à l'aide d'une approximation polygonale.

Le principe mis en œuvre tente de modéliser le champ de force, qui, appliqué sur le contour source, permet d'obtenir le contour but.

Ce champ de force est en fait une famille de vecteurs de force ; chaque vecteur de force est défini comme une force d'attraction ou de répulsion.

Le contour source est représenté par C_s avec :

$$C_s = \{(x_i, y_i), 0 \leq i < N_s\}$$

où N_s est le nombre de sommets de l'approximation polygonale du contour source.

Le i -ième sommet de C_s est noté $p_s(i)$.

Le contour but est défini de manière identique par :

$$C_b = \{(x'_j, y'_j), 0 \leq j < N_b\}$$

où N_b est le nombre de sommets de l'approximation polygonale du contour but.

Le j -ième sommet de C_b est noté $p_b(j)$.

Dans le cas où $N_s \geq N_b$ le champ de force est calculé à l'aide de la formule suivante :

$$DS(x, y) = \frac{1}{\gamma} \left[\frac{\sum_{i=0}^{N_s-1} G_s^i \cdot D_s(i, J_i)}{\sum_{i=0}^{N_s-1} G_s^i} - \frac{\sum_{j=0}^{N_b-1} G_b^j \cdot D_b(j, I_j)}{\sum_{j=0}^{N_b-1} G_b^j} \right]$$

avec
$$G_s^i = \exp \frac{-(x - x_i)^2 - (y - y_i)^2}{\sigma_k^2}$$

et
$$G_b^j = \exp \frac{-(x - x'_j - D_{b_x}(j, I_j))^2 - (y - y'_j - D_{b_y}(j, I_j))^2}{\sigma_k^2}$$

G_s^i et G_b^j sont des facteurs de pondération permettant d'accentuer l'influence des proches voisins par rapport à celle des voisins plus éloignés.

$D_s(i, J_j)$ est un vecteur de déplacement défini comme étant le vecteur le plus court reliant le sommet $p_s(i) \in C_s$ au segment dont les extrémités sont $p_b(J_j)$ et $p_b(J_{j+1}) \in C_b$.

$D_b(j, I_j)$ est défini de manière analogue en inversant les rôles des contours source et but.

Le paramètre γ représente un facteur d'amortissement. Le paramètre σ_k exprime la "raideur" utilisée à l'itération k . De manière empirique il semble qu'il faille que ce paramètre décroisse au fur et à mesure que les itérations se produisent :

$$\sigma_k = \sigma_0 \cdot f^{-k} \text{ avec } f \in [1, 2].$$

A partir des équations précédentes, tous les contours intermédiaires peuvent être calculés. Ils sont indexés de 1 à k et obtenus en appliquant la formule :

$$W^k(i) = W^{k-1}(i) + DS^{k-1}(x_i^{k-1}, y_i^{k-1})$$

Le processus de calcul s'interrompt lorsque, étant donné un réel ϵ petit on a :

$$\frac{1}{N_s} \sum_{i=0}^{N_s-1} |DS^k(x_i, y_i)| < \epsilon$$

Malheureusement cette méthode ne permet pas de résoudre le problème de l'inférence de contours dans le cas où apparaissent des branches, les forces produites empêchant la déconnexion d'un ensemble de sommets.

Pour résoudre ce problème il faut modifier les facteurs de pondération G_s^i et G_b^j comme suit :

$$G_s^i = \begin{cases} 0, & \text{si } m_1 > t \\ \exp \frac{-(x - x_i)^2 - (y - y_i)^2}{\sigma_k^2}, & \text{si } m_1 \leq t \end{cases}$$

avec

$$m_1 = \frac{|D_s(i, J_i)|}{\left[\frac{\sum_{i=0}^{N_s-1} |D_s(i, J_i)|}{N_s} \right]}$$

et

$$G_b^j = \begin{cases} 0, & \text{si } m_2 > t \\ \exp \frac{-(x - x'_j - D_{bx}(j, I_j))^2 - (y - y'_j - D_{by}(j, I_j))^2}{\sigma_k^2}, & \text{si } m_2 \leq t \end{cases}$$

avec

$$m_2 = \frac{|D_b(j, I_j)|}{\left[\frac{\sum_{j=0}^{N_b-1} |D_b(j, I_j)|}{N_b} \right]}$$

t est un seuil fixé par l'utilisateur.

Ces nouvelles valeurs permettent de résoudre ce problème de branches en feignant d'ignorer les forces qui sont devenues trop fortes. Cela permet à certains contours de se déplacer alors que d'autres restent à la même place.

Ainsi on obtient, à partir des contours initiaux, des contours intermédiaires. Si ces contours intermédiaires sont assez ressemblants entre eux (et avec les contours initiaux) la méthode s'arrête là, il suffit alors d'empiler les contours pour obtenir le volume reconstruit.

Dans certains cas, il peut être intéressant de modéliser la surface de l'objet reconstruit. Cela est fait à l'aide de fonctions splines [Chen J.S. 88] [Lin 89]. Pour pouvoir effectuer cette modélisation Chen et Lin utilisent une représentation paramétrique des contours qui permet de transformer l'ensemble des contours en une grille, grille qui sert ensuite à calculer les splines utilisées pour la reconstruction proprement dite.

Cette méthode d'assortiment élastique est tout à fait adaptée à la résolution du problème de la reconstruction tridimensionnelle à partir de coupes sériées.

En effet, le procédé consistant à déformer les contours correspond assez bien à ce que l'imagination tend naturellement à faire (cf. figure 4.1). Malheureusement, la modélisation du processus est assez lourde car elle nécessite l'utilisation de nombreuses équations (champ de force, déformation élastique, etc).

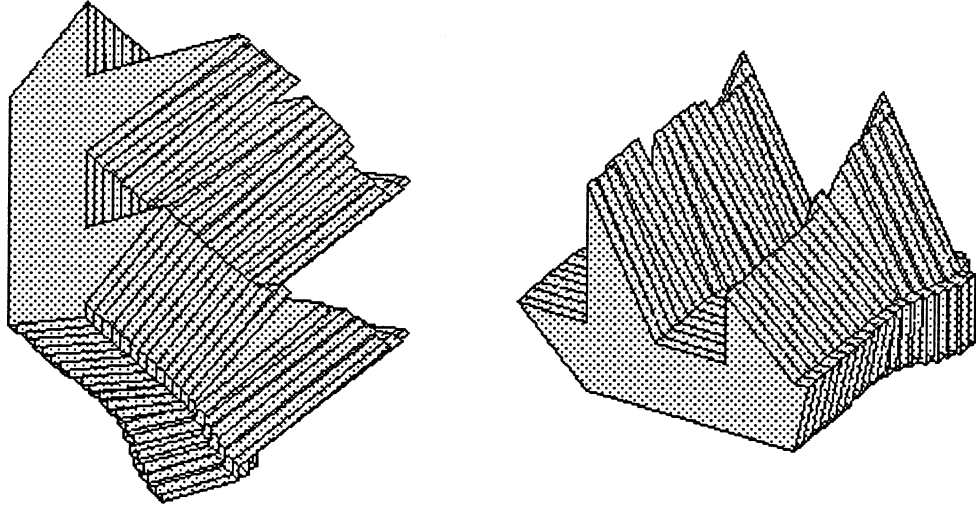


Figure 4.1. Exemple de résultats obtenus par assortiment élastique.
Deux visualisations différentes du même objet.

De plus cette modélisation nécessite l'introduction de divers paramètres ($\gamma, f, \varepsilon, \dots$), ceux-ci n'étant pas très faciles à contrôler. Ces paramètres étant très importants lors des calculs, un mauvais choix peut parfois aboutir à des résultats tout à fait surprenants.

On remarque aussi, lors de l'utilisation de la méthode, l'apparition de phénomènes typiquement physiques tels que les oscillations. Ces phénomènes apparaissent selon les valeurs des paramètres, plus ou moins souvent, et avec plus ou moins de répercussions sur les résultats obtenus.

La méthode par assortiment élastique est donc assez bien adaptée à la résolution de notre problème mais s'avère assez difficile à mettre en œuvre pour l'obtention de résultats très probants.

La méthode que j'ai choisie pour essayer de résoudre ce problème de reconstruction tridimensionnelle à partir de coupes sériées est assez différente car fondée sur une représentation par régions des différentes coupes.

4.2. Méthode utilisant un modèle hiérarchique

J'ai délibérément choisi d'orienter ma méthode, contrairement à celles existantes, vers une approche utilisant uniquement des représentations internes des différentes composantes du problème.

La plupart des méthodes s'appuient sur les contours des différentes coupes. Ainsi elles se décomposent en deux principales phases. La première consiste à extraire les contours des différentes coupes, la seconde à effectuer, à partir de ces contours, un assortiment élastique ou une triangulation. En cas de triangulation, on peut être amené à calculer localement des portions de contours intermédiaires (cf. figure 4.2).

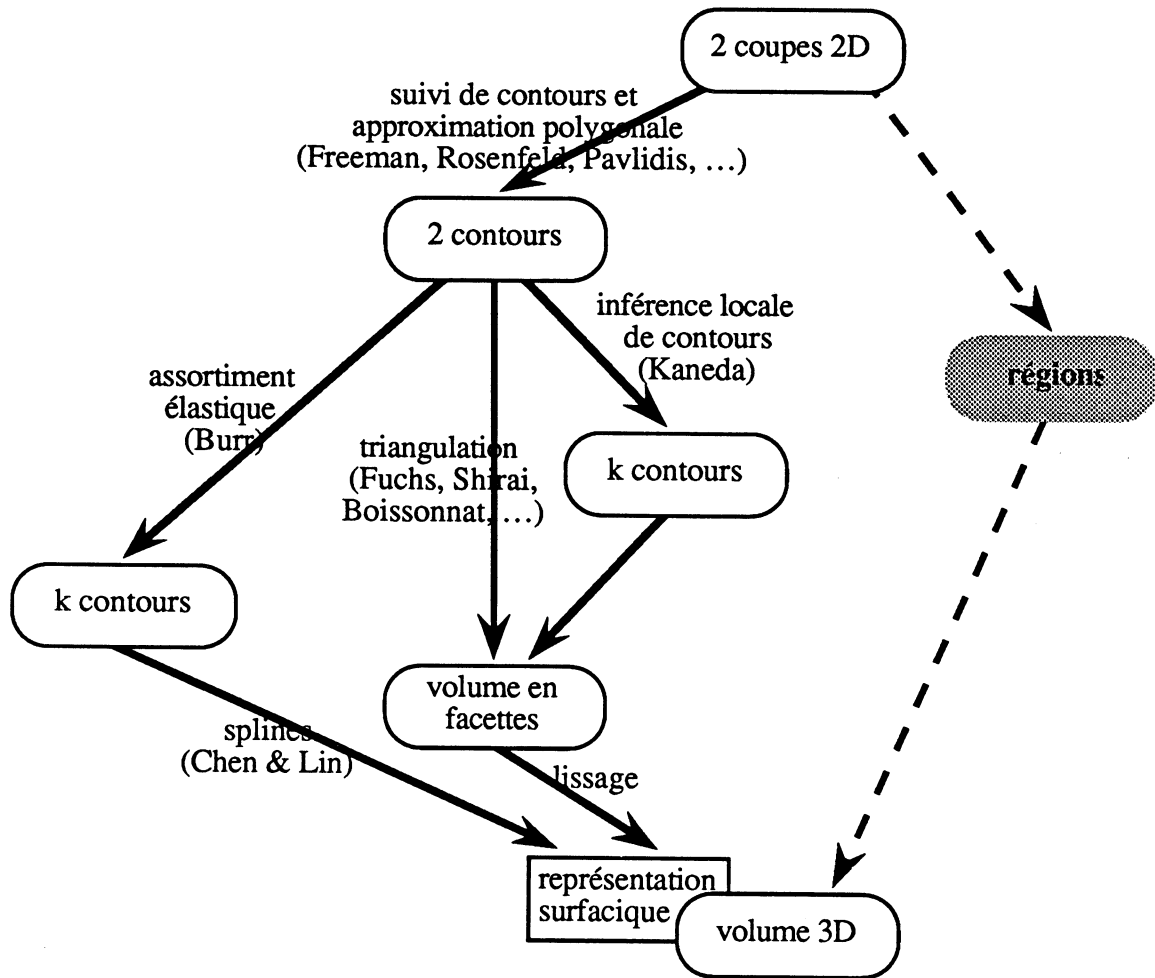


Figure 4.2. Différentes approches par les contours.

La méthode que j'ai développée s'appuie sur une représentation hiérarchique des régions représentant les différentes coupes données. Le modèle de représentation des coupes 2D choisi est le graphe de la ligne médiane. Ce graphe de la ligne médiane est une représentation hiérarchique du squelette pondéré qui permet de décrire et manipuler la forme qu'il code (cf. § 2.2.6).

Le volume 3D ne sera, lui non plus, pas abordé par les contours ; une représentation par voxels sera tout d'abord mise en œuvre.

La technique développée dans la suite de cette thèse est basée sur le principe suivant. A partir de deux coupes données, on calcule pour chacune d'elles un graphe de la ligne médiane. C'est ensuite à partir de ces graphes que tout le travail est réalisé. Dans un premier temps il s'agit d'inférer des graphes de la ligne médiane intermédiaires. A partir de ces graphes les coupes intermédiaires sont régénérées puis rendues épaisses pour pouvoir être empilées afin de rendre un volume représenté en voxels (cf. figure 4.3).

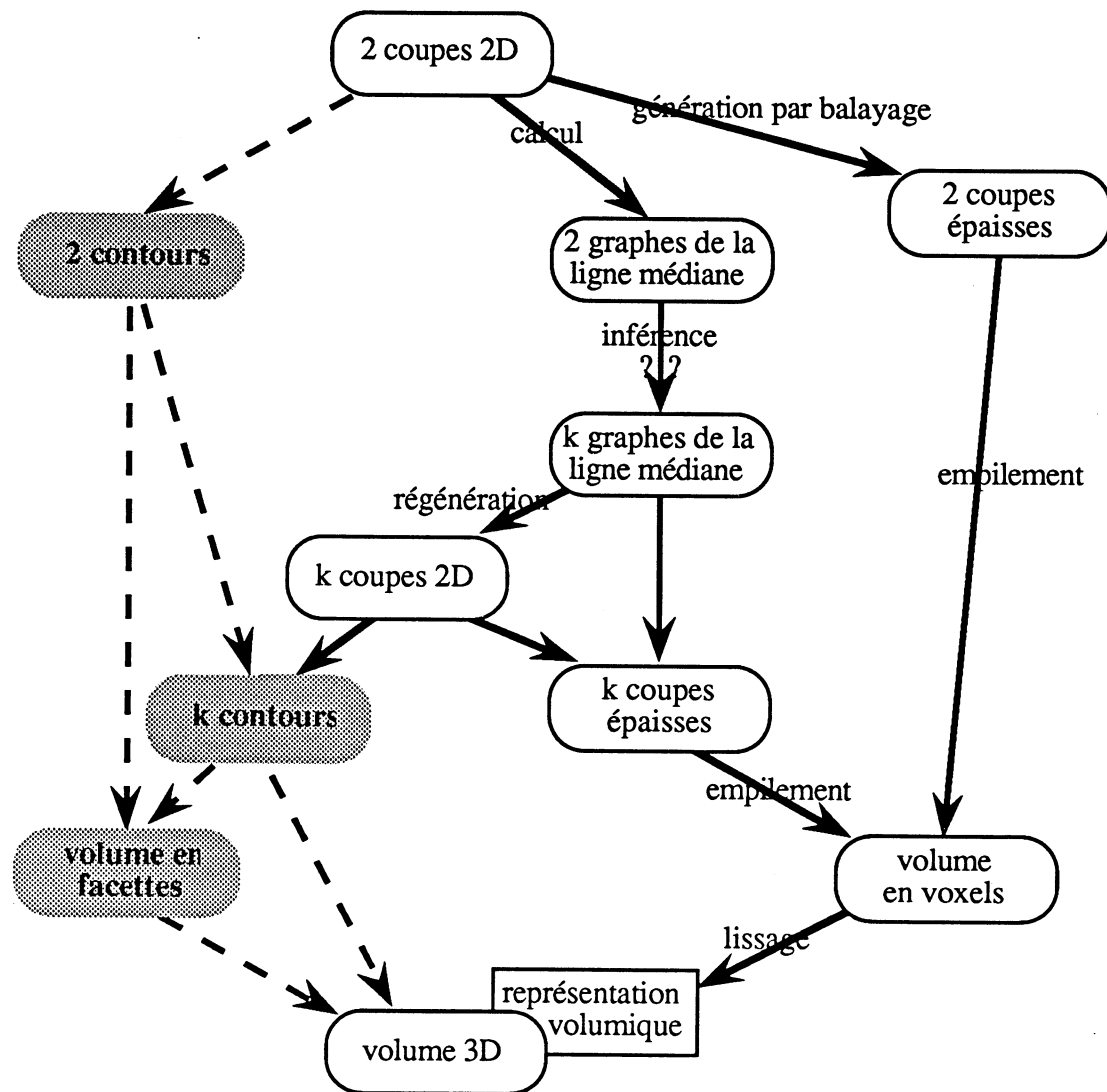


Figure 4.3. Approche par les régions choisies.

On remarque que la méthode choisie permet, si on le souhaite, de revenir très facilement à une approche de type contour (k coupes 2D donnent k contours), ceci à des fins de représentation, par exemple.

La clé de voûte de la méthode est constituée par l'inférence de graphes de la ligne médiane.

Cette inférence a été développée, dans un premier temps, de manière assez empirique en essayant de trouver de manière heuristique quel graphe de la ligne médiane peut être considéré comme intermédiaire entre deux autres [Rolland 89a] [Rolland 89b].

La méthode que je présente dans le chapitre suivant est plus systématique. Elle s'appuie sur des résultats théoriques qui permettent d'espérer des résultats encore meilleurs.

J'ai décomposé cette inférence de graphes de la ligne médiane en deux étapes. La première consiste à faire une mise en correspondance de graphes [Rolland 90] [Rolland 91b], la seconde à calculer les graphes intermédiaires (inférence proprement dite) à partir de la mise en correspondance.

Ces deux étapes (mise en correspondance et inférence) sont développées de manière indépendante dans les deux chapitres suivants.

Chapitre 5

Mise en correspondance de graphes

Ce chapitre est consacré à l'étude des différentes méthodes de mise en correspondance de graphes, et à la mise en œuvre de l'une de ces méthodes pour la mise en correspondance de graphes de la ligne médiane. La méthode choisie est détaillée dans la deuxième partie de ce chapitre.

Avant de pouvoir choisir la méthode la plus appropriée pour effectuer la mise en correspondance de graphes de la ligne médiane il faut explorer et comparer les différentes méthodes existantes.

5.1. Méthodes de mise en correspondance de graphes

5.1.1. Définitions

D'une manière intuitive la mise en correspondance de graphes correspond à un processus qui tend à associer "au mieux" les deux graphes, en mettant en correspondance des sommets qui partagent les mêmes propriétés. Ainsi, un sommet du premier graphe ne peut être mis en relation avec un sommet du second graphe que si ce second sommet a, plus ou moins, les mêmes attributs, dans le second graphe, que le premier, dans le premier graphe.

D'un point de vue plus formel on retrouve de nombreuses définitions de la mise en correspondance de graphes.

D.H. Ballard et C.M. Brown [Ballard 82] ne proposent pas moins de cinq définitions différentes de la mise en correspondance :

Soit les graphes $G_1 : (V_1, E_1)$ et $G_2 : (V_2, E_2)$, où V_1 (respectivement V_2) est l'ensemble des sommets de G_1 (respectivement G_2), et où E_1 (respectivement E_2) est l'ensemble des arêtes.

- 1 - Isomorphisme de graphes :

Effectuer la mise en correspondance consiste à trouver un isomorphisme f entre V_1 et V_2 tel que :

pour $v_1 \in V_1, v_2 \in V_2$ avec $f(v_1) = v_2$, on a

$\forall v_1' \in V_1 / (v_1, v_1') \in E_1 \Rightarrow (v_2, f(v_1')) \in E_2.$

- 2 - Isomorphisme de sous-graphes :

La mise en correspondance consiste à trouver un isomorphisme entre le graphe $G_1 : (V_1, E_1)$ et l'un des sous-graphes de $G_2 : (V_2, E_2)$.

- 3 - Isomorphisme "double" de sous-graphes :

La détermination d'une mise en correspondance consiste à rechercher un isomorphisme entre l'un des sous-graphes de (V_1, E_1) et les sous-graphes de (V_2, E_2) .

- 4 - Mise en correspondance incomplète :

Certains sommets peuvent être considérés comme non importants et peuvent être omis lors de la recherche d'isomorphismes (cf. figure 5.1).

- 5 - La cinquième définition est beaucoup plus large :

Lors de la mise en correspondance, un sous-graphe peut être mis en correspondance avec un autre sous-graphe sans que pour autant la correspondance ne soit définie par un isomorphisme entre les sommets et les arcs des graphes ; seule une certaine similarité peut exister entre les deux sous-graphes. Cette similarité suffit pour définir la mise en correspondance. Celle-ci peut donc exister entre deux graphes se ressemblant et dont l'un est plus "riche" que l'autre.

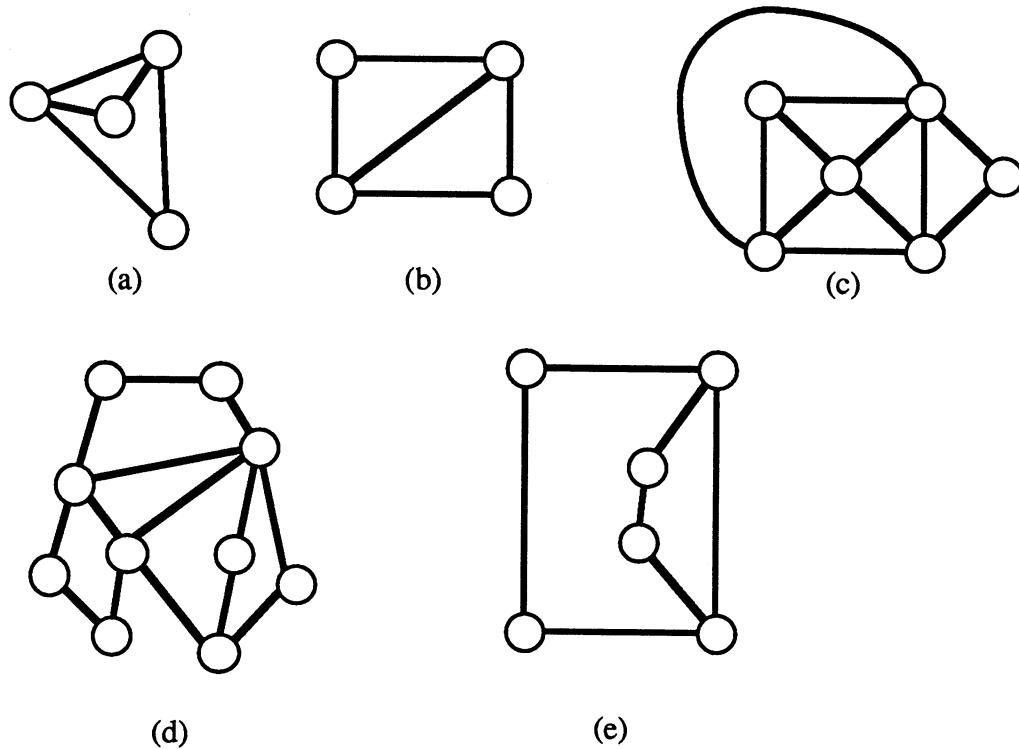


Figure 5.1. Mise en correspondance et isomorphismes.
 Il existe, un isomorphisme entre le graphe (a) et le graphe (b),
 des isomorphismes de sous-graphes entre le graphe (a) et le graphe (c),
 de nombreux isomorphismes de sous-graphes "doubles" entre (a) et (d)
 et plusieurs mises en correspondance partielles entre (a) et (e)
 dépendant du sommet ignoré (extrait de [Ballard 82])

Z. Galil propose quatre problèmes (et algorithmes) de mise en correspondance [Galil 86] :

Soit $G = (V, E)$ un graphe non orienté.

V et E représentent respectivement l'ensemble des sommets et des arêtes du graphe G .

Tout le travail est effectué à l'intérieur de ce graphe G , la mise en correspondance se faisant entre deux classes de sommets.

Les sommets (V) représentent les "personnes", et les arêtes correspondent à des "mariages".

Le but est de sélectionner certaines de ces arêtes.

Les quatre problèmes posés excluent la "polygamie", c'est-à-dire que deux arêtes différentes ne doivent pas joindre le même sommet.

Les quatre problèmes sont décrits ainsi :

- cardinalité maximale de la mise en correspondance dans un graphe biparti :

L'ensemble des sommets V est composé d'"hommes" et de "femmes".

Il ne peut exister d'arêtes qu'entre deux éléments de classes différentes (un élément de la classe "homme" et un élément de la classe "femme").

On recherche alors la mise en correspondance ayant la plus grande cardinalité, c'est-à-dire, la mise en correspondance formant le plus grand nombre de "mariages" possibles.

- cardinalité maximale de la mise en correspondance dans un graphe quelconque :

Les "personnes" sont, cette fois, "asexuées", il peut donc y avoir une arête entre deux éléments quelconques de V , la polygamie étant toujours prohibée.

On cherche alors, là aussi, la mise en correspondance créant le plus de "mariages" possibles.

- poids maximal de la mise en correspondance dans un graphe biparti :

L'ensemble V est, à nouveau, composé de deux classes ("hommes" et "femmes"). A chaque arête ("mariage") est associée une pondération.

Le but est ici, de trouver la mise en correspondance qui maximise le "bénéfice" (somme des pondérations des arêtes retenues) sans se soucier du nombre de "mariages" effectués.

- poids maximal de la mise en correspondance dans un graphe quelconque :

Les sommets sont "asexués" et à chaque arête correspond un "bénéfice".

On recherche, là encore, la mise en correspondance maximisant le "bénéfice" total.

D'autres définitions ont été proposées par différents auteurs [Barrow 76] [Hérault 90]. Certaines d'entre elles ne s'appliquent que dans certains cas bien particuliers. Dans les cas où les graphes sont des ARGs (*Attributed Relational Graphs*) les définitions font souvent intervenir des notions de "distance" entre graphes [Eshera 84] [Shapiro 85] [Chianese 89].

D'autres définitions, pour lesquelles il n'y a pas de véritable association, utilisent la notion de "minimum d'énergie" nécessaire pour déformer l'un des graphes vers l'autre [Ballard 82].

Ces différentes définitions sont souvent conditionnées par l'application à laquelle la mise en correspondance est dédiée. Dans les domaines s'intéressant à la reconnaissance de formes, ce sont les définitions faisant intervenir des "distances" qui sont le plus souvent employées [Miclet 84]. La "distance" calculée correspond, alors, à la "note" attribuée lors de la reconnaissance. Dans ce cas, la distance entre graphes permet de savoir si le graphe mis en correspondance peut ou non être considéré comme un élément de la classe dont le modèle de base est le graphe initial. En robotique et en vision par ordinateur, par contre, ce sont plutôt les définitions à base d'isomorphismes qui sont utilisées. Dans ce dernier cas une mise en correspondance de sommet à sommet est calculée.

Pour la résolution du problème de la mise en correspondance de graphes représentant des formes, la définition la plus appropriée est, sans nul doute, celle qui reste la plus générale. Elle peut s'exprimer par : effectuer une mise en correspondance de graphes consiste à trouver une correspondance maximale entre ces deux graphes ; cette correspondance est en fait une mise en relation d'un sous-graphe de G_1 avec un sous-graphe de G_2 , qui préserve, autant que possible, certaines propriétés. Cette correspondance doit être maximale au sens d'un maximum de représentativité des différents sous-graphes par rapport aux graphes initiaux.

C'est à partir de cette définition que le problème de la mise en correspondance de graphes sera abordé dans la suite. Ce sera notamment sur cette définition que reposera la méthode présentée dans le cadre de la mise en correspondance de graphes de la ligne médiane pour la reconstruction tridimensionnelle à partir de coupes sériées.

5.1.2. Méthodes de mise en correspondance

Bien qu'il existe de nombreuses définitions de la mise en correspondance de graphes, il n'existe en fait que peu de méthodes permettant de résoudre ce problème. Trois méthodes sont généralement utilisées. Il s'agit tout d'abord des méthodes métriques, reposant sur une mesure de similarité entre graphes, ensuite des méthodes de type recherche par retour arrière et enfin des méthodes utilisant un graphe de mise en correspondance sur lequel est effectué une recherche de cliques.

5.1.2.1. Méthodes métriques

Les méthodes métriques procèdent en effectuant des mesures de similarité ou de corrélation entre deux graphes [Shapiro 85]. Ces méthodes permettent en fait de quantifier la fiabilité des mises en correspondance calculées empiriquement [Ballard 82] [Nevatia 82] [Eshera 84]. Elles sont fondées sur des modèles physiques tel que celui des "gabarits et ressorts" (*templates and springs*). Dans ce cas le processus de mise en correspondance procède par déformation. Le graphe initial est modélisé à l'aide d'un ensemble de gabarits et de ressorts. Le processus de mise en correspondance tend à déformer le modèle du graphe initial jusqu'à ce qu'il colle avec le modèle similaire de l'autre graphe. La mise en correspondance ainsi obtenue, une mesure de sa qualité est effectuée, c'est une fonction à la fois des similarités locales des différents gabarits et de l'énergie dépensée dans la déformation des ressorts pour obtenir une telle mise en correspondance.

Cette fonction, appelée fonction *Coût*, s'exprime de manière générale par :

$$\begin{aligned} \text{Coût} = & \sum_{d \in \{\text{TrouvRef}\}} \text{CoûtGabarits} (d , F(d)) \\ + & \sum_{(d,e) \in \{\text{TrouvRef}\} \times \{\text{TrouvRef}\}} \text{CoûtRessorts} (F(d) , F(e)) \\ + & \sum_{c \in \{\text{ManqRef}\} \times \{\text{ManqEntrée}\}} \text{CoûtOubliés} (c) \end{aligned}$$

Le *CoûtGabarits* mesure la dissimilitude entre les gabarits en entrée et les gabarits du modèle.

Le *CoûtRessorts* mesure la dissimilitude des relations, entre éléments mis en correspondance, par rapport aux relations, entre les gabarits de référence.

CoûtOubliés prend en compte les pénalités dues aux éléments, jugés non importants, et omis lors de la mise en correspondance.

F est une fonction des gabarits de référence vers les éléments donnés en entrée.

F partitionne l'ensemble des gabarits de référence en deux classes : la classe $\{\text{TrouvRef}\}$ des gabarits de référence ayant une image parmi les gabarits donnés; la classe $\{\text{ManqRef}\}$ de ceux n'en ayant pas.

Une partition identique peut être faite sur l'entrée permettant ainsi de définir la classe $\{ManqEntrée\}$ des éléments de l'entrée ne possédant pas d'antécédent par F dans l'ensemble des gabarits de référence.

Les méthodes métriques sont assez facilement formalisables, mais, comme la plupart des méthodes tentant de simuler des modèles physiques, elles ne peuvent être implémentées que de manière très coûteuse, tant en temps de calcul qu'en capacités de calcul (précisions) nécessaires.

D'autre part, ces méthodes ne permettent que très difficilement de se rendre compte des mises en correspondance locales. En effet, la mesure de la qualité de la mise en correspondance est une évaluation globale de similarité, donc elle ne permet pas de localiser les portions des graphes qui sont mis en correspondance de manière intéressante. Ces méthodes ne donnent d'ailleurs généralement pas une véritable mise en correspondance entre les sommets des différents graphes. Elles donnent plutôt une évaluation de la similarité entre les deux graphes.

5.1.2.2. Méthodes de recherche par retour arrière

Sous le nom de recherche par retour arrière sont rassemblées toutes les méthodes utilisant des structures étagées dans lesquelles elles effectuent une recherche de solution de manière exhaustive.

Elles procèdent généralement en construisant un arbre de solutions dans lequel chaque étage est l'extension d'une solution partielle de l'étage précédent (cf. figure 5.2).

C'est ensuite en parcourant cet arbre que la solution est recherchée.

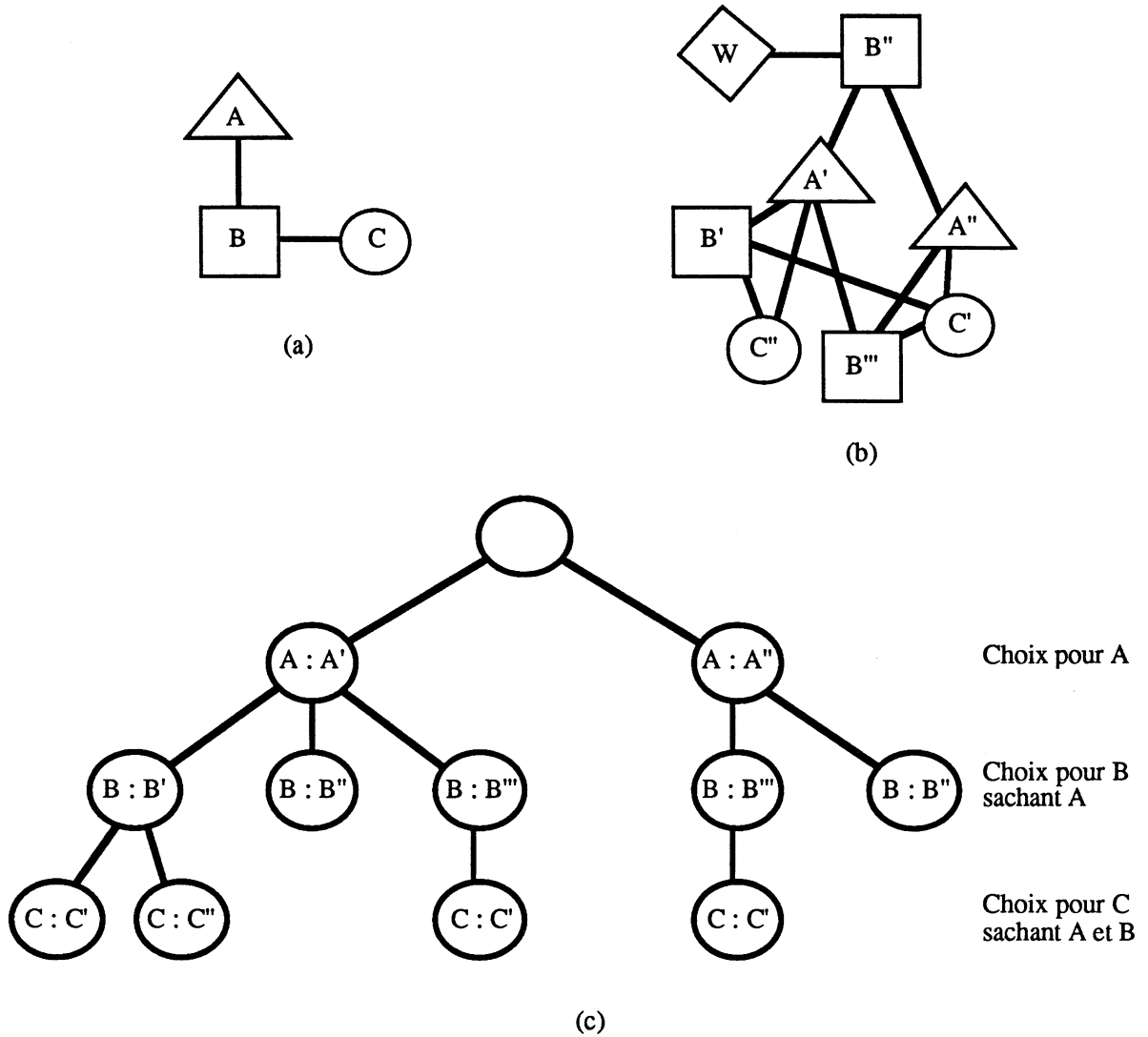


Figure 5.2. Méthode de recherche par retour arrière.
 Le graphe (a) doit être mis en correspondance avec le graphe (b), la mise en correspondance doit conserver les caractéristiques des sommets (données par leur forme) et les relations entre eux (les arêtes).
 (c) est l'arbre de solutions construit par une méthode de recherche par retour arrière (extrait de [Ballard 82]).

L'algorithme proposé par D.H. Ballard et C.M. Brown pour la construction d'isomorphismes de graphes est le suivant :

Soit deux graphes orientés $X = (V_X, E_X)$
 $Y = (V_Y, E_Y)$
 où V_X et V_Y sont les ensembles de sommets et E_X et E_Y les ensembles d'arcs.
 X est le graphe "de référence" et Y le graphe en entrée de l'algorithme.

L'isomorphisme est noté F c'est-à-dire que :

si $i \in V_X$ alors il est mis en correspondance avec $F(i) \in V_Y$.

S dénote l'ensemble des sommets mis en correspondance, et k le niveau de recherche dans l'arbre.

ALGORITHME : recherche d'isomorphismes de graphes orientés

Procédure `isomorphismes_graphes_orientés` (S , k)

début

si $S = V_Y$ alors `isomorphisme_trouvé` (F)

sinon pour tout $v \in (V_Y - S)$ faire

 si `mise_en_correspondance_possible` (k , v) alors

début

$F(k) := v$;

`isomorphismes_graphes_orientés`($S \cup \{v\}$, $k+1$) ;

fin ;

fin ;

Le premier appel est fait avec S vide, le sommet de départ pour la recherche (A dans l'exemple) influera sur l'arbre obtenu mais pas sur le résultat final. En effet, on pourrait penser que le résultat obtenu par cette méthode est fonction de l'ordre dans lequel les sommets sont choisis. Ce n'est en fait pas le cas, si une solution existe (branche la plus profonde de l'arbre) elle doit prendre en compte tous les sommets de l'un des graphes donc, quel que soit le sommet duquel le processus débute, la solution sera la même. Par contre, les solutions partielles dépendent du sommet de départ et de l'ordre de parcours choisi.

Certaines solutions optimisées ont été proposées pour effectuer la construction de l'arbre, elles procèdent en ne créant pas l'arbre tout entier. Dans le cas où certaines branches ne permettent pas d'évolutions vers une solution globale du problème, ces branches sont coupées, ce qui réduit d'autant le processus de parcours de l'arbre (procédé de *branch and bound*).

Ces méthodes de recherche par retour arrière ont l'avantage de pouvoir être implémentées assez facilement. Malheureusement, elles n'autorisent pas la "polygamie". En effet, deux sommets de l'un des graphes ne peuvent pas être mis en correspondance avec un même autre sommet de l'autre graphe. Cette contrainte ne permettra pas d'utiliser cette méthode pour la mise en correspondance de graphes de la ligne médiane.

On peut noter aussi l'existence de méthodes alliant une méthode par recherche par retour arrière et les méthodes métriques [Chianese 89]. Malheureusement, ces méthodes n'autorisant pas non

plus la "polygamie", elles ne pourront être envisagées pour effectuer la mise en correspondance de graphes de la ligne médiane.

5.1.2.3. Méthodes par recherche de cliques

On peut démontrer facilement que la recherche d'une mise en correspondance entre graphes peut se ramener à une méthode de recherche de cliques dans un graphe de mise en correspondance [Barrow 76].

La méthode consiste donc tout d'abord à construire un graphe, appelé graphe de mise en correspondance ou graphe d'association, dans lequel une recherche de cliques est ensuite effectuée.

Le graphe de mise en correspondance est un graphe non orienté composé de sommets et d'arêtes. Chaque sommet de ce graphe est défini par un couple de sommets des graphes initiaux (un sommet de chacun des graphes). Les arêtes, elles, expriment la compatibilité entre les deux sommets qu'elles relient, elles représentent ainsi la compatibilité existant entre deux appariements de sommets des graphes initiaux (cf. figure 5.3).

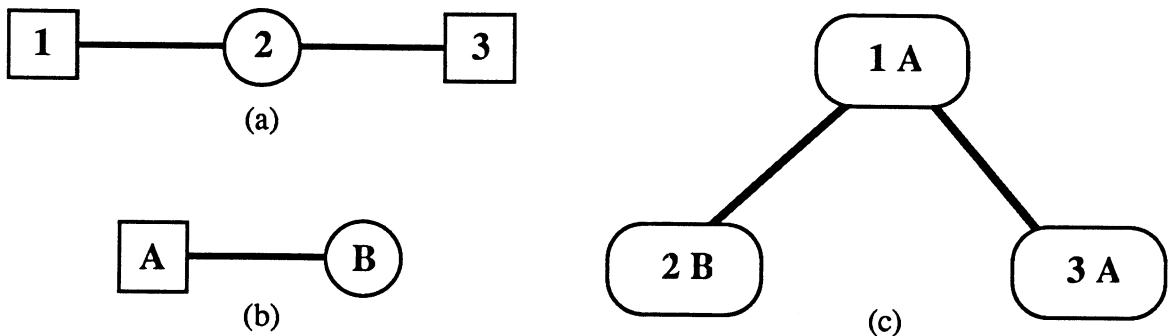


Figure 5.3. Graphe de mise en correspondance.
Pour mettre en correspondance le graphe (a) avec le graphe (b) on construit le graphe de mise en correspondance (c) où les arêtes représentent la compatibilité des appariements.

La recherche d'une mise en correspondance est alors effectuée par l'intermédiaire d'une recherche de cliques dans ce graphe de mise en correspondance [Barrow 76] [Skordas 88] [Horaud 89] [Hérault 90]. En effet, une clique, sous-graphe complet du graphe de mise en correspondance, est un graphe complet, donc tous les appariements représentés par les sommets sont compatibles deux à deux ; la clique exprime donc bien une mise en correspondance entre deux sous-graphes. Dans le cas où les deux graphes initiaux s'apparient

exactement, la mise en correspondance est représentée par la clique maximale (clique possédant le plus grand nombre de sommets) du graphe de mise en correspondance.

Cette méthode, grâce à la définition des sommets du graphe de mise en correspondance, autorise la "polygamie". Par contre, la recherche de cliques est un problème NP-complet, qui constitue le problème central de cette méthode. C'est cette méthode que j'ai retenue pour effectuer la mise en correspondance de graphes de la ligne médiane, car elle est la seule à autoriser la "polygamie".

5.2. Méthode de mise en correspondance de graphes de la ligne médiane

Le but à atteindre est d'effectuer une mise en correspondance entre deux graphes de la ligne médiane. Cette mise en correspondance peut n'être que très peu précise : elle peut n'être qu'une mise en correspondance entre quelques sommets des graphes permettant de maintenir certaines caractéristiques.

La recherche de mise en correspondance s'effectue en recherchant des cliques dans un graphe de mise en correspondance.

Dans un premier temps, c'est la construction de ce graphe de mise en correspondance qui va être décrite. Dans un second temps, les différentes méthodes de recherche de cliques seront abordées. Enfin, des résultats issus du processus de mise en correspondance seront présentés.

5.2.1. Construction du graphe de mise en correspondance

Tout d'abord, la méthode de construction du graphe de mise en correspondance est donnée dans un cadre assez général. Ensuite la construction dans le cas où les graphes initiaux sont des graphes de la ligne médiane est précisée.

5.2.1.1. Principe général

Le graphe de mise en correspondance est composé de sommets et d'arêtes construits de la manière suivante :

- les sommets :

Les sommets représentent des appariements de sommets des graphes initiaux.

Chaque sommet du graphe de mise en correspondance est une paire (v_1, v_2) avec $v_1 \in V_1$ et $v_2 \in V_2$, il représente donc l'appariement entre un sommet du premier graphe $G_1 : (V_1, E_1)$ et un sommet du second $G_2 : (V_2, E_2)$.

Toutes les paires de sommets possibles sont généralement construites ; seules certaines contraintes liées aux types des sommets de G_1 et de G_2 peuvent permettre de supprimer certains appariements.

- les arêtes :

Elles expriment la compatibilité entre deux sommets du graphe de mise en correspondance en tenant compte des particularités géométriques des graphes de la ligne médiane.

Chaque arête exprime la compatibilité entre deux appariements. Cette notion de compatibilité représente simplement le fait que, si les sommets (v_1, v_2) et (v'_1, v'_2) du graphe mise en correspondance sont reliés, alors il est possible de mettre en correspondance à la fois v_1 avec v_2 et v'_1 avec v'_2 .

Pour pouvoir orienter la recherche de cliques vers la solution la plus intéressante il est ensuite nécessaire de pondérer les sommets du graphe en fonction de l'importance de l'appariement qu'ils représentent par rapport à la mise en correspondance globale. Ce sont les sommets qui sont pondérés plutôt que les arêtes ; une pondération des arêtes n'apporterait rien (elle ne permettrait d'exprimer qu'une "compatibilité conditionnelle") et rendrait encore plus complexe la recherche de cliques.

5.2.1.2. Construction du graphe de mise en correspondance

Je décris ici comment le principe de construction d'un graphe de mise en correspondance a été appliqué aux graphes de la ligne médiane.

Soit GLM_1 et GLM_2 les graphes de la ligne médiane à mettre en correspondance.

Il faut dans un premier temps définir les sommets du graphe de mise en correspondance, dans un second temps, définir les arêtes entre ces sommets et enfin pondérer les sommets des graphes de mise en correspondance.

Les sommets

⌋ Ce sont des couples (v_1, v_2) avec $v_1 \in \text{GLM}_1$ et $v_2 \in \text{GLM}_2$.

L'exploitation des caractéristiques des graphes de la ligne médiane va permettre de ne pas construire tous les couples (v_1, v_2) .

On sait que (cf. § 2.2.6), dans le graphe de la ligne médiane, tous les sommets principaux codent la partie principale de la forme, les sommets de rétrécissement expriment les rétrécissements de la forme et les sommets de proéminence représentent les déformations sur les composantes principales ou de rétrécissement.

Ces particularités sont utilisées lors de la construction des sommets du graphe de mise en correspondance. Ainsi, tous les couples ne sont pas construits.

Les proéminences larges, par exemple, ne servent en général qu'à assurer la connexion entre une proéminence stricte et le sommet qu'elle déforme, aussi elles ne sont pas retenues pour la construction des sommets du graphe de mise en correspondance.

D'autre part, certaines proéminences ne déforment que très peu la forme, ainsi une sélection des sommets de proéminence stricte les plus déformants est effectuée. Cette sélection est possible grâce aux diverses informations mémorisées pour chacun des sommets du graphe de la ligne médiane.

En effet, la déformation produite par une proéminence stricte est inversement proportionnelle à la valeur χ où χ est définie par :

$$\chi = \frac{\text{poids du sommet déformé} - \text{poids de la proéminence stricte}}{d}$$

⌋ d représente la distance séparant la proéminence stricte du sommet déformé (principal ou rétrécissement) mesurée selon la distance d_g .

Ainsi, seules les proéminences strictes pour lesquelles la valeur χ est inférieure à un certain seuil sont retenues pour la construction des sommets du graphe de mise en correspondance.

Etant donné $0 < \chi \leq 1$, un seuil a été retenu de manière empirique, sa valeur est 0,8. Ce seuil semble assez bien refléter la réalité de l'importance des déformations.

Ainsi :

$$\left| \begin{array}{l} 0 < \chi_{\text{proéminences strictes retenues}} < 0,8 \leq \chi_{\text{proéminences strictes non retenues}} \leq 1 \end{array} \right.$$

Cette sélection a aussi pour effet de réduire les erreurs liées à l'utilisation de la distance de l'échiquier (d_g) lors du calcul des graphes de la ligne médiane (non isotropie).

Cette sélection correspond en fait à un filtrage du graphe de la ligne médiane dans un but de description de formes (cf. figure 5.4).

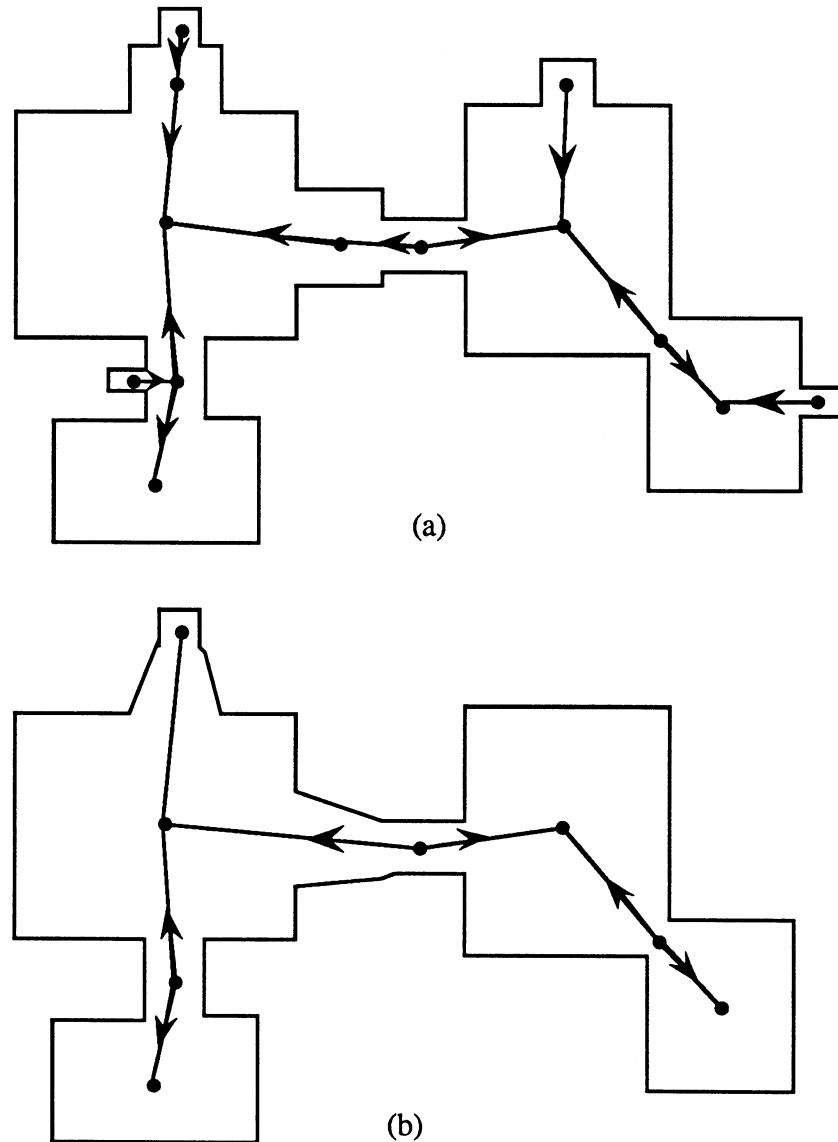


Figure 5.4. Filtrage du graphe de la ligne médiane.
La forme initiale (a) et la forme issue du graphe filtré (b).

Sept types d'appariements, au lieu des seize possibles (quatre types de sommets \times quatre types de sommets), sont ainsi retenus pour la construction des sommets du graphe de mise en correspondance. Les appariements (sommets principaux - sommets de proéminence stricte) ne sont pas retenus car ils ne représenteraient rien d'intéressant au niveau de la forme.

Les sept types d'appariements retenus sont :

- * sommet de type principal - sommet de type principal

- * sommet de type rétrécissement - sommet de type rétrécissement
- * sommet de type proéminence stricte ($\chi < \text{seuil}$) - sommet de type proéminence stricte ($\chi < \text{seuil}$)
- * sommet de type principal - sommet de type rétrécissement
- * sommet de type rétrécissement - sommet de type principal
- * sommet de type proéminence stricte ($\chi < \text{seuil}$) - sommet de type rétrécissement
- * sommet de type rétrécissement - sommet de type proéminence stricte ($\chi < \text{seuil}$).

Les sommets du graphe de mise en correspondance sont construits à l'aide des numéros des sommets des graphes de la ligne médiane dont ils représentent l'appariement.

Les arêtes

Chaque arête du graphe de mise en correspondance doit exprimer une compatibilité entre les deux appariements qu'elle relie.

La seule relation utilisée ici tient compte de la planarité du graphe de la ligne médiane. Elle met en valeur le fait qu'on souhaite privilégier les translations par rapport au rotations. Elle utilise les coordonnées des sommets des graphes de la ligne médiane (cf. figure 5.5).

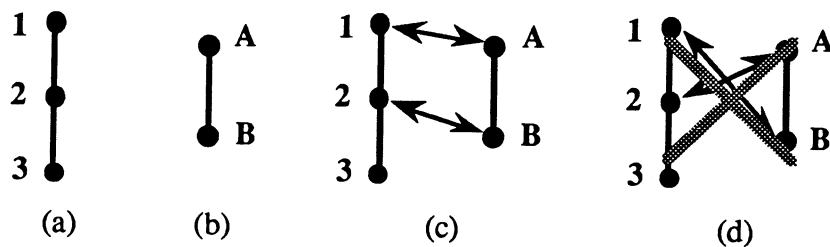


Figure 5.5. Compatibilité des appariements.
Lors de la mise en correspondance de (a) avec (b),
l'appariement (1, A) est compatible avec l'appariement (2, B) (c),
par contre (d) l'appariement (2, A) ne l'est pas avec (1, B).

Lors de la construction du graphe de mise en correspondance tous les appariements sont analysés deux à deux afin de déterminer s'ils sont compatibles ou non.

Pondération des sommets

Afin de rendre la recherche de cliques plus performante et cohérente, et afin d'exploiter davantage les spécificités des graphes de la ligne médiane et ainsi de permettre que la clique recherchée (et trouvée) ait une signification au niveau des formes mises en correspondance, il est souhaitable de pondérer les sommets du graphe de la mise en correspondance.

La pondération attribuée aux sommets du graphe de mise en correspondance correspond en fait à une notion d'invraisemblance de l'appariement considéré. Par exemple, il est plus vraisemblable de mettre en correspondance un sommet principal, avec un sommet principal, plutôt qu'avec un sommet de rétrécissement. Cette pondération peut aussi être considérée comme relatant l'"improbabilité" d'avoir tel ou tel appariement.

Ainsi, la pondération (le coût) d'un sommet du graphe de mise en correspondance sera d'autant plus forte que l'appariement, entre les sommets des graphes de la ligne médiane, considéré est peu vraisemblable.

Pour ce faire, quatre critères ont été retenus pour permettre le calcul de la pondération d'un sommet du graphe de mise en correspondance. A ces quatre critères correspondent en fait quatre pondérations, pondérations qui permettent, par combinaison linéaire, de calculer la pondération globale du sommet. Ces pondérations utilisent les informations liées aux sommets des graphes de la ligne médiane.

Les quatre pondérations retenues sont :

- une pondération selon le type des sommets des graphes de la ligne médiane
- une pondération selon la taille et le nombre de boules discrètes codées par ces sommets
- une pondération selon la configuration locale des graphes de la ligne médiane autour de ces mêmes sommets
- une pondération selon la position des composantes codées par les sommets des graphes de la ligne médiane considérés par rapport aux formes globales codées par les graphes.

• **Pondération selon le type de sommets**

Cette pondération a pour effet de mettre en correspondance plutôt des sommets de même type entre eux plutôt qu'avec des sommets de type différent.

On rappelle que seuls sept types d'appariements sont possibles. La figure 5.6 donne la table des pondérations utilisée. L'ordre ainsi défini tend à privilégier la mise en correspondance des

sommets les plus représentatifs de la forme (principaux puis rétrécissements puis proéminences).

type du sommet du GLM ₁	type du sommet du GLM ₂	<i>pondération</i>
principal	principal	1
principal	rétrécissement	5
rétrécissement	rétrécissement	2
rétrécissement	proéminence stricte ($\chi < 0.8$)	9
rétrécissement	principal	5
proéminence stricte ($\chi < 0.8$)	proéminence stricte ($\chi < 0.8$)	3
proéminence stricte ($\chi < 0.8$)	rétrécissement	9

Figure 5.6. Table des pondérations liées aux types de sommets.

Il faut noter que ces valeurs, comme celles données plus loin, ne sont données qu'à titre indicatif, elles n'ont ainsi aucune valeur de généralité. Cependant, ces valeurs sont celles qui ont été utilisées lors de l'implémentation de la mise en correspondance de graphes de la ligne médiane ; les choix ont été validés par les résultats obtenus.

Les valeurs des différentes pondérations ont été volontairement toutes bornées entre 1 et 9, cela afin de permettre, lors du calcul de la pondération globale, de rendre prédominante telle ou telle pondération.

- **Pondération selon la taille et le nombre de boules discrètes codées**

Cette pondération est construite afin de mettre en valeur la similarité, au niveau surfacique, des différentes composantes codées par les sommets.

Cette mesure de similarité est effectuée en utilisant la taille et le nombre de boules codées.

Ainsi, cette pondération met en valeur le fait qu'un sommet codant quatre boules de taille trois doit avoir plus de "chance" d'être mis en correspondance avec un sommet codant quatre boules de taille trois, ou trois boules de taille quatre, qu'avec un sommet codant deux boules de taille deux, ou cinq boules de taille un (cf. figure 5.7).

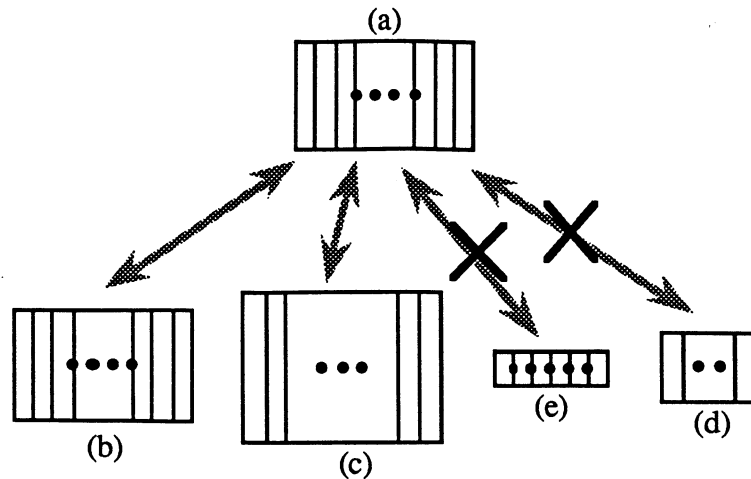


Figure 5.7. Pondération selon la taille et le nombre de boules codées : (a) doit pouvoir être plus facilement apparié avec (b) ou (c) qu'avec (d) ou (e).

Cette évolution, à l'aide du nombre et de la taille des boules codées n'est qu'une approximation de la surface couverte par les composantes.

Un calcul exact de la surface nécessiterait une reconstruction partielle de l'image ou des calculs assez complexes [Cordella 89], il serait donc coûteux en temps de calcul. Aussi nous ne l'avons pas mis en œuvre, d'autant plus que l'évaluation, à partir des tailles et du nombre de boules, est très rapide et fournit une évaluation suffisante.

Les pondérations attribuées sont données figure 5.8 en utilisant les définitions suivantes :

On note (p_1, n_1) (respectivement (p_2, n_2)) les poids des boules et le nombre de boules discrètes codées par le sommet de GLM_1 (respectivement GLM_2).

On note $\varepsilon \in \mathbb{N}$ le paramètre définissant la différence de taille autorisée et $\alpha \in \mathbb{N}$ le paramètre désignant la différence de nombre de boules autorisée (les valeurs de ε et α sont fixées empiriquement par l'utilisateur).

<i>pondération</i>	$p_1 = p_2$	$p_1 - p_2 > -\varepsilon$	$p_1 - p_2 \leq \varepsilon$	$p_1 - p_2 \leq -\varepsilon$	$p_1 - p_2 > \varepsilon$
$n_1 = n_2$	1	3	3	5	5
$n_1 - n_2 > -\alpha$	2	5	4	7	6
$n_1 - n_2 \leq \alpha$	2	4	5	6	7
$n_1 - n_2 \leq -\alpha$	5	7	6	9	8
$n_1 - n_2 > \alpha$	5	6	7	8	9

Figure 5.8. Table des pondérations liées au poids et au nombre de boules discrètes codées.

• **Pondération selon la configuration du graphe de la ligne médiane.**

Cette pondération tend à mettre en correspondance des sommets qui sont au centre de configurations locales identiques du graphe de la ligne médiane. Cette pondération correspond aussi à des réalités au niveau des déformations du graphe de la ligne médiane.

Cette pondération, assez proche de celle liée au type des sommets, puisque les types des sommets sont déterminés à l'aide des configurations locales du graphe de la ligne médiane, est plus complète que cette dernière. C'est en fait une évaluation non seulement locale mais aussi globale des configurations du graphe. Ceci est dû au fait que les degrés, utilisés pour le calcul de la pondération, ne sont pas exactement calculés de la même manière.

En effet, ici, le degré sortant (respectivement entrant) est calculé jusqu'à la fin (respectivement jusqu'au début) des chemins d'origine (respectivement de destination) du sommet considéré. Alors que pour la détermination des types, les degrés sont calculés localement au niveau du sommet considéré (cf. figure 5.9).



Figure 5.9. Calcul des degrés.

Lors de la détermination du type un tel sommet est considéré comme ayant un degré sortant égal à 1 (évaluation locale).

Lors du calcul de la pondération, il est considéré comme ayant un degré sortant égal à 2 (évaluation globale).

La pondération liée aux configurations des graphes est donnée à l'aide des définitions suivantes dans la table de la figure 5.10.

On note (de_1, ds_1) (respectivement (de_2, ds_2)) les degrés entrant et sortant du sommet $v_1 \in GLM_1$ (respectivement $v_2 \in GLM_2$).

On représente par $\beta \in \mathbb{N}$ et $\gamma \in \mathbb{N}$ les paramètres définissant les différences autorisées sur les degrés entrants et sortants.

<i>pondération</i>	$de_1 = de_2$	$ de_1 - de_2 \leq \beta$	$ de_1 - de_2 > \beta$
$ds_1 = ds_2$	1	3	7
$ ds_1 - ds_2 \leq \gamma$	3	5	8
$ ds_1 - ds_2 > \gamma$	7	8	9

Figure 5.10. Table des pondérations liées à la configuration des graphes de la ligne médiane au niveau des sommets appariés.

- **Pondération liée à la position de la composante codée par rapport à la forme globale.**

Cette pondération a pour but de mettre de préférence en correspondance des parties de l'image qui se trouvent dans la même zone par rapport à la forme globale que les parties se trouvant dans des zones éloignées l'une de l'autre, toujours par rapport à la forme globale (cf. figure 5.11). Cette pondération favorise ainsi les déformations par rapport aux rotations.

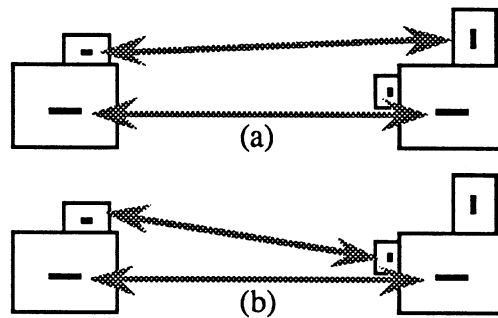


Figure 5.11. Pondération liée à la position de la composante codée par rapport à la forme globale : les appariements représentés par (a) sont préférables à ceux présentés en (b).

La pondération est définie cette fois de manière algorithmique en utilisant les définitions suivantes :

On note θ_1 (respectivement θ_2) l'angle entre l'horizontale et la droite reliant le centre de gravité de la forme codée par GLM₁ (respectivement GLM₂) et celui de la composante codée par v_1 (respectivement v_2).

On a $0 \leq \theta_1 < 2\pi$ et $0 \leq \theta_2 < 2\pi$

Soit $\delta \in \mathbb{R}^+$ la différence d'angle autorisée.

La pondération s'exprime alors par :

$$\begin{aligned} &\text{si } \frac{|\theta_1 - \theta_2|}{\delta} > 8 \text{ alors pondération} := 9 \\ &\text{sinon pondération} := \left\lfloor \frac{|\theta_1 - \theta_2|}{\delta} \right\rfloor + 1 \end{aligned}$$

La pondération obtenue est ainsi bien dans l'intervalle discret [1 .. 9].

- **Pondération globale.**

La pondération globale attribuée à un sommet du graphe de mise en correspondance est calculée à l'aide d'une combinaison linéaire des quatre pondérations précédentes. Les coefficients de la combinaison linéaire permettent d'augmenter ou de diminuer les incidences des différentes pondérations.

Les valeurs des différents coefficients permettent de définir une heuristique de mise en correspondance. La pondération selon le type des sommets semble plus importante que les autres, un coefficient 2 lui est attribué, les autres coefficients étant à 1. Les exemples des graphes de mise en correspondance donnés dans la suite le sont en utilisant ces coefficients. La figure 5.12 présente un exemple de graphe de mise en correspondance construit à partir de deux formes représentées par leur graphe de la ligne médiane.

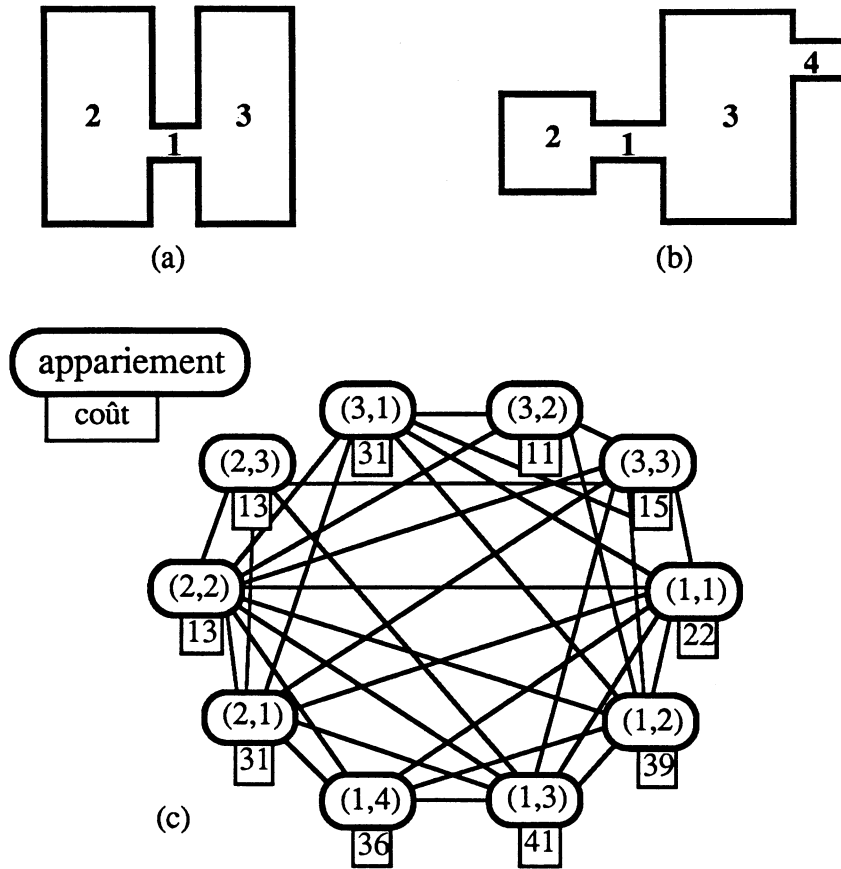


Figure 5.12. Deux formes (a) et (b) avec les numéros des sommets des graphes de la ligne médiane correspondants ; (c) le graphe de mise en correspondance construit.

5.2.2. Recherche de cliques

Le graphe étant construit, le problème de la mise en correspondance de graphes se transforme en un problème de recherche de cliques. En effet, un sous-graphe complet du graphe de mise en correspondance définit une correspondance entre le graphe G_1 et le graphe G_2 .

Le sous-graphe complet $C : (S, A)$ produit deux sous-ensembles de sommets $C_1 \subseteq V_1$ et $C_2 \subseteq V_2$ tels que :

$$\forall v_1 \in C_1 \exists v_2 \in C_2 \text{ avec } (v_1, v_2) \in S \text{ et } \forall v_2' \in C_2 \\ (v_1, v_2') \in S \Rightarrow ((v_1, v_2), (v_1, v_2')) \in A$$

et

$$\forall v_2 \in C_2 \exists v_1 \in C_1 \text{ avec } (v_1, v_2) \in S \text{ et } \forall v_1' \in C_1 \\ (v_1', v_2) \in S \Rightarrow ((v_1, v_2), (v_1', v_2)) \in A$$

Ainsi une mise en correspondance est obtenue entre C_1 et C_2 , elle est définie par :

$$\begin{array}{l} \mathbf{M} : C_1 \rightarrow C_2 \\ \quad v_1 \rightarrow v_2 \text{ tel que } (v_1, v_2) \in S. \end{array}$$

La clique maximale exprime donc la meilleure mise en correspondance entre les graphes G_1 et G_2 , tout en préservant des relations de compatibilité entre les composantes de ces graphes.

Ainsi, pour trouver une mise en correspondance, il ne reste plus qu'à effectuer une recherche de cliques dans ce graphe de mise en correspondance !

C'est à ce niveau, en fait, que le problème se complique. En effet, la recherche de cliques (sous-graphe complet) maximales ou de taille k donnée est un problème de la classe des problèmes NP-complets [Christofides 75] [Bondy 76].

Les problèmes NP-complets sont des problèmes dont on ne sait pas dire s'ils peuvent ou s'ils ne peuvent pas être résolus par un algorithme polynomial, mais, de toutes façons, on ne connaît pas d'algorithme polynomial pour les résoudre [Sakarovitch 84].

La recherche de cliques a fait l'objet de nombreux travaux [Bron 73] [Bolles 79] [Ballard 82] [Skordas 88].

Deux types de méthodes sont principalement employés. Il s'agit des méthodes opérant par recherches exhaustives et les méthodes heuristiques.

Avant de présenter ces méthodes il convient de noter ici que si, lors de la construction du graphe de mise en correspondance, les arêtes expriment l'incompatibilité des appariements (au lieu d'exprimer la compatibilité), la recherche d'une mise en correspondance se transforme en recherche de stables (graphes sans arête maximaux).

Malheureusement, le problème de recherche des stables est, lui aussi, de la classe des problèmes NP-complets (c'est le dual du problème de recherche de cliques).

5.2.2.1. Recherches exhaustives

Ces méthodes sont généralement utilisées pour la recherche de cliques dans des graphes non orientés classiques.

Elles sont fondées sur une recherche exhaustive de toutes les cliques contenues dans le graphe afin de trouver celle qui est intéressante.

La construction des différentes cliques est généralement faite par induction en utilisant la caractéristique qu'un sous-graphe complet à k sommets est un sous-graphe complet à $k-1$ sommets auquel on a ajouté un sommet adjacent à ces $k-1$ sommets.

L'algorithme peut être écrit de la manière suivante [Ballard 82] :

La fonction *Cliques* (X, Y) a comme argument une clique X et un ensemble Y de sommets qui inclus X . Il retourne toutes les cliques incluant X qui sont contenues dans Y .

Cliques (\emptyset, S), si S est l'ensemble des sommets d'un graphe, permet de calculer toutes les cliques de ce graphe.

Cliques (X, Y) :=

si aucun sommet de $Y - X$ n'est adjacent à tous les éléments de X

alors $\{X\}$

sinon *Cliques* ($X \cup \{y\}, Y$) \cup *Cliques* ($X, Y - \{y\}$) où y est relié à tous les éléments de X .

Cette méthode n'est utilisable que dans le cas où le nombre de sommets du graphe n'est pas trop important. Lors de l'implémentation, pour un graphe composé de 209 sommets et 7828 arêtes, le calcul a été interrompu au bout de 76 heures de calcul, le nombre de cliques parcourues était déjà de 22 800 000 sans que pour autant elles n'aient été toutes visitées (cf. figure 5.18).

D'autres méthodes ont été développées ; elles permettent de rechercher la clique maximale (clique ayant le plus de sommets) à l'aide d'une technique de type *branch and bound* [Bron 73] [Bolles 79].

Elles sont largement utilisées pour certaines applications telles que la reconnaissance de formes ou la stéréovision [Skordas 88] [Heraud 89].

Ces différents algorithmes sont très rapides mais, malheureusement, dans notre cas, ce n'est pas nécessairement la clique maximale qui est la plus intéressante.

Malgré cela, la recherche de la clique maximale peut parfois être utilisée. En effet, la recherche est rapide et dans certains cas, la clique minimisant la fonction coût est la clique maximale.

La recherche de la clique maximale est en fait utilisée systématiquement en guise de prétraitement. Cette recherche permet de déterminer rapidement s'il existe ou non une clique à k sommets. En effet, si la clique maximale possède plus de k sommets on est assuré qu'il existe

au moins une clique à k sommets, dans le cas contraire ce n'est pas la peine de rechercher une clique à k sommets puisque la plus grande clique possède moins de k sommets.

5.2.2.2. Recherches heuristiques

Les recherches exhaustives n'étant pas très réalistes pour résoudre notre problème, nous nous sommes tournés vers les recherches heuristiques.

Avant de développer les deux méthodes utilisées, il convient de préciser à nouveau le cadre dans lequel on doit effectuer la recherche de cliques.

Dans le cas qui nous intéresse, le graphe est pondéré ; le problème consiste donc à trouver une clique qui minimise une fonction calculée à l'aide des pondérations des sommets qui la composent.

Ainsi un compromis doit être trouvé entre le nombre de sommets de la clique et le "coût" de cette même clique. En effet, en ne retenant, par exemple, que les sommets dont les coûts sont très faibles, on peut arriver à des aberrations si le nombre de sommets retenus est trop faible. En effet, la clique représente une mise en correspondance entre deux graphes, ainsi, si le nombre de sommets de la clique est trop faible alors la mise en correspondance ne signifie plus rien au niveau des objets manipulés (coupes) ; d'où la nécessité de retenir un nombre suffisamment grand de sommets pour la clique.

Pour tenir compte de ces deux paramètres, chaque clique, après son obtention, doit être "évaluée".

Cette évaluation se fait selon deux critères :

- le coût des sommets que la clique contient
- la représentativité de la clique par rapport aux formes mises en correspondance.

Une fonction *coût-clique* est introduite. Elle est basée à la fois sur le coût des sommets constituant la clique, et sur le nombre de sommets de chaque graphe de la ligne médiane, non présents dans les appariements (sommets) de la clique.

La fonction de calcul du coût d'une clique est la suivante :

Soit $G_1 : (V_1, E_1)$ à mettre en correspondance avec $G_2 : (V_2, E_2)$, la clique étant $C : (S, A)$ on a :

$$\begin{aligned} \text{coût-clique (C)} &= \sum_{v \in C} \text{poids (v)} \\ &+ \left(\left| \left\{ v_1 \in V_1 / \forall v_2 \in V_2 (v_1, v_2) \notin S \right\} \right| \right. \\ &\quad \left. + \left| \left\{ v_2 \in V_2 / \forall v_1 \in V_1 (v_1, v_2) \notin S \right\} \right| \right) \\ &\quad) * \text{coût-sommet-non-apparié} \end{aligned}$$

Etant donné que le non-appariement d'un sommet est très préjudiciable par rapport à un appariement, même pas très bon, de celui-ci (on souhaite appairier le plus grand nombre de sommets), et que le coût attribué à un appariement est inférieur à 45, la valeur de la constante *coût-sommet-non-apparié* a été fixée à 50.

Ces précisions apportées, il faut voir comment une recherche heuristique de solution peut être envisagée.

Le problème consiste à trouver un sous-graphe complet qui minimise la fonction coût. Pour pouvoir utiliser la recherche heuristique de solution il faut transformer ce problème de recherche de cliques en un problème d'optimisation combinatoire.

Un parcours (heuristique) de l'ensemble des cliques ne semble pas très réaliste étant donnée la difficulté d'obtention de l'ensemble des cliques ; il semble plus intéressant (et aussi plus simple) de parcourir plutôt l'ensemble des sous-graphes (complets ou non).

Afin de pouvoir effectuer ce parcours tout en conservant le but fixé, il est nécessaire de transformer la fonction *coût-clique* en une fonction utilisable sur tous les sous-graphes. Cette fonction devra, malgré tout, toujours donner un résultat plus faible pour un sous-graphe complet que pour un sous-graphe non complet. La fonction doit d'ailleurs aussi rester minimale pour la clique minimisant la fonction *coût-clique*.

Or on sait que :

$$\begin{aligned} &\text{Pour toute clique C} \\ &\text{coût-clique (C)} \leq (|V_1| + |V_2|) * \text{coût-sommet-non-apparié} \end{aligned}$$

Ainsi, si un coût de $(|V_1| + |V_2|) * \text{coût-sommet-non-apparié} + 1$ est affecté à tout sous-graphe non complet, et si *coût-clique* est affecté aux sous-graphes complets, le parcours de l'ensemble des sous-graphes devient suffisant, les sous-graphes non complets étant ainsi rejetés parmi les plus mauvais.

La fonction d'évaluation ainsi construite, le problème de recherche de clique se transforme donc en un problème d'optimisation combinatoire pour lequel on doit minimiser la fonction coût.

L'ensemble des sous-graphes étant encore plus grand que l'ensemble des cliques, c'est vers une méthode de parcours non exhaustive qu'il convient de se tourner.

Parmi les méthodes classiquement utilisées pour résoudre les problèmes NP-complets de ce type, on trouve le recuit simulé [Sakarovitch 84] [Laarhoven 87] et les algorithmes génétiques [Holland 75]. Ce sont ces deux méthodes que j'ai mises en œuvre, toutes deux permettent de parcourir, de manières très différentes, l'ensemble des sous-graphes du graphe de mise en correspondance. Ces deux méthodes sont tout d'abord présentées de manière théorique, leur utilisation étant précisée ensuite.

Le recuit simulé

Le recuit simulé, dans sa forme originelle, est fondé sur l'analogie entre les problèmes d'optimisation combinatoire et les phénomènes de la physique statistique [Kirkpatrick 83] [Kirkpatrick 84].

Le principe qu'il développe est le suivant :

Soit S une solution réalisable d'un problème, et $V(S)$ un voisinage de S .

On tire au hasard $S' \in V(S)$.

Si $f(S') < f(S)$ (la solution tirée est meilleure que la solution courante) alors S' remplace S et on recommence.

Si ce n'est pas le cas, on peut malgré tout remplacer S par S' , cela dépend d'un autre tirage aléatoire (simulation de Monte-Carlo) dont la probabilité est :

$$p(T) = 1 - e^{-k(T-T^0)} \quad \text{avec } T \geq T^0$$

T (appelée "température", par analogie avec le modèle physique) est un paramètre de contrôle du processus que l'on fait décroître de la température initiale jusqu'à la température finale (inférieure à T^0).

Lorsque T devient inférieur ou égal à T^0 seuls les changements bénéfiques sont admis. Par contre, on autorise d'autant plus d'échanges désavantageux que T est supérieur à T^0 , l'idée étant que les mouvements localement désavantageux permettront de ne pas se laisser "piéger" dans un optimum local (ils permettront de sauter les "barrières de potentiels") (cf. figure 5.13).

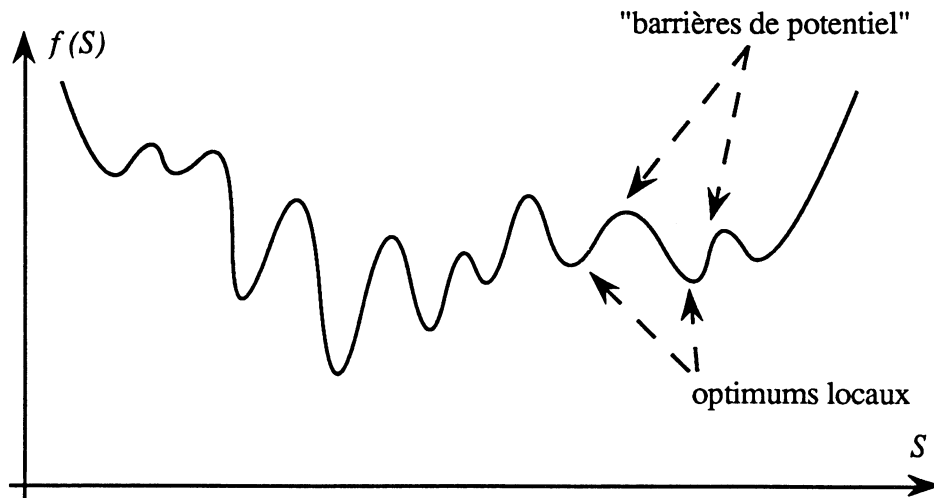


Figure 5.13. Optimums locaux et "barrières de potentiel"
(extrait de [Sakarovitch 84]).

Cette méthode de recuit simulé a prouvé son efficacité pour la résolution de problèmes tels que celui du "voyageur de commerce" en donnant des résultats spectaculaires.

C'est tout d'abord cette méthode de recuit simulé que j'ai mise en œuvre.

Afin de pouvoir utiliser le recuit simulé pour la recherche de la clique minimisant la fonction coût, il faut définir la notion de voisinage d'une solution réalisable.

Dans le cas qui nous intéresse, on remarque qu'on recherche en fait une clique à k sommets où k peut être considéré comme une donnée du problème. En effet, les graphes $G_1 : (V_1, E_1)$ et $G_2 : (V_2, E_2)$ sont connus, donc le nombre de sommets à apparier, afin que la mise en correspondance soit représentative, peut être évalué.

On recherche donc une clique à k sommets. La notion de voisinage d'une solution réalisable pour permettre de se déplacer dans l'espace de solutions (sous-graphes à k sommets) est définie de la manière suivante :

Le voisinage d'un sous-graphe à k sommets est composé des sous-graphes à k sommets qui ne diffèrent du premier que par un seul de leurs sommets.

Cette notion de voisinage permet de mettre en œuvre la recherche de la clique minimisant la fonction coût par recuit simulé.

Cette méthode de recherche de cliques a l'avantage de permettre de maîtriser le temps de calcul, tout en étant sûr que la solution trouvée est assez proche de l'optimum.

En effet, la fonction de décroissance de la température étant contrôlable ainsi que la température finale (température à laquelle on arrête le calcul), on peut en fait choisir le nombre d'itérations réalisées avant l'arrêt du processus.

D'autre part, en mémorisant la plus petite valeur de la fonction coût obtenue au cours des itérations successives, on obtient "toujours" une bonne solution. Ceci peut paraître très bizarre étant donné que le procédé de recuit simulé fait intervenir de nombreux tirages aléatoires (choix de la solution initiale, choix de S' dans $V(S)$, probabilité, ...), mais dans la pratique tout se passe très bien ("Y aurait-il un ordre dans les chaos ?" [Ordre & Chaos]).

La figure suivante (figure 5.14) présente un exemple de l'évolution de la fonction coût au cours d'une recherche de clique.

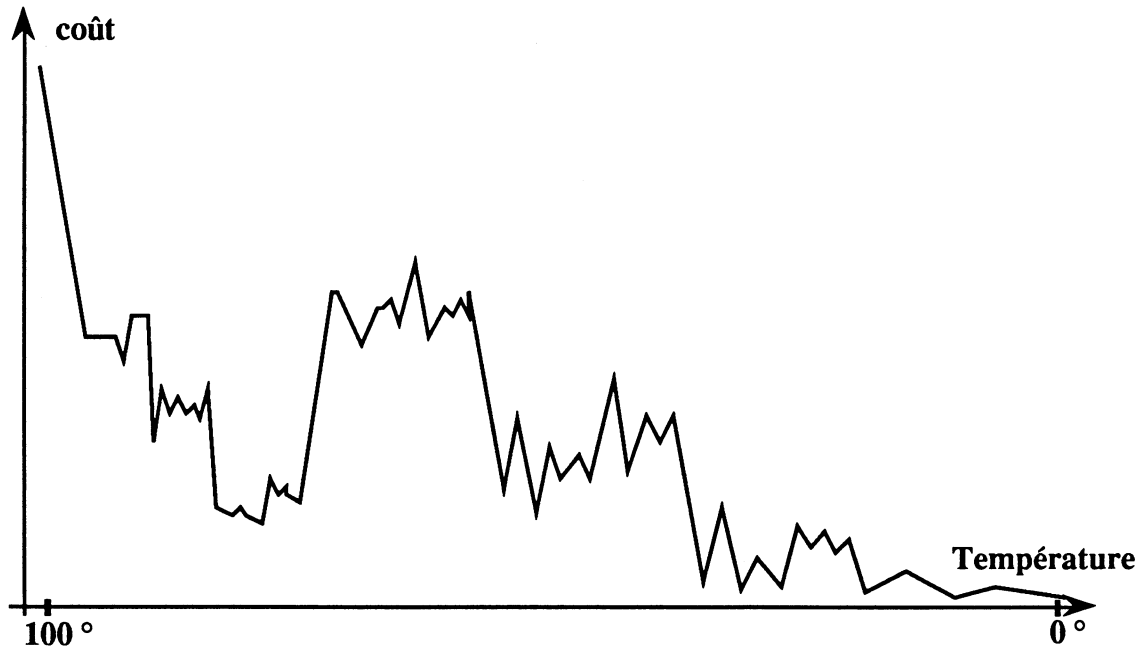


Figure 5.14. Exemple d'évolution du coût quand la température décroît.

Le seul reproche que l'on peut faire à cette méthode est l'absence de reproductibilité des recherches. En effet, la méthode étant fondée sur une bonne part de hasard, les parcours de l'ensemble des sous-graphes ne peuvent que très difficilement être reproduits (il faudrait pour cela modéliser le hasard).

Algorithmes génétiques

Le principe des algorithmes génétiques a été proposé par J.H. Holland [Holland 75]. Il est fondé sur le principe de l'évolution des espèces vivantes (les espèces évoluent de façon à s'adapter à leur environnement).

De manière formelle le principe peut s'exprimer de la façon suivante :

Soit un espace de recherche Σ de taille M^N (espace à M symboles), tout point de Σ peut être représenté par un vecteur de N symboles.

Soit une fonction d'ajustement F (*fitness function*) de Σ dans \mathbb{R} qui à tout point de Σ associe sa qualité.

Soit un ensemble initial de vecteurs, appelé population initiale.

Une phase dite de "reproduction" est mise en œuvre à l'aide d'opérateurs génétiques afin de générer de nouveaux points de Σ à partir d'anciens.

Durant cette phase certains points de la population sont remplacés afin que la taille de la population reste fixe.

Le principe fondamental des algorithmes génétiques est : "la reproduction d'un vecteur est d'autant plus probable que ce vecteur est bon".

De manière mathématique cela signifie que la probabilité de reproduction P croît avec la valeur de F :

$$\forall \sigma_1, \sigma_2 \in \Sigma \quad F(\sigma_1) > F(\sigma_2) \Rightarrow P(\sigma_1) > P(\sigma_2).$$

Ainsi décrit, l'algorithme génétique peut s'exprimer de la manière suivante :

Générer une population d'individus tirés au hasard

tant que nombre-de-génération \leq *nombre-max-de-génération* faire

Evaluation : attribuer une valeur d'ajustement à chaque individu

Sélection : construire une liste de couples d'individus, où les individus les meilleurs sont représentés plus souvent

Reproduction : appliquer les opérateurs génétiques aux couples sélectionnés ; les nouveaux individus produits constituent la nouvelle population.

fin tant que

L'individu finalement retenu comme la meilleure solution est le meilleur individu de la population de la génération *nombre-max-de-génération*.

Les opérateurs génétiques généralement utilisés lors de la reproduction sont : le crossing-over et la mutation.

L'opérateur de crossing-over, à partir de deux vecteurs, les coupe tous les deux en un point choisi aléatoirement (point de crossing-over) et échange les portions de "chromosomes" ainsi découpées (cf. figure 5.15).

L'opérateur de mutation, lui, modifie simplement une des valeurs contenues dans le vecteur (cf. figure 5.15).

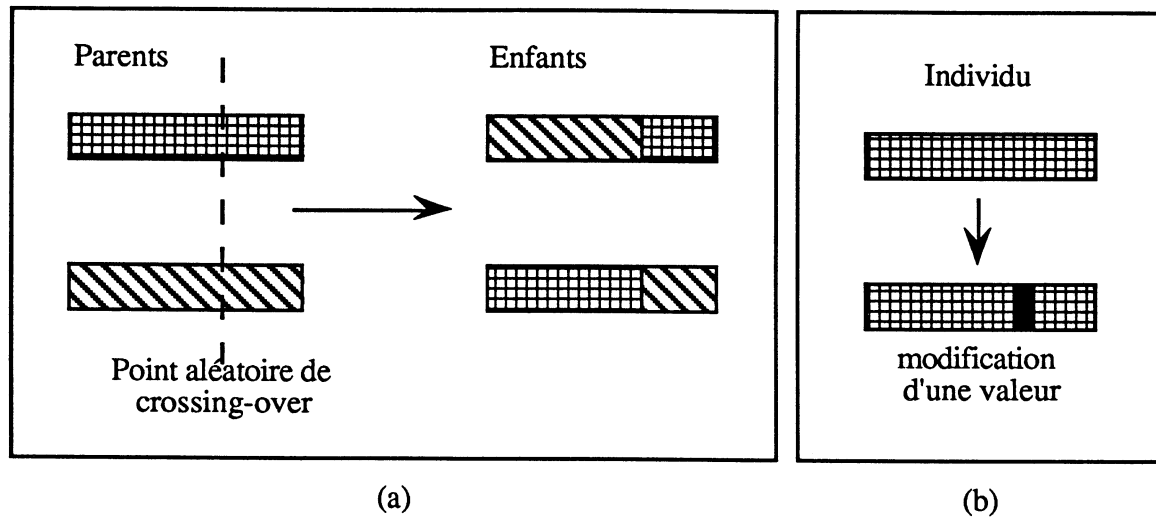


Figure 5.15. Opérateurs génétiques : (a) crossing-over, (b) mutation.

Afin de pouvoir utiliser ces opérateurs, il est nécessaire de définir deux paramètres P_c et P_m , qui représentent respectivement la probabilité d'application de l'opérateur de crossing-over et l'opérateur de mutation.

On trouve dans la littérature d'autres opérateurs, ceux-ci étant plus particuliers à certaines applications. Dans notre cas, ce sont les deux opérateurs classiques crossing-over et mutation qui sont utilisés. Leur définition est assez simple à mettre en place.

En effet, on recherche des sous-graphes à k sommets (ces sommets sont pris dans l'ensemble des sommets du graphe de mise en correspondance), le crossing-over consiste donc à échanger un nombre quelconque (tiré au hasard) de ces sommets entre deux individus. En ce qui concerne l'opérateur de mutation sa mise en œuvre permet de changer l'un des sommets constituant le sous-graphe initial. Le procédé est en fait le même que celui utilisé comme voisinage dans le cas du recuit simulé.

La fonction d'ajustement choisie est bien entendu liée à la fonction de coût définie précédemment (fonction coût d'un sous-graphe). C'est en fait une fonction inversement proportionnelle à cette fonction coût qui a été retenue, étant donné que la mise en correspondance est d'autant meilleure que le coût est faible.

Les résultats obtenus à l'aide de cette méthode sont eux aussi satisfaisants. Il faut noter ici que c'est l'ensemble des individus qui converge vers la meilleure solution. Ainsi, généralement, tous les individus finissent par être identiques, donc, n'importe quel individu peut être considéré comme la meilleure des solutions si le nombre de générations est suffisamment grand.

Les mêmes reproches que ceux fait à l'encontre de la méthode de recherche par recuit simulé peuvent être faits à cette méthode : non reproductibilité des recherches, utilisation massive du hasard, non déterminisme du procédé. Là encore, on est d'ailleurs tenté de se demander pourquoi les résultats obtenus sont si satisfaisants alors que très peu de solutions ont été parcourues.

Cette méthode a, elle aussi, l'avantage de permettre un certain contrôle du temps de calcul. Ceci est fait à l'aide du paramètre *nombre-max-de-génération* qui permet d'arrêter le processus au bout d'un certain temps.

Par rapport au recuit simulé, cette méthode semble permettre d'obtenir une décroissance plus rapide de la fonction coût au cours de l'évaluation (surtout au début) (cf. figure 5.16).

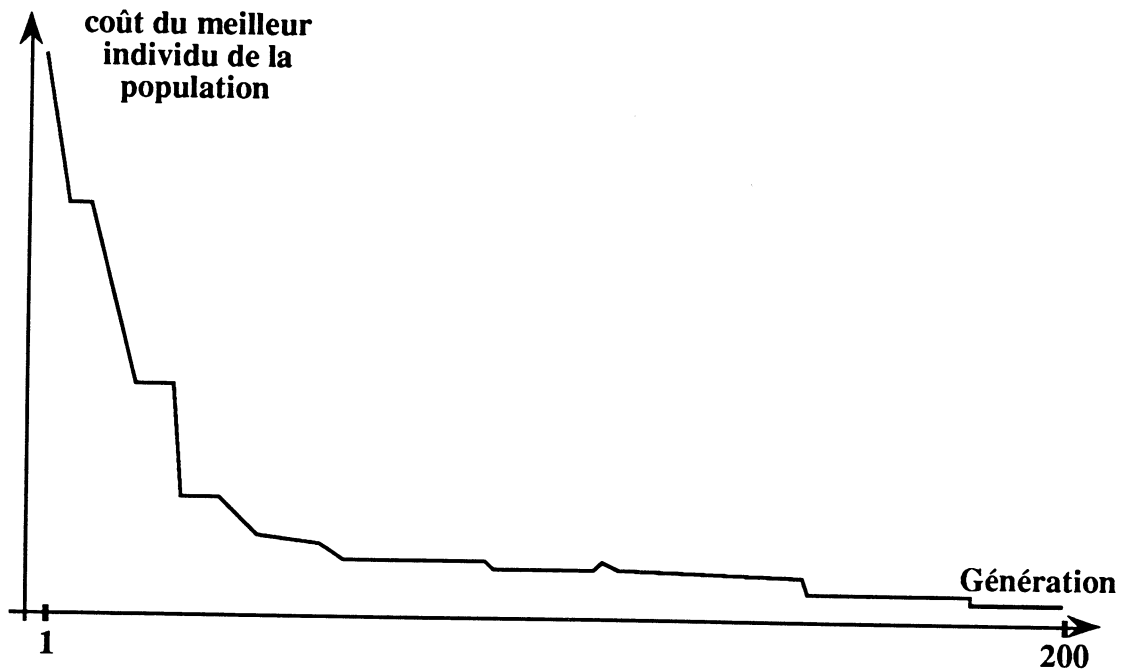


Figure 5.16. Exemple d'évaluation de la fonction coût avec les générations.

De plus, les algorithmes génétiques peuvent être parallélisés [Talbi 91], alors qu'un véritable recuit simulé parallèle est très difficile à mettre en œuvre (les travaux de parallélisation de recuit simulé se contentent généralement d'effectuer plusieurs recuits simulés en parallèle).

Par contre, l'utilisation des algorithmes génétiques étant encore plus récente que celle des algorithmes de recuit simulé, il est peut-être encore un peu tôt pour pouvoir dire dans quels cas ils ne sont pas très efficaces.

On remarque aussi que le choix des paramètres semble plus influant dans le cas des algorithmes génétiques que dans le cas du recuit simulé. En effet, dans le cas du recuit simulé, si on choisit une température de départ assez élevée, on est sûr (enfin presque) d'obtenir la solution optimale, ce qui ne semble pas toujours être le cas avec les algorithmes génétiques.

En fait, le recuit simulé semble plus fiable, et les algorithmes génétiques plus rapides. En effet, avec le recuit simulé, on est plus ou moins sûr du résultat si on ne brusque pas trop le phénomène ; avec les algorithmes génétiques le calcul est plus rapide mais on n'obtient pas à chaque fois la bonne solution.

Ceci étant dit, ces phénomènes sont ceux que j'ai observés, ils n'ont pas valeur de généralité, une plus grande expérience des algorithmes génétiques permettrait peut-être de résoudre ces problèmes.

Remarques sur les recherches heuristiques

Lors de l'utilisation des recherches heuristiques, on note que la recherche de cliques s'effectue en recherchant des sous-graphes à k sommets.

Or, dans le graphe de mise en correspondance, certains sommets ne sont adjacents qu'à moins de $k-1$ sommets. Ces sommets ne peuvent donc faire partie d'aucune des cliques recherchées. C'est pour cette raison qu'avant toute recherche de clique, un "élagage" du graphe de mise en correspondance peut être effectué. Cet élagage consiste à éliminer tous les sommets du graphe de mise en correspondance possédant moins de $k-1$ sommets.

L'expérience montre que cet élagage n'est en fait bénéfique que pour des graphes ne possédant pas trop de sommets par rapport au nombre de sommets de la clique recherchée.

Cet élagage étant très peu coûteux en temps de calcul et éliminant un nombre non négligeable de sommets dans certains cas, il est effectué systématiquement avant toute recherche.

Il faut noter ici que les résultats obtenus à l'aide des méthodes heuristiques dépendent pour beaucoup des différentes valeurs des paramètres utilisées. Ceci est à mon avis le problème majeur de ces méthodes.

De plus, ces méthodes étant assez jeunes, il est assez difficile de trouver des publications dans lesquelles des exemples de valeurs des différents paramètres sont donnés. Dans le cas du recuit simulé, beaucoup de publications en parlent mais aucune ne donne une idée précise des valeurs à donner aux différents paramètres. Dans ce cas-là d'ailleurs, en plus des problèmes de paramètres, il faut noter l'existence de nombreuses variantes, au niveau de l'implémentation, de l'algorithme, toutes semblent malgré tout donner des résultats similaires.

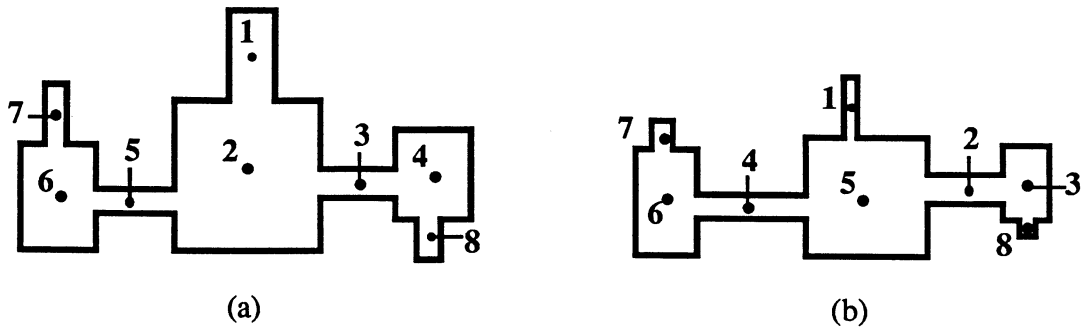
Ces problèmes de choix des valeurs des paramètres ne semblent pouvoir être résolus que par deux méthodes différentes. La première, et celle utilisée ici, consiste à faire des essais et ainsi à se forger une expérience. La seconde consisterait à faire une étude systématique de l'influence des différents paramètres, par exemple, à l'aide d'espions (comme cela est fait pour les réseaux de neurones [Paugam-Moisy 90]), en effectuant, en parallèle si possible, plusieurs essais avec des paramètres différents et en espionnant le comportement et les différents résultats obtenus.

5.2.3. Résultats de la recherche de cliques

De nombreuses recherches de cliques ont été effectuées. Malheureusement, il n'est pas possible ici de toutes les présenter.

En effet, lorsqu'un graphe de mise en correspondance est petit, la recherche de cliques est assez simple et le résultat n'a que très peu d'intérêt. Par contre, dès que la recherche de cliques devient intéressante, le graphe de mise en correspondance devient impossible à représenter.

La figure 5.17 donne un exemple de graphe de mise en correspondance et de la clique optimale trouvée. On remarque que, déjà pour deux formes aussi simples, il est impossible de représenter le graphe de mise en correspondance. Seule une représentation matricielle serait possible ; comme elle n'apporte pas grand-chose de plus au raisonnement théorique elle n'est pas présentée ici.



Le graphe de mise en correspondance est composé de :
 - 46 sommets
 - 574 arêtes.

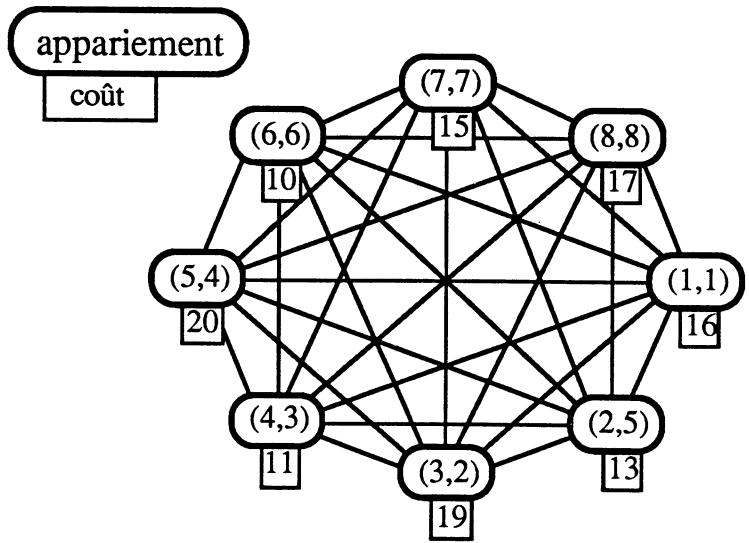


Figure 5.17. Exemple de résultat de recherche de cliques.

Le tableau suivant (figure 5.18) présente les résultats chiffrés obtenus à partir d'une base d'exemples, présentée en annexe, ils permettent d'étayer les différentes remarques faites à propos des méthodes de recherche de cliques.

<i>RECHERCHE DE CLIQUES</i> ⊗	test 1 □	test 2 □	test 3 □	test 4 □	test 5 □	test 6 □
Nombre de sommets	9	30	46	57	72	209
Nombre d'arêtes	27	270	574	712	1035	7828
Nombre de cliques	103	22704	110887	59167	139040	***
Temps CPU du calcul du nombre de cliques	0 s	26 s	194 s	130 s	445 s	***
Taille de la clique recherchée	3	6	8	9	10	17
Taille de la clique max	5	9	8	10	10	21
Temps CPU calcul clique max	0 s	0 s	1 s	0 s	2 s	123 s
Temps CPU * recuit simulé	0 s	5 s	15 s	17 s	35 s	157 s
Temps CPU * algorithme génétique	0 s	3 s	12 s	15 s	39 s **	127 s **

⊗ Calculs effectués sur une station de travail Apollo 3550.

□ La base d'exemples (test 1 à 6) est présentée en Annexe.

* Temps moyen pour l'obtention de l'optimum (clique minimisant la fonction coût).

** Plusieurs essais ont été nécessaires pour obtenir l'optimum.

*** Le calcul a été interrompu après 76 heures CPU de calcul, le nombre de cliques était alors de 22 800 000.

Figure 5.18. Recherche de cliques.

On remarque clairement que, dès que le nombre de sommets et le nombre d'arêtes du graphe de mise en correspondance deviennent grands, la recherche exhaustive de toutes les cliques incluses dans ce graphe de mise en correspondance devient rapidement très coûteuse (les temps de calcul donnés sont les temps CPU nécessaires au comptage du nombre de cliques).

Il faut noter que chaque fois que la taille de la clique cherchée est différente de la taille de la clique max (clique avec le plus de sommets), la méthode consistant à rechercher la clique maximale au lieu de rechercher la clique, minimisant la fonction coût, de taille donnée, donne des résultats erronés. Par contre, quand ce n'est pas le cas cette méthode est très rapide (voir colonne *test 5*).

Les temps de calcul obtenus avec les méthodes de recuit simulé et les algorithmes génétiques vont en croissant avec la taille des graphes de mise en correspondance. Ceci est tout à fait normal. Dans le cas du recuit simulé, plus le graphe est grand, plus la température de départ doit être élevée si on veut être sûr d'obtenir le bon résultat. Dans le cas de l'algorithme génétique, ce sont le nombre d'individus de la population et le nombre de générations nécessaires pour trouver l'optimum qui augmentent.

Dans les deux cas, de plus, le temps nécessaire pour effectuer une itération augmente aussi avec la taille du graphe de mise en correspondance. En effet, le calcul du voisinage d'une solution réalisable pour le recuit simulé d'une part, la sélection des meilleurs individus et l'application des opérateurs génétiques pour l'algorithme génétique d'autre part, nécessitent un plus grand nombre de calculs quand le graphe de mise en correspondance est grand que quand celui-ci est formé de peu de sommets et de peu d'arêtes.

Malgré quelques petits problèmes, les méthodes heuristiques permettent d'obtenir rapidement de très bons résultats. Ces méthodes semblent donc promises à un bel avenir pour la recherche de cliques dans un graphe de mise en correspondance. Ces méthodes devraient d'ailleurs pouvoir être utilisées dans d'autres applications où il est nécessaire de faire une recherche de cliques optimales dans un graphe pondéré.

5.3. Résultats de la mise en correspondance de graphes de la ligne médiane

Les résultats de la mise en correspondance de graphes de la ligne médiane sont très satisfaisants. Deux exemples sont donnés en annexe, tous les résultats ne pouvant être présentés ici.

On remarque cependant sur les différents résultats que la mise en correspondance obtenue a une signification évidente au niveau de la forme. En effet, les différentes composantes de la forme sont appariées de la manière qui semble la plus naturelle.

Comme la mise en correspondance a été effectuée sur les graphes filtrés tous les sommets n'ont pas été mis en correspondance. Ainsi, c'est tout d'abord une phase de propagation des appariements qui est mise en œuvre avant d'effectuer l'inférence de graphes proprement dite.

Chapitre 6

Inférence structurelle de graphes de la ligne médiane

La mise en correspondance de certains sommets des graphes de la ligne médiane significatifs des formes ayant été réalisée, il faut maintenant passer à l'inférence de graphes de la ligne médiane. Cette inférence permettra ensuite de régénérer des coupes intermédiaires. Ces coupes seront alors empilées pour reconstruire le volume tridimensionnel.

Cette inférence de graphes de la ligne médiane est composée de deux phases.

La première consiste, à partir des quelques sommets mis en correspondance, à propager les appariements afin que chacun des sommets des graphes de la ligne médiane soient appariés. La seconde phase consiste alors, à partir des sommets appariés, à inférer un graphe de la ligne médiane intermédiaire.

6.1. Propagation des appariements et résultats finals de la mise en correspondance de graphes de la ligne médiane

6.1.1. Propagation des appariements

Comme on l'a vu dans le paragraphe 5.2.1.2 tous les sommets des graphes de la ligne médiane ne sont pas utilisés pour la construction du graphe de mise en correspondance, donc tous les sommets des graphes de la ligne médiane ne sont pas appariés.

Afin d'apparier les sommets qui ne le sont pas encore une phase de propagation des appariements est mise en œuvre. Cette phase est très importante pour la suite du processus de reconstruction car, si elle est très précise, elle permettra de transformer le processus d'inférence de graphes de la ligne médiane en un processus très local d'inférence de sommets et ainsi de reconstruire plus facilement les coupes intermédiaires.

Cette phase de propagation des appariements utilise à la fois les appariements obtenus lors de la mise en correspondance et la structure des graphes de la ligne médiane.

La propagation des appariements consiste, à partir de sommets des graphes de la ligne médiane déjà mis en correspondance, à propager les appariements sur les sommets non encore mis en correspondance. Cette propagation est effectuée par parcours des graphes de la ligne médiane en utilisant les appariements déjà établis.

Les sommets qui ne sont pas pris en compte lors de la construction du graphe de mise en correspondance sont les proéminences larges et les proéminences strictes peu déformantes ($\chi \geq \text{seuil}$). Deux stratégies différentes ont été développées pour ces deux cas.

6.1.1.1. Appariement des proéminences larges

On sait que les proéminences larges servent de chemin de connexion entre une proéminence stricte et le sommet déformé (sommet principal ou de rétrécissement) (cf. figure 2.26). Ainsi, si les proéminences strictes et les sommets déformés sont mis en correspondance de manière correcte, l'appariement des proéminences larges se fait en mettant en correspondance les "chemins" reliant les sommets déformés aux proéminences strictes.

Cette mise en correspondance se fait de manière assez évidente dans le cas où tous les sommets se trouvent sur une même branche et que le nombre de proéminences larges est le même (cf. figure 6.1).

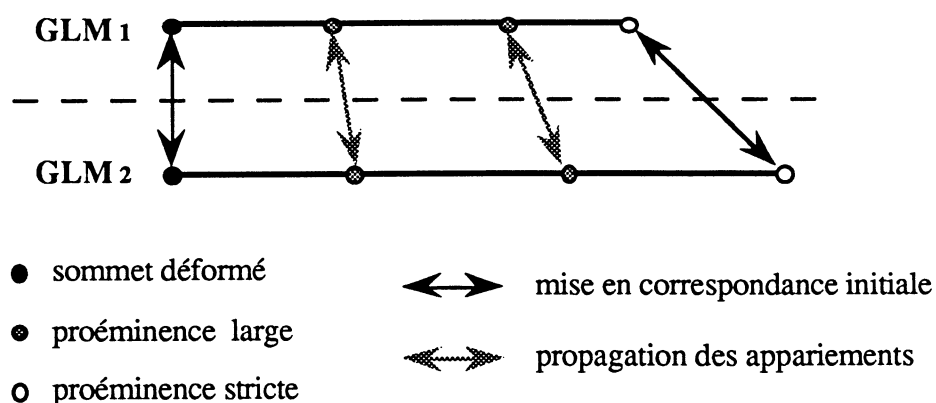
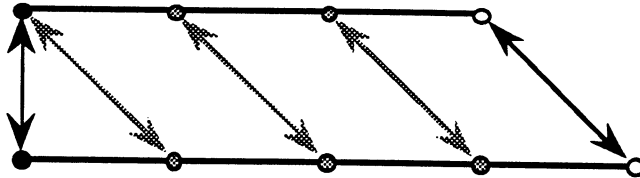


Figure 6.1. Appariement de proéminences larges : cas simple.

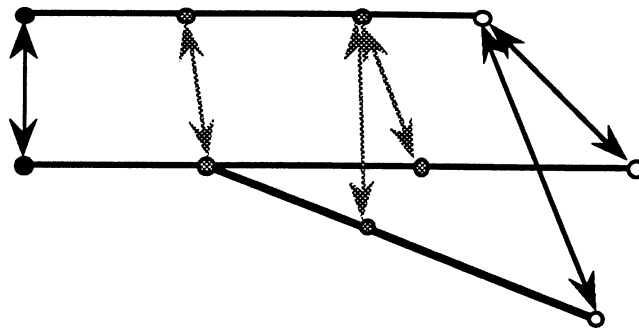
Lorsque les nombres de proéminence larges sont différents dans les deux graphes de la ligne médiane, il faut faire un choix pour les appariements. Une "accumulation", soit sur le sommet déformé, soit sur la proéminence stricte est nécessaire. Etant donné que les surfaces codées par les proéminences larges sont souvent beaucoup plus importantes que celles couvertes par les proéminences strictes, il est préférable de faire l'accumulation sur le sommet déformé (cf. figure 6.2). Ainsi, certaines proéminences larges se trouvent mises en correspondance avec des sommets principaux ou de rétrécissement.



- sommet déformé
 - ⊙ proéminence large
 - proéminence stricte
- ⇔ mise en correspondance initiale
 ⇔ propagation des appariements

Figure 6.2. : Appariement de proéminence larges pour des nombres de proéminences larges différents.

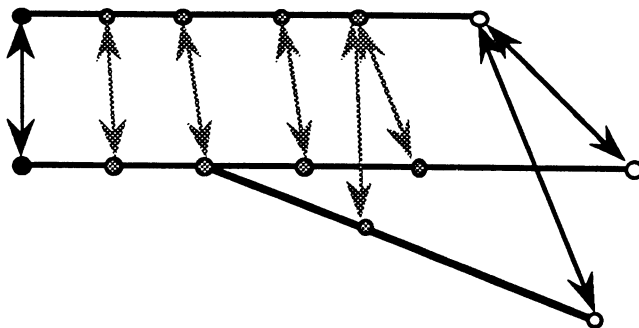
Le problème de propagation des appariements se complique lorsque, par exemple, une proéminence stricte est mise en correspondance avec deux autres (cas de "polygamie"). Dans ce cas, ce sont deux "chemins" qui doivent être mis en correspondance avec un même autre. Tout se passe assez bien lorsque les deux branches sont composées d'un même nombre de sommets (cf. figure 6.3).



- sommet déformé
 - ⊙ proéminence large
 - proéminence stricte
-
- ←→ mise en correspondance initiale
 ←→ propagation des appariements

Figure 6.3. Propagation des appariements : appariements multiples.

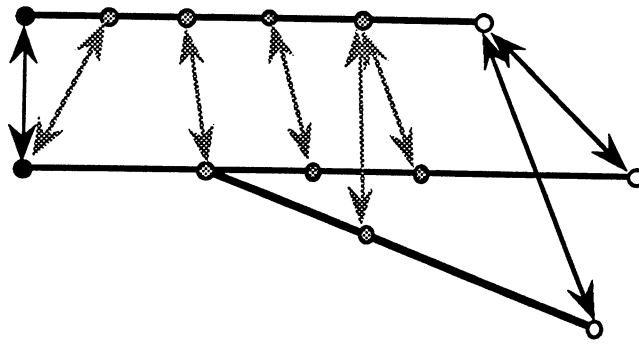
Par contre, cela se complique lorsque les deux chemins sont de longueurs différentes. Dans ce cas c'est le chemin le plus long des deux qui détermine la base des appariements, les autres appariements sont ensuite ajoutés sur cette base (cf. figure 6.4).



- sommet déformé
 - ⊙ proéminence large
 - proéminence stricte
-
- ←→ mise en correspondance initiale
 ←→ propagation des appariements

Figure 6.4. Propagation des appariements : appariements multiples avec chemins de longueurs différentes.

Ces divers cas pouvant se combiner, certaines propagations peuvent être assez complexes à mettre en œuvre (cf. figure 6.5).



- sommet déformé
 - ⊗ proéminence large
 - proéminence stricte
- ⇔ mise en correspondance initiale
 ⇨ propagation des appariements

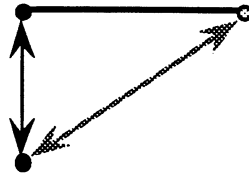
Figure 6.5. Propagation des appariements : cas plus complexe.

Cette propagation pour les proéminences larges, bien que parfois complexe, est toujours possible. Elle donne d'ailleurs de très bons résultats dont la signification au niveau des formes est tout à fait intéressante.

6.1.1.2. Appariement des proéminences strictes peu déformantes

De même que dans le cas des proéminences larges, les proéminences strictes peu déformantes ne sont pas utilisées lors de la construction du graphe de mise en correspondance. Ces proéminences strictes ne peuvent donc pas être appariées lors du calcul de la clique. Une "propagation" est donc là encore nécessaire afin d'apparier ces proéminences strictes.

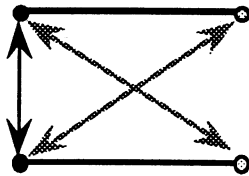
Etant donné que ces proéminences sont peu déformantes, donc peu importantes par rapport à la forme globale, j'ai choisi de les apparier simplement avec le sommet auquel le sommet déformé est apparié (cf. figure 6.6).



- sommet déformé \longleftrightarrow mise en correspondance initiale
- ⊙ proéminence stricte peu déformante \longleftrightarrow propagation des appariements

Figure 6.6. Appariement des proéminences strictes peu déformantes.

Le choix de faire ce type d'appariement peu aboutir parfois à des situations paradoxales (cf. figure 6.7).



- sommet déformé \longleftrightarrow mise en correspondance initiale
- ⊙ proéminence stricte peu déformante \longleftrightarrow propagation des appariements

Figure 6.7. Situation paradoxale lors de l'appariement des proéminences strictes peu déformantes.

En effet, dans ce cas on aurait plutôt voulu que les proéminences strictes soient mises en correspondance l'une avec l'autre.

Malgré cela, ces situations ne sont que très peu préjudiciables pour la suite du processus de reconstruction, étant donné les faibles contributions à la forme apportées par les proéminences strictes considérées.

Ces deux propagations des appariements permettent finalement de mettre en correspondance la totalité des sommets des graphes de la ligne médiane alors que le nombre de sommets initialement retenus pour faire la mise en correspondance initiale reste raisonnable (cf. figure 6.8). Ceci est lié au fait que, dans bon nombre de graphes de la ligne médiane, se sont les proéminences larges qui constituent le gros de l'effectif des sommets.

<i>MISE EN CORRESPONDANCE</i>	test 1	test 2	test 3	test 4	test 5	test 6
Nombre de sommets de GLM ₁	3	6	8	10	48	47
Nombre de sommets retenus pour la mise en correspondance initiale	3	6	8	9	10	17
Taux de réduction	0 %	0 %	0 %	10 %	79 %	64 %
Nombre de sommets de GLM ₂	3	6	8	10	46	47
Nombre de sommets retenus pour la mise en correspondance initiale	3	6	8	9	10	17
Taux de réduction	0 %	0 %	0 %	10 %	78 %	64 %
Nombre de sommets initialement appariés	3	6	8	9	10	17
Nombre de sommets mis en correspondance après propagation	3	6	8	10	48	47
Taux de mise en correspondance	100 %	100 %	100 %	100 %	100 %	100 %

Figure 6.8. Mise en correspondance.
Les images utilisées (test 1 à 6) sont données en annexe.

6.1.2. Résultats de la mise en correspondance de graphes de la ligne médiane

Les résultats de la mise en correspondance de graphes de la ligne médiane obtenus sont très satisfaisants. Ces résultats permettent en outre de valider le procédé de mise en correspondance de graphes de la ligne médiane utilisé. Des exemples de résultats sont donnés en annexe.

Ce sont ces mises en correspondance de graphes de la ligne médiane qui sont utilisées pour effectuer l'inférence de graphes de la ligne médiane.

6.2. Inférence de graphes de la ligne médiane

Les sommets des deux graphes de la ligne médiane ayant été mis en correspondance les uns avec les autres le processus d'inférence de graphes de la ligne médiane revient donc à un processus d'inférence au niveau des sommets. Les différents cas de mise en correspondance peuvent donc être traités indépendamment les uns des autres.

Avant de développer ces différents cas, il convient de préciser la manière selon laquelle est construit le graphe de la ligne médiane inféré. Ce graphe de la ligne médiane est obtenu à partir de l'un des deux graphes donnés (GLM_1 par exemple) en le modifiant en fonction des appariements trouvés lors de la mise en correspondance. L'inférence est donc simplement un processus modifiant l'un des graphes afin qu'il soit plus "proche" de l'autre. Les principes proposés dans la suite correspondent à l'inférence d'un seul graphe de la ligne médiane intermédiaire. La stratégie plus globale d'inférence est décrite en § 6.3.

La structure du graphe de la ligne médiane étant assez complexe, les différentes modifications du graphe initial opérées afin d'obtenir le graphe de la ligne médiane intermédiaire sont données de manière plutôt qualitative. Seuls les principes des différents algorithmes sont donnés, une description plus précise de ceux-ci serait trop contraignante ici, ces algorithmes nécessitant souvent l'emploi de structures assez lourdes à manipuler. Ces descriptions sont ensuite étayées par des exemples simples, résultats de l'implémentation faite. Les résultats plus réalistes des combinaisons des différents cas sont donnés en annexe.

On dénombre vingt possibilités d'appariements entre les sommets des graphes de la ligne médiane, dont dix sont les symétriques des dix autres. Ainsi, dix types d'appariements sont étudiés, ils sont déterminés par les types des sommets mis en jeu.

Les dix appariements possibles sont :

- sommet principal \leftrightarrow sommet principal
- sommet principal \leftrightarrow sommet de rétrécissement
- sommet principal \leftrightarrow sommet de proéminence stricte
- sommet principal \leftrightarrow sommet de proéminence large
- sommet de rétrécissement \leftrightarrow sommet de rétrécissement
- sommet de rétrécissement \leftrightarrow sommet de proéminence stricte
- sommet de rétrécissement \leftrightarrow sommet de proéminence large
- sommet de proéminence stricte \leftrightarrow sommet de proéminence stricte
- sommet de proéminence stricte \leftrightarrow sommet de proéminence large
- sommet de proéminence large \leftrightarrow sommet de proéminence large.

6.2.1. Appariement sommet principal \leftrightarrow sommet principal

C'est le cas le plus simple à aborder. On le rencontre chaque fois que deux composantes principales sont mises en correspondance (cf. figure 6.9).

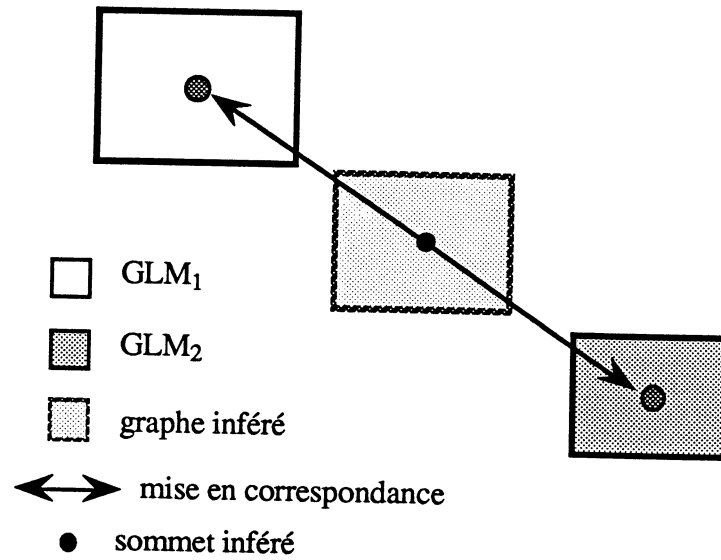


Figure 6.9. Inférence dans le cas d'un appariement sommet principal ↔ sommet principal.

Dans ce cas le graphe intermédiaire qu'on souhaite inféré doit représenter une forme dont la surface est intermédiaire entre celle de chacune des composantes, et occupant une position dans l'image moyenne par rapport aux deux composantes.

Les informations disponibles dans le graphe de la ligne médiane pour chacun des sommets principaux sont : la taille des boules codées, leur nombre et les coordonnées des centres de chacune d'elles.

Soit s_1 le sommet principal de GLM₁ et s_2 celui de GLM₂.

On note t_1 (respectivement t_2) la taille des boules codées par s_1 (respectivement par s_2).

On note n_1 (respectivement n_2) leur nombre.

On note (x_1^i, y_1^i) (respectivement (x_2^j, y_2^j)) les coordonnées des centres de chacune d'elles.

Le sommet du graphe de la ligne médiane intermédiaire inféré est obtenu à partir de s_1 :

- en remplaçant t_1 par $[t_1 + ((t_2 - t_1) \text{div } 2)]$

- et (x_1^i, y_1^i) par $(x_1^i, y_1^i) + \left(\left(\left[\sum_{j=1}^{n_2} \frac{(x_2^j, y_2^j)}{n_2} \right] - \left[\sum_{i=1}^{n_1} \frac{(x_1^i, y_1^i)}{n_1} \right] \right) \text{div } 2 \right)$.

6.2.2. Appariement sommet principal \leftrightarrow sommet de proéminence stricte

Le cas se produit chaque fois que dans l'une des formes, il y a une proéminence stricte alors qu'il n'y en a pas dans l'autre (cf. figure 6.10).

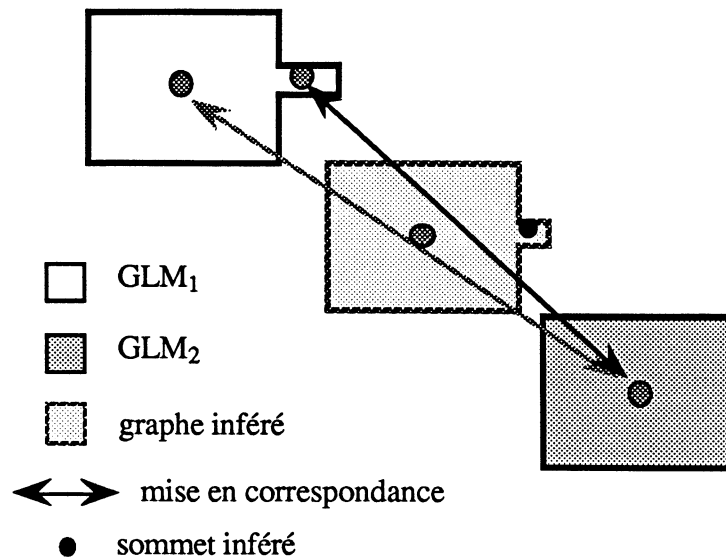


Figure 6.10. Inférence dans le cas d'un appariement sommet principal \leftrightarrow sommet de proéminence stricte.

La proéminence stricte étant fortement liée au sommet qu'elle déforme (sommet principal où de rétrécissement), elle subit donc la même translation que lui. Comme la proéminence stricte doit disparaître de moitié, c'est sa contribution à la forme qui va être divisée par deux (par l'intermédiaire de la taille et du nombre des boules codées).

Soit p_1 le sommet déformé de GLM₁, soit p_2 le sommet de GLM₂ mis en correspondance avec p_1 et s_1 le sommet de proéminence stricte de GLM₁.

On note (px_1, py_1) (respectivement (px_2, py_2)) le centre de gravité de p_1 (respectivement de p_2).

On note t_1 la taille des boules codées par s_1 et (x_1^i, y_1^i) les coordonnées des centres de chacune d'elles.

Le sommet du graphe de la ligne médiane intermédiaire inféré est obtenu à partir de s_1 :

- en remplaçant t_1 par $t_1 - (t_1 \text{ div } 2)$

- et en remplaçant (x_1^i, y_1^i) par

$$(x_1^i, y_1^i) + \left(\left((px_1, py_1) - (px_2, py_2) \right) \text{div } 2 \right) + \varepsilon (s_1, p_1, p_2)$$

où $\varepsilon (s_1, p_1, p_2)$ représente l'ajustement nécessaire afin, de ne pas créer de déconnexion entre la composante codée par le sommet déformé et la composante codée par le sommet de proéminence large dans tous les cas où cela a tendance à se produire, et afin que la contribution de la proéminence stricte à la forme soit réduite de moitié.

Ces déconnexions ont tendance à apparaître chaque fois que les tailles des sommets déformés sont modifiées au cours de l'inférence. Dans ce cas-là des calcul de distances entre boules et de différences de taille sont nécessaires pour établir la valeur de $\varepsilon (s_1, p_1, p_2)$ permettant d'éviter la déconnexion.

6.2.3. Appariement sommet principal \leftrightarrow sommet de proéminence large

Ce cas peut être considéré comme la généralisation du cas précédent. Ici, simplement une proéminence large est nécessaire à la connexion entre le sommet déformé et la proéminence stricte (cf. figure 6.11).

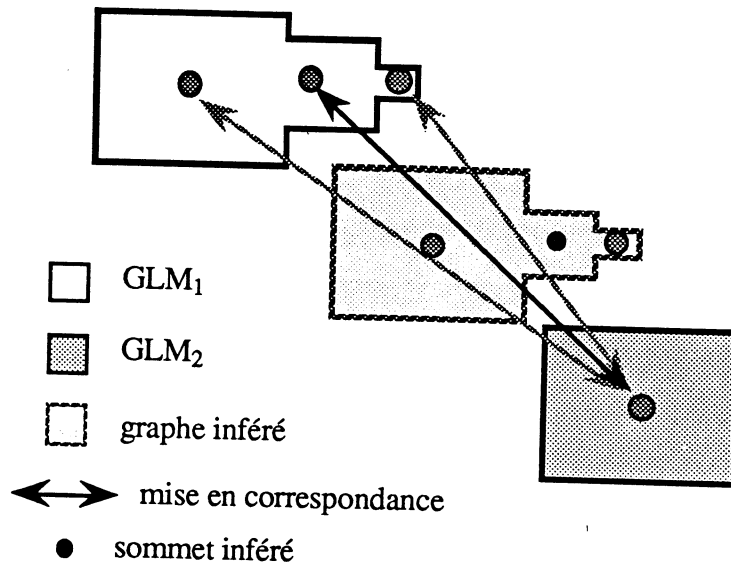


Figure 6.11. Inférence dans le cas d'un appariement sommet principal \leftrightarrow sommet de proéminence large.

Dans ce cas-là c'est la contribution de la proéminence large à la forme qui doit être divisée par deux. La proéminence est transformée avec le même genre de formules que celles utilisées dans le cas d'une proéminence stricte.

6.2.4. Appariement sommet de proéminence stricte \leftrightarrow sommet de proéminence stricte

On rencontre ce cas chaque fois que les deux formes comportent une déformation et que ces déformations se situent dans la même région par rapport à la composante déformée (cf. figure 6.12).

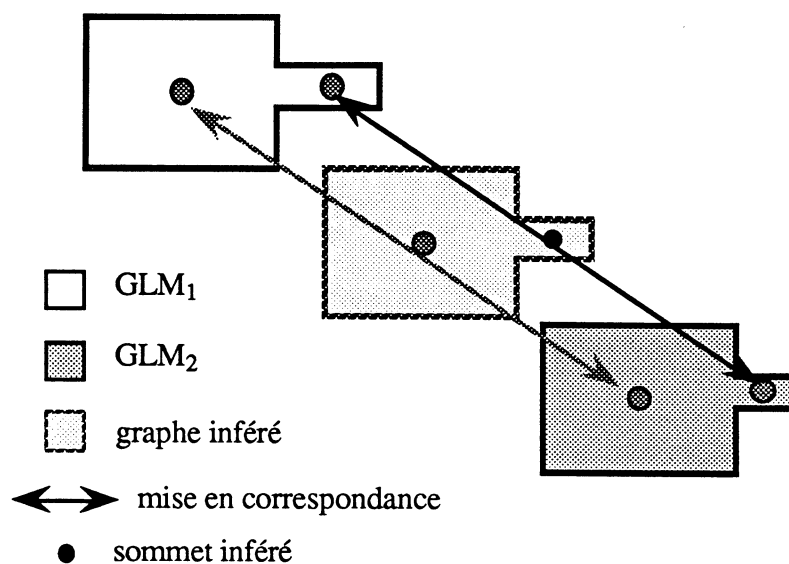


Figure 6.12. Inférence dans le cas d'un appariement sommet de proéminence stricte \leftrightarrow sommet de proéminence stricte.

La proéminence inférée doit avoir une contribution à la forme moyenne par rapport aux proéminences des graphes initiaux. Cette proéminence doit elle aussi subir les déplacements liés au sommet déformé. Un ajustement des positions des boules codées par la proéminence stricte doit être effectué, afin qu'il ne se produise pas de déconnexion entre la composante déformée et la déformation, et que la contribution à la forme soit bien divisée par deux. Cette déconnexion a surtout tendance à se produire lorsque la taille de la composante principale est modifiée lors de l'inférence sommet principal \leftrightarrow sommet principal.

Soit p_1 le sommet déformé de GLM_1 , soit p_2 le sommet de GLM_2 mis en correspondance avec p_1 , soit s_1 le sommet de proéminence stricte de GLM_1 et s_2 le sommet de proéminence stricte de GLM_2 apparié avec s_1 .

On note (px_1, py_1) (respectivement (px_2, py_2)) le centre de gravité de p_1 (respectivement de p_2).

On note t_1 (respectivement t_2) la taille des boules codées par s_1 (respectivement par s_2) et (x_1^i, y_1^i) les coordonnées des centres des boules codées par s_1 .

La proéminence stricte inférée est construite à partir de s_1 :

- en remplaçant t_1 par $(t_1 + t_2) \text{ div } 2$

- et en remplaçant (x_1^i, y_1^i) par

$$(x_1^i, y_1^i) + \left(\left((px_1, py_1) - (px_2, py_2) \right) \text{ div } 2 \right) + \epsilon(s_1, s_2, p_1, p_2)$$

où $\epsilon(s_1, s_2, p_1, p_2)$ représente la fonction d'ajustement.

6.2.5. Appariement sommet de proéminence large \leftrightarrow sommet de proéminence large

Ce cas est en fait la généralisation du cas précédent. Dans ce cas, une proéminence large est nécessaire pour effectuer la connexion avec la proéminence stricte et le sommet déformé, et cela dans chacun des deux graphes (cf. figure 6.13).

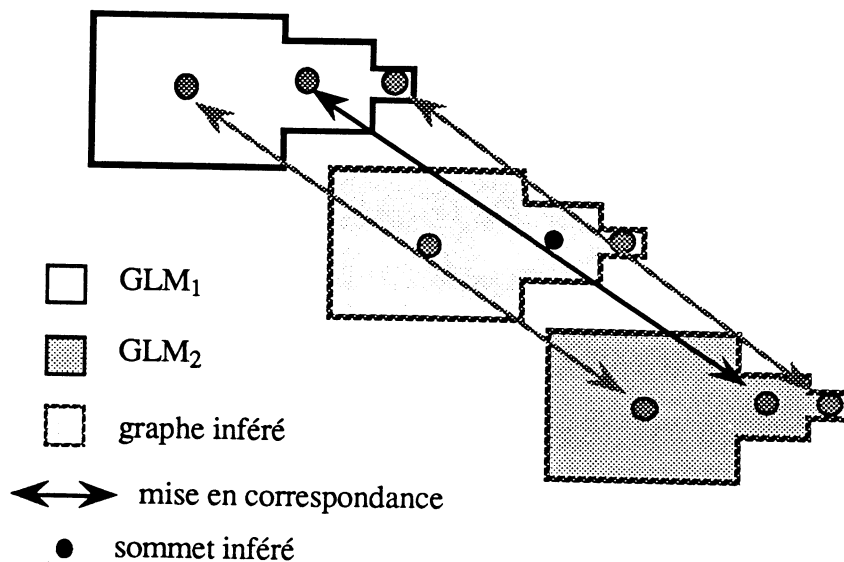


Figure 6.13. Inférence dans le cas d'un appariement sommet de proéminence large \leftrightarrow sommet de proéminence large.

Ici, la contribution à la forme globale de la proéminence large inférée doit être intermédiaire entre celle de la proéminence stricte du premier graphe et celle du second graphe. Ceci est fait en utilisant le même type de formalisme que dans le cas d'appariements entre proéminences strictes.

6.2.6. Appariement sommet de proéminence stricte \leftrightarrow sommet de proéminence large

Etant donné la méthode mise en œuvre pour la propagation des appariements, ce cas ne peut se rencontrer qu'assez rarement. Il ne se produit en fait que dans le cas particulier où deux proéminences strictes proviennent d'une même proéminence large dans l'un des graphes, et que dans l'autre graphe l'une des proéminences a disparu (cf. figure 6.14).

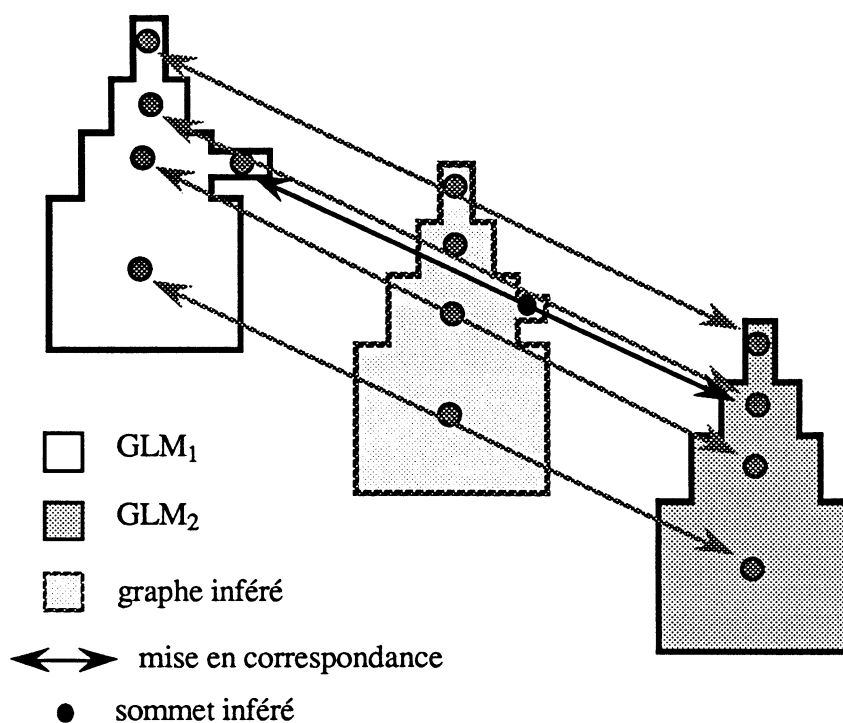


Figure 6.14. Inférence dans le cas d'un appariement sommet de proéminence stricte \leftrightarrow sommet de proéminence large.

Dans ce cas, la contribution de la proéminence stricte doit être divisée par deux. Ceci est fait en divisant par deux la taille des boules codées par la proéminence stricte, de la même manière que cela a été fait dans les cas précédents. Les centres des boules doivent aussi être déplacés afin de subir les déplacements de la proéminence large à laquelle la proéminence stricte est attachée et afin de ne pas créer de déconnexion.

6.2.7. Appariement sommet de rétrécissement \leftrightarrow sommet de rétrécissement

On rencontre ce type d'appariement chaque fois que deux formes présentant un rétrécissement sont mises en correspondance (cf. figure 6.15).

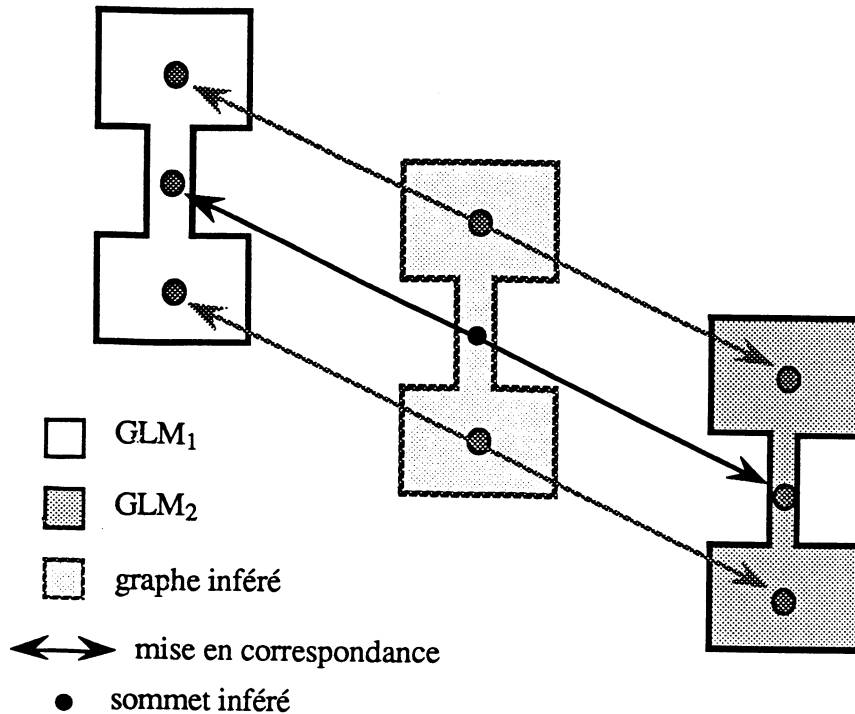


Figure 6.15. Inférence dans le cas d'un appariement sommet de rétrécissement \leftrightarrow sommet de rétrécissement.

Le sommet de rétrécissement inféré doit être une "moyenne" entre les deux sommets de rétrécissement mis en correspondance. Cette inférence utilise à la fois une modification des tailles des boules codées et un déplacement des centres des boules afin de "suivre" les déplacements des sommets principaux.

6.2.8. Appariement sommet de rétrécissement \leftrightarrow sommet principal

Ce cas se produit lorsqu'une forme possédant un rétrécissement est mise en correspondance avec une forme n'en possédant pas (cf. figure 6.16).

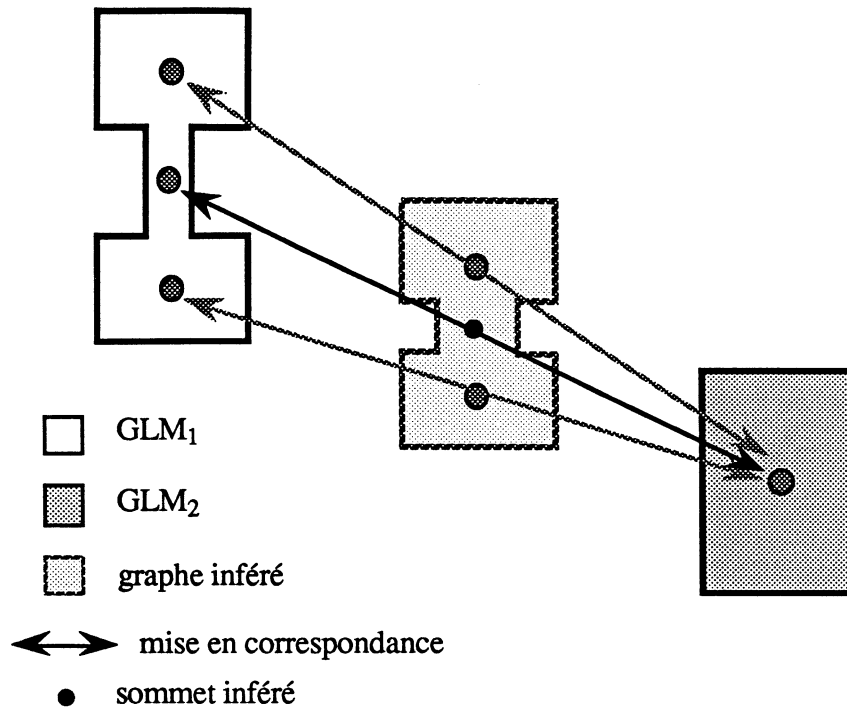


Figure 6.16. Inférence dans le cas d'un appariement sommet de rétrécissement ↔ sommet principal.

Le sommet de rétrécissement inféré doit avoir une largeur intermédiaire entre celle du sommet de rétrécissement et celle du sommet principal. Cette déformation permet que le rétrécissement se fonde de plus en plus avec la forme au fur et à mesure que les coupes sont inférées.

6.2.9. Appariement sommet de rétrécissement ↔ sommet de proéminence stricte

Ce cas se produit chaque fois que les deux formes initiales possèdent un rétrécissement et que l'un des rétrécissements comporte une déformation alors que l'autre n'en a pas (cf. figure 6.17).

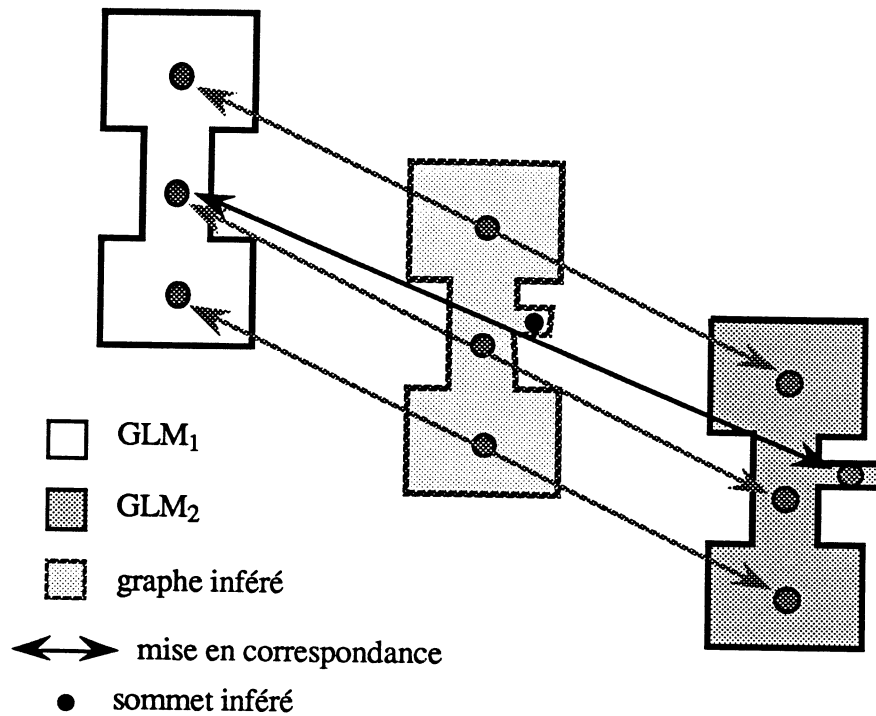


Figure 6.17. Inférence dans le cas d'un appariement sommet de rétrécissement ↔ sommet de proéminence stricte.

Dans ce cas, la proéminence stricte inférée doit contribuer deux fois moins au rétrécissement que la proéminence stricte le fait dans la forme initiale. Cette modification de la contribution de la proéminence stricte s'effectue de la même manière que lors de l'appariement d'un sommet principal avec un sommet de proéminence stricte (cf. § 6.2.2).

6.2.10. Appariement sommet de rétrécissement ↔ sommet de proéminence large

Il s'agit de la simple généralisation du cas précédent. Dans ce cas, un sommet de proéminence large est nécessaire pour assurer la connexion entre le sommet de rétrécissement et le sommet de proéminence stricte (cf. figure 6.18).

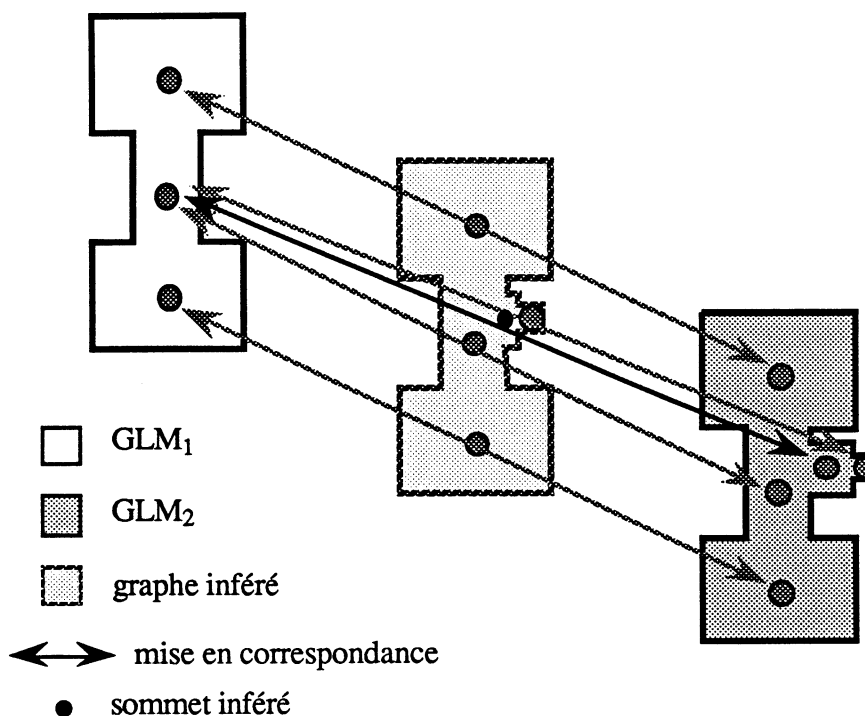


Figure 6.18. Inférence dans le cas d'un appariement sommet de rétrécissement ↔ sommet de proéminence large.

La résolution de ce cas se fait d'une manière similaire au cas où le sommet déformé est un sommet principal (appariement sommet principal ↔ sommet de proéminence large).

A partir de ces dix sortes d'appariements possibles, de nombreuses combinaisons peuvent être effectuées, chacune d'elles est traitée à l'aide des dix cas précédemment cités.

A ces dix "cas de base" et à leurs combinaisons, il convient d'ajouter deux cas particuliers qui permettent de résoudre les cas d'apparition de branches et trous. Ces deux cas correspondent à des appariements bien distincts : appariement multiple "sommet principal ↔ sommets principaux" et appariement multiple "sommet principal ↔ sommets principaux", sommets de rétrécissement.

6.2.11. Appariement multiple sommet principal ↔ sommets principaux

C'est l'appariement typique qui se produit dans le cas où un branchement apparaît. En effet, la première forme est composée de deux composantes connexes (ou plus encore) et la seconde forme d'une seule composante connexe. Ainsi, lors de la mise en correspondance l'une et l'autre des deux composantes connexes de la première forme sont appariées indépendamment avec la composante de la seconde forme (cf. figure 6.19).

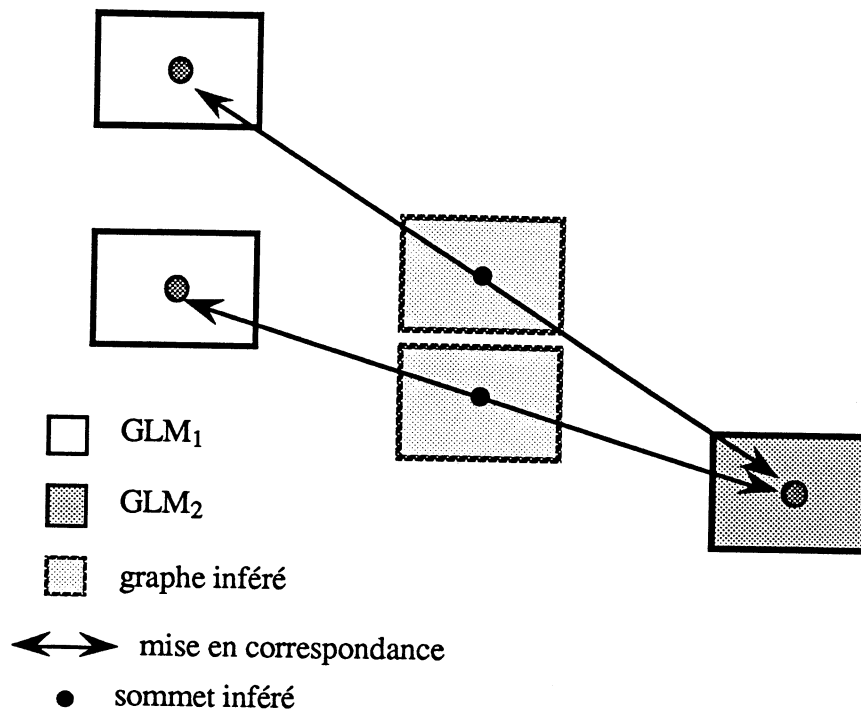


Figure 6.19. Inférence dans le cas d'un appariement multiple
sommet principal ↔ sommets principaux.

Dans ce cas, les différentes composantes connexes sont traitées indépendamment les unes des autres. Le graphe de la ligne médiane inféré est donc composé de deux composantes connexes (comme GLM₁ dans l'exemple). Les différentes composantes connexes étant toutes appariées avec une même composante connexe de l'autre graphe, chaque composante connexe du graphe inféré est donc le résultat de l'inférence entre une des composantes connexes du premier graphe et la composante connexe du second. Ces différentes composantes connexes "se rapprochent" afin de "converger" vers la composante connexe du second graphe.

Ce cas-là peut en fait être considéré comme la réunion de combinaisons de cas de base, étant donné que pour l'inférence de chaque composante connexe on utilise une combinaison des dix cas de base.

Le cas des "branches" est ainsi résolu de manière totalement transparente par la méthode d'inférence proposée. Ceci est l'un des très grands avantages procurés par cette méthode par rapport aux méthodes par triangulation pour lesquelles des traitements particuliers doivent être mis en œuvre (calculs d'intersections de surface par exemple) dans ce cas.

6.2.12. Appariement multiple sommet principal \leftrightarrow sommets principaux, sommets de rétrécissement

Ce genre d'appariement apparaît chaque fois que l'une des deux formes comporte un trou alors que l'autre n'en a pas. En effet, toute forme comportant un trou a un graphe de la ligne médiane comportant un cycle [Rolland 89a]. Ce cycle est composé d'une succession de sommets principaux et de sommets de rétrécissement (et éventuellement de proéminences larges). Tous les sommets du cycle sont donc mis en correspondance avec le sommet de la composante de l'autre graphe (cf. figure 6.20).

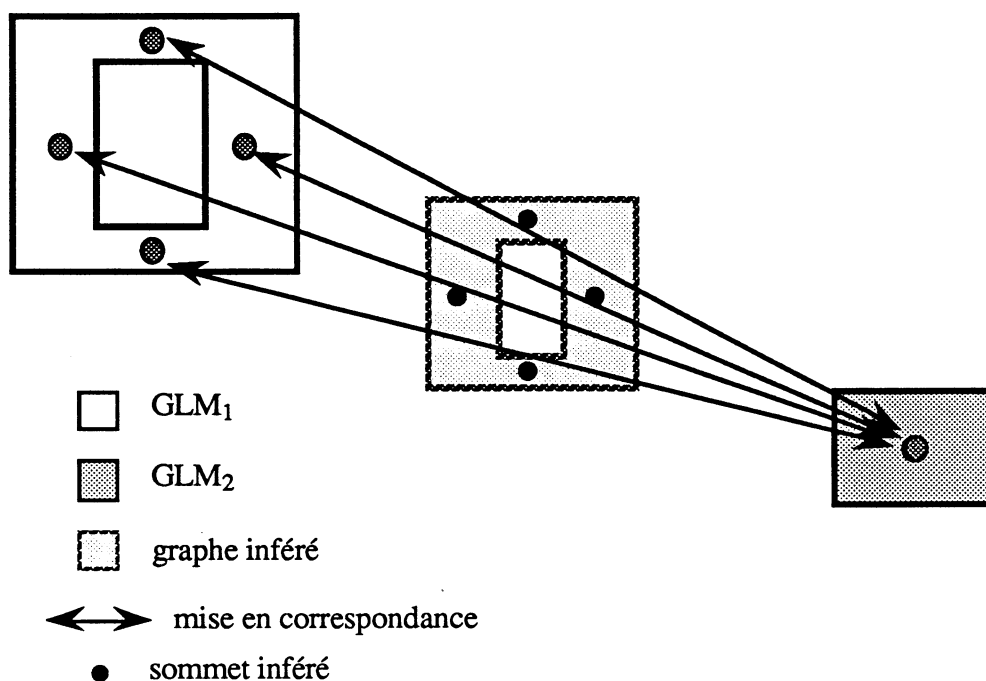


Figure 6.20. Inférence dans le cas d'un appariement multiple sommet principal \leftrightarrow sommets principaux, sommets de rétrécissement.

Dans ce cas, l'inférence du graphe de la ligne médiane intermédiaire se fait en effectuant une inférence pour chacun des sommets du cycle. Les sommets du cycle se déplacent pour, dans un premier temps, reboucher le trou et finalement se confondre avec le sommet du graphe de la seconde forme.

Le traitement de ce cas est plus délicat que le précédent, malgré tout, la méthode générale d'inférence de forme utilisée reste valable, compte tenu de certaines contraintes qui seront précisées dans le chapitre 7.

6.3. Résultats de l'inférence de graphes de la ligne médiane

Comme on a pu le voir précédemment de nombreux cas d'appariements se présentent lorsqu'on effectue l'inférence, heureusement les traitements de ces différents cas étant relativement semblables les uns avec les autres ils ne s'avèrent pas trop contraignants.

6.3.1. Stratégie globale

Le processus d'inférence de coupes se déroule de manière dichotomique.

Les coupes données sont les coupes extrémales. Celles-ci permettent de calculer les deux graphes de la ligne médiane initiaux. Ces deux graphes de la ligne médiane sont alors mis en correspondance l'un avec l'autre. Les appariements de cette mise en correspondance sont ensuite mis à profit pour permettre d'inférer un graphe de la ligne médiane intermédiaire. Le graphe ainsi inféré l'est afin de représenter la coupe centrale (à mi-distance des coupes données). Ce graphe inféré n'étant pas exactement un graphe de la ligne médiane au sens le plus strict du terme, un processus de régénération de la coupe centrale puis de calcul du graphe de la ligne médiane intermédiaire est nécessaire pour pouvoir ensuite inférer les autres coupes.

Le processus d'inférence est alors mis en œuvre à nouveau pour permettre de calculer les autres coupes. L'une d'elles est inférée à partir du graphe de la ligne médiane "haut" et du graphe de la ligne médiane intermédiaire, l'autre à partir de ce même graphe de la ligne médiane intermédiaire et du graphe de la ligne médiane "bas".

Ce processus peut ensuite être réitéré autant de fois que l'utilisateur le souhaite afin d'obtenir la précision voulue.

Il faut remarquer qu'au niveau de l'inférence le processus est entièrement guidé par les appariements obtenus lors de la phase de mise en correspondance. Aucun choix n'est fait à ce niveau, le processus d'inférence peut même opérer simultanément sur plusieurs sommets étant donné que seuls les appariements de sommets deux à deux sont utilisés.

6.3.2. Résultats

Les résultats présentés dans les figures 6.21 à 6.31 sont des illustrations simples des différents appariements proposés précédemment ; ce sont des résultats de l'implémentation effectuée. Pour ces résultats une représentation tridimensionnelle a été choisie, elle permet de mieux se rendre compte de l'évolution des différentes coupes. L'épaisseur de chacune des coupes a été volontairement augmentée pour la même raison.

Chacun des résultats a été obtenu à partir des deux coupes extrémales, les autres coupes ont été inférées.

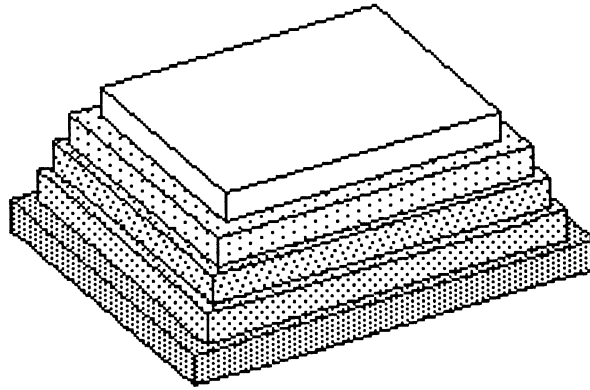


Figure 6.21. Résultat de l'inférence dans le cas de l'appariement sommet principal \leftrightarrow sommet principal.

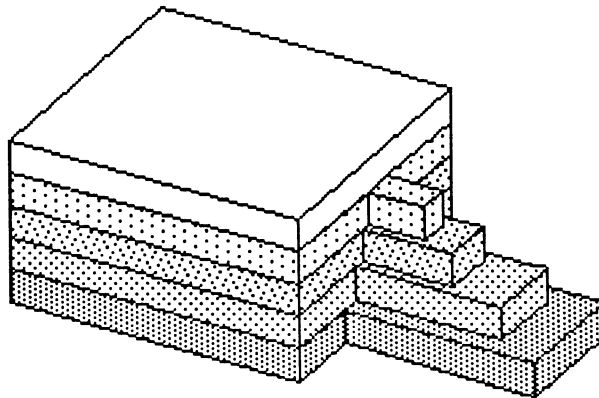


Figure 6.22. Résultat de l'inférence dans le cas de l'appariement sommet principal \leftrightarrow sommet de proéminence stricte.

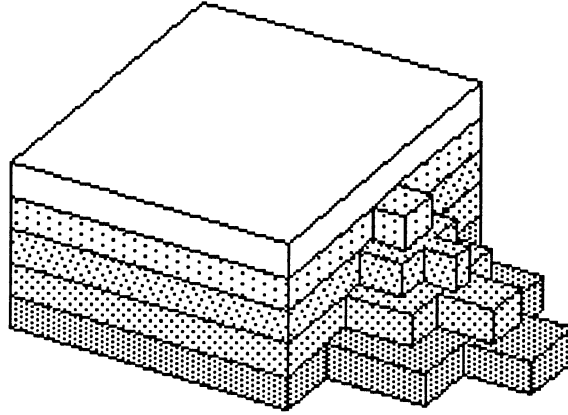


Figure 6.23. Résultat de l'inférence dans le cas de l'appariement sommet principal \leftrightarrow sommet de proéminence large.

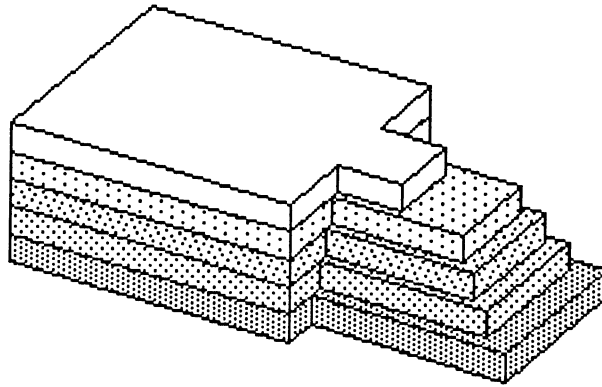


Figure 6.24. Résultat de l'inférence dans le cas de l'appariement sommet de proéminence stricte \leftrightarrow sommet de proéminence stricte.

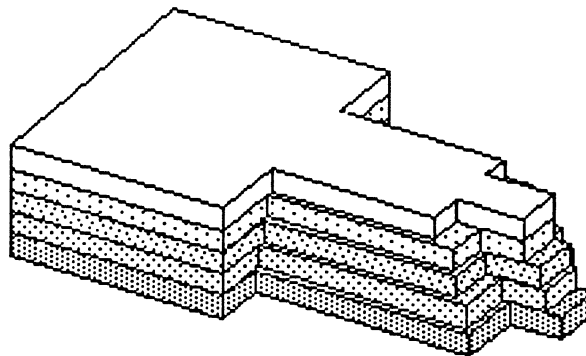


Figure 6.25. Résultat de l'inférence dans le cas de l'appariement sommet de proéminence large \leftrightarrow sommet de proéminence large.

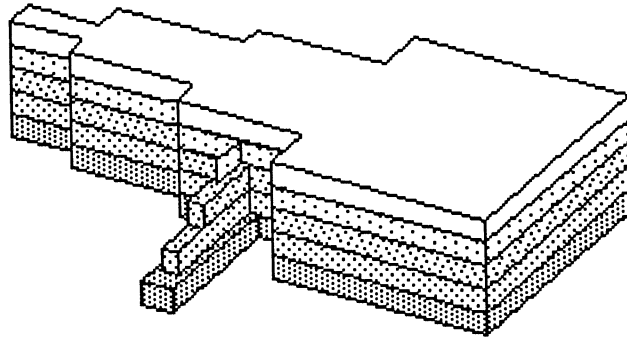


Figure 6.26. Résultat de l'inférence dans le cas de l'appariement sommet de proéminence stricte ↔ sommet de proéminence large.

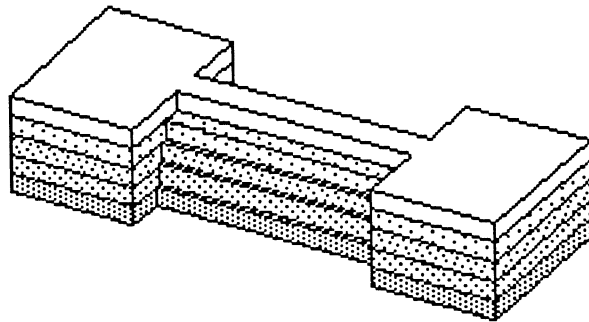


Figure 6.27. Résultat de l'inférence dans le cas de l'appariement sommet de rétrécissement ↔ sommet de rétrécissement.

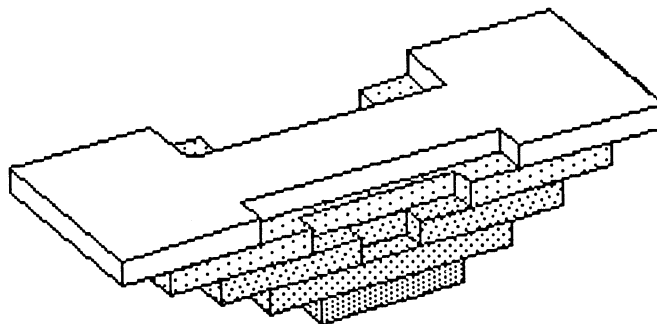


Figure 6.28. Résultat de l'inférence dans le cas de l'appariement sommet de rétrécissement ↔ sommet principal.

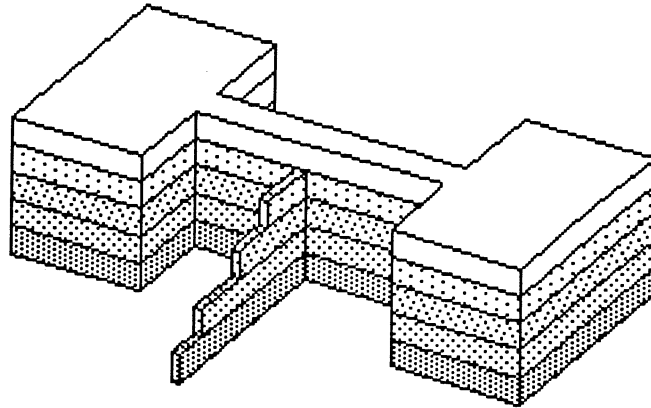


Figure 6.29. Résultat de l'inférence dans le cas de l'appariement sommet de rétrécissement \leftrightarrow sommet de proéminence stricte.

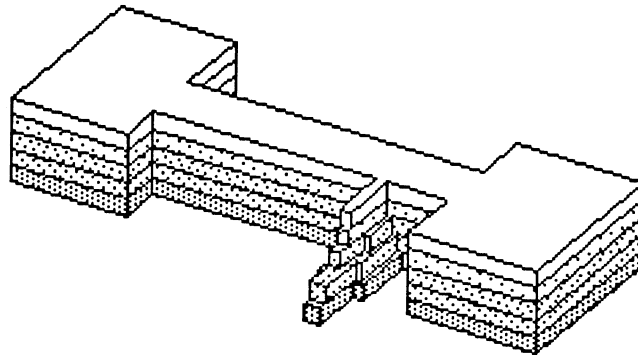


Figure 6.30. Résultat de l'inférence dans le cas de l'appariement sommet de rétrécissement \leftrightarrow sommet de proéminence large.

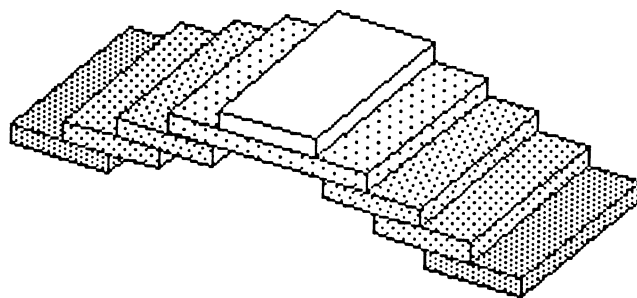


Figure 6.31. Résultat de l'inférence dans le cas de l'appariement sommets principaux \leftrightarrow sommet principal.

Des résultats correspondants à des combinaisons de ces différents cas, et donc plus complexes sont présentés en annexe. Ils correspondent à des cas de figures plus réalistes. Ils permettent de rendre compte de la translation, de l'apparition de déformation, de combinaisons de ces deux cas ainsi que de l'apparition de branchement.

Chapitre 7

Résultats

Les différentes techniques et méthodes présentées dans les chapitres précédents ont été implémentées en langage PASCAL sur des stations de travail APOLLO™ 3550, sous système d'exploitation mixte Unix - Domain Aegis. Certains développements ont nécessité l'utilisation de la bibliothèque graphique spécifique permettant de gérer l'affichage d'objets tridimensionnels (bibliothèque GMR3D).

Les trois principaux points d'intérêt de cette recherche ont été :

- la squelettisation tridimensionnelle
- la mise en correspondance de graphes appliquée aux graphes de la ligne médiane
- l'inférence de formes pour la reconstruction tridimensionnelle à partir de coupes sériées.

Les différents résultats étant présentés en annexe, je vais maintenant présenter ce qui, pour chacun des trois points précédents, est apparu, au cours de ma recherche, comme avantages, inconvénients, limites et perspectives des différentes méthodes employées.

7.1. Squelettisation tridimensionnelle

La méthode de squelettisation tridimensionnelle présentée dans le chapitre 3 permet, à partir d'un objet tridimensionnel, d'extraire une structure filaire (squelette filaire).

La technique proposée procède par érosions successives. Au cours de ces érosions, deux conditions sont maintenues afin de conserver les caractéristiques topologiques de l'objet initial. L'une des conditions garantit la connexité de l'objet, la seconde empêche une érosion trop importantes des extrémités des différentes branches.

Les résultats obtenus permettent de dire que le squelette filaire proposé est un bon descripteur de formes tridimensionnelles. En effet, la signification visuelle du squelette filaire par rapport à la forme initiale est tout à fait établie. De plus, si on ajoute au squelette les pondérations attribuées lors du calcul de l'axe médian, on obtient une quasi réversibilité suffisamment représentative de l'objet.

Comme tous les squelettes, ce squelette filaire est sensible au bruit (cf. figure 2.21 en 2D). Pour pouvoir résoudre ce problème de sensibilité au bruit, deux sortes de méthodes peuvent être mises en œuvre.

La première technique consiste à prendre le problème à la base, c'est-à-dire à réduire autant que possible le bruit à la surface de l'objet. Les techniques utilisables pour ce faire sont des généralisations de celles employées dans le cas de squelettes 2D. Deux méthodes semblent assez facilement généralisables en 3D. La première consiste à procéder à l'aide d'un masque qui, convolué avec le volume, permet de lisser la surface de ce dernier ; des méthodes de morphologie mathématique permettent aussi d'obtenir de tels lissages.

La seconde technique permettant de résoudre le problème de la sensibilité au bruit du squelette consiste, comme on le fait pour les squelettes 2D, à effectuer un ébarbulage du squelette obtenu. Diverses techniques peuvent là encore être utilisées, chacune définissant les points à supprimer. Les définitions des points à éroder telles que "points n'ayant qu'un seul 26-voisin" ou "points dont le 26-voisinage n'est composé que d'une seule composante 6-connexe" peuvent être utilisées. Les configurations tridimensionnelles de points dans le 26-voisinage étant très nombreuses, des conditions strictement tridimensionnelles nécessitent études et développements afin de réduire efficacement la sensibilité au bruit du squelette 3D.

Les deux utilisations directes du squelette filaire sont la détermination d'un parcours de l'objet 3D et la connexion de l'axe médian. Cette connexion de l'axe médian pourrait, comme en 2D, aboutir à la construction d'un graphe de la ligne médiane 3D, graphe qui pourrait être utilisé à des fins de filtrages, décompositions ou autres.

Une des applications du squelette 3D est la quantification de distances à l'intérieur de formes 3D. Une application pour la mesure de distances intra-chromosomiques est d'ailleurs en cours d'étude. En effet, la mesure de la distance entre deux marqueurs fluorescents à l'intérieur d'un chromosome ne peut se faire de manière directe (calcul de la distance en ligne droite) car elle ne

reflète aucune réalité en termes de paires de bases. Il convient en fait de calculer la longueur du chemin interne le plus court et le plus centré dans le chromosome, reliant les deux marqueurs. Le calcul d'un tel chemin, directement sur l'ensemble du chromosome, s'avère assez complexe (utilisation d'algorithmes de type A^*) et long étant donné le grand nombre de voxels composant le chromosome. Par contre, un chemin à l'intérieur du squelette du chromosome est beaucoup plus facile à obtenir. Ainsi, si on détermine le point du squelette le plus proche de chacun des deux marqueurs, il suffit alors de déterminer le chemin le plus court entre ces deux points par parcours du squelette filaire pour pouvoir obtenir une valeur de la distance qui est significative et qui tient compte de la topologie du chromosome.

Dans tous les domaines où il est nécessaire d'appréhender la forme d'un objet tridimensionnel, le squelette peut être utilisé en première approximation de la structure globale de l'objet. Le squelette permet ainsi de modéliser, à moindre frais (en coût de stockage et en temps de calcul) la structure spatiale de l'objet à analyser.

7.2. Mise en correspondance de graphes

La mise en correspondance de graphes est le deuxième point que j'ai plus particulièrement étudié. Cette mise en correspondance de graphes est appliquée aux graphes de la ligne médiane.

La méthode que j'ai utilisée procède en construisant un graphe de mise en correspondance dans lequel on effectue ensuite une recherche de cliques. La construction du graphe de mise en correspondance nécessite la définition des sommets, des arêtes et enfin de la pondération des sommets. La mise en correspondance étant dédiée aux graphes de la ligne médiane, c'est la structure et les différentes caractéristiques de ces graphes qui sont utilisées pour ces définitions. Un filtrage, à des fins de hiérarchisation, des graphes de la ligne médiane est tout d'abord effectué, il permet de ne retenir pour la construction du graphe de mise en correspondance que les sommets du graphe de la ligne médiane vraiment significatifs de la forme codée, et ainsi de réduire d'autant la taille du graphe de mise en correspondance. Lors de ces différentes définitions (sommets, arêtes, pondérations) il est nécessaire de donner des valeurs à certains paramètres. C'est certainement là que se situe un point encore incomplet de la méthode. Comment, en effet, donner des valeurs à ces paramètres pour que la mise en correspondance obtenue soit correcte ? La méthode que j'ai utilisée est empirique, elle est fondée sur l'expérience que j'ai eu de la mise en correspondance de graphes de la ligne médiane.

Un processus d'apprentissage pourrait être mis en œuvre pour la détermination des paramètres susceptibles de favoriser tel ou tel type d'évolution. Ainsi, dans certains domaines

d'applications les translations seront privilégiées, dans d'autres les rotations le seront, dans d'autres encore l'apparition et la disparition de déformations seront favorisées.

Pour la détermination des valeurs des différents paramètres utilisés pour l'évaluation de la pondération attribuée à chaque sommet du graphe de mise en correspondance, là encore un apprentissage semble possible. Les différents paramètres étant assez dépendants les uns avec les autres lors de la combinaison linéaire de ceux-ci, la distinction des différentes valeurs devrait malgré tout être assez difficile à obtenir. De même, un apprentissage permettant à partir d'un graphe d'obtenir ensuite directement la mise en correspondance est assez utopique étant donné le nombre de possibilités offertes par la mise en correspondance.

La deuxième phase de la mise en correspondance est constituée par la recherche de cliques. Cette recherche consiste à trouver le sous-graphe complet à un nombre de sommets donné qui minimise une fonction coût.

Cette recherche a tout d'abord été abordée à l'aide de méthodes de recherches exhaustives. Ces techniques étant déraisonnables dès que les formes à mettre en correspondance deviennent moins géométriques, elles ont été abandonnées au profit de méthodes heuristiques. Les méthodes heuristiques développées sont fondées sur le recuit simulé et sur un algorithme génétique. Elles permettent d'obtenir de très bons résultats en des temps très courts, par rapport à la complexité du problème posé. Les seuls reproches possibles à l'encontre de ces méthodes sont leur non reproductibilité et le fait qu'il faille choisir des paramètres. Malheureusement ces défauts ne peuvent être gommés, ils sont certainement la contrepartie de résultats aussi impressionnants. Les résultats de la mise en correspondance obtenus sont très satisfaisants, ils ont des significations très pertinentes au niveau des formes.

La recherche de cliques proposée peut trouver d'autres applications que la mise en correspondance de graphes de la ligne médiane. L'une d'elles est la décomposition d'objets en éléments convexes.

Une décomposition en éléments convexes peut être obtenue en trouvant les cliques dans un graphe de "visibilité" construit à partir des sommets du polygone à décomposer [Shapiro 79] [Shapiro 80]. En effet, tout élément convexe correspond à une clique dans le graphe de visibilité (tout point de l'élément convexe est visible depuis tous les autres points de cet élément convexe), donc la recherche de clique permet de trouver une décomposition de l'objet en éléments convexes (cf. figure 7.1). La méthode de recherche de cliques permettant de trouver aussi des graphes qui ne sont pas des cliques mais sont presque complets (il manque une ou deux arêtes pour que le graphe soit complet), elle s'adapte donc très bien à ce problème de décomposition en éléments convexes.

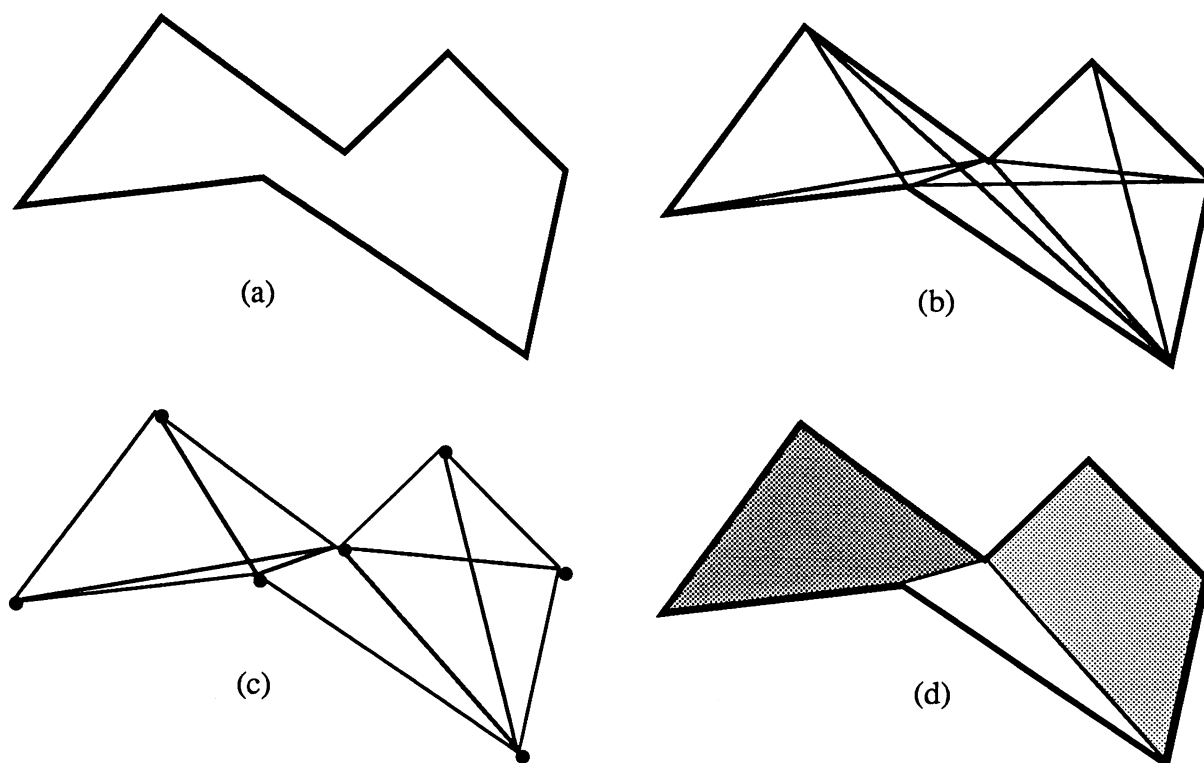


Figure 7.1. Décomposition en éléments convexes.
 (a) un polygone, (b) le graphe de visibilité, (c) les cliques extraites et
 (d) la décomposition en éléments convexes.

7.3. Inférence de formes et reconstruction tridimensionnelle

Ce troisième point est l'aboutissement des développements précédents.

Deux méthodes à ce sujet sont en concurrence, l'une s'appuie sur une représentation par contours, l'autre sur une représentation par régions. Il s'agit de la méthode par assortiment élastique de contours et de celle par inférence de graphes de la ligne médiane.

La méthode que j'ai développée consiste à produire des coupes intermédiaires, entre deux coupes données, à l'aide d'un processus d'inférence mis en œuvre à l'aide de graphes de la ligne médiane.

Cette inférence de graphes de la ligne médiane est guidée par les appariements obtenus lors de la mise en correspondance des graphes de la ligne médiane donnés.

Cette inférence consiste, à partir de deux graphes de la ligne médiane initiaux et de l'appariement de leurs sommets, à calculer un graphe de la ligne médiane représentant, autant que possible, une forme intermédiaire entre les deux formes données. Les sommets des graphes initiaux étant appariés, le processus d'inférence de graphes se réduit à un processus d'inférence de sommets. Le procédé d'inférence actuellement implanté est de type dichotomique : un seul

graphe de la ligne médiane est inféré ; si on souhaite inférer plus de graphes il faut calculer le graphe intermédiaire puis relancer le processus entre le graphe intermédiaire et le graphe de la coupe “haut” d’une part, et le graphe de la coupe “bas” d’autre part. Actuellement, le graphe intermédiaire inféré correspond à une coupe placée à mi-distance entre les deux coupes initiales. Les sommets des graphes de la ligne médiane étant appariés et le processus de déformation défini qualitativement, il serait naturellement possible d’inférer un graphe représentant une coupe se plaçant ailleurs entre les coupes données. De même, il serait aussi possible d’inférer directement plusieurs graphes de la ligne médiane.

L’évolution des déformations entre les deux coupes données étant faite, le processus d’inférence permet de faire une interpolation de formes mais aussi, dans certain cas, une extrapolation (inférence de coupes en dessus de la coupe “haut” ou en dessous de la coupe “bas”).

La méthode d’inférence de formes, et par conséquent celle de reconstruction tridimensionnelle, proposées donnent de bons résultats.

Certaines limites sont cependant à noter.

Tout d’abord, il faut remarquer les problèmes rencontrés avec les rotations. Si les deux formes données ne se différencient l’une de l’autre que par une rotation, il est souvent bien difficile de faire une reconstruction correcte. En effet, le graphe de la ligne médiane étant calculé à l’aide de distances carrées (d_g), il est très sensible aux rotations. Donc, même si les deux formes sont presque identiques leur graphe de la ligne médiane peut être très différent. Ainsi, malgré les filtrages opérés sur les graphes, la mise en correspondance ne donne pas toujours de bons résultats. Ainsi, le résultat de l’inférence, et finalement celui de la reconstruction ne sont pas très bons. Ce problème devrait pouvoir être assez facilement résolu par l’introduction d’un graphe de la ligne médiane utilisant des distances plus isotropes (distance du chanfrein par exemple), si l’utilisation de telles distances ne pose pas trop de problèmes par ailleurs [Thiel 91b]. On peut remarquer par contre que la méthode utilisée s’affranchit très bien de tous les cas de translations et de changement de taille.

Le second problème rencontré se situe au niveau de l’inférence de sommets dans le cas où les sommets du graphe de la ligne médiane codent des ensembles assez allongés de points de l’axe médian. Il est vrai qu’il est difficile de savoir quel genre de sommet inférer entre un sommet représentant un ensemble horizontal de points de l’axe médian et un sommet représentant un ensemble vertical de points. En effet, chaque sommet du graphe de la ligne médiane code un ensemble de points de l’axe médian ; cet ensemble de points peut avoir une forme tout à fait différente pour deux sommets de graphes de la ligne médiane appariés l’un avec l’autre. Ainsi, lors de l’inférence de sommets, les sommets des graphes de la ligne médiane étant traités

comme s'ils étaient ponctuels, les résultats obtenus sont quelquefois décevants. Ce genre de cas correspond en fait à des cas extrêmes. Pour résoudre ce problème, il faudrait en plus des différents traitements opérés lors de l'inférence, effectuer une "inférence d'ensemble de points". Les ensembles de sommets étant en fait des portions de squelettes (au maximum d'épaisseur 2), il s'agit en fait de faire une inférence d'ensemble de segments. Cette inférence pourrait être abordée à l'aide de méthodes basées sur les points d'extrémité, de jonction, etc ou pourquoi pas à l'aide de modèles de déformation (méthode de D.J. Burr [Burr 81], par exemple).

Tous les autres problèmes rencontrés au cours de l'inférence sont principalement liés à l'utilisation de l'espace discret. En effet, lorsque l'on doit inférer un graphe intermédiaire, il est parfois nécessaire de faire des choix au niveau des tailles des boules et de leur position. Par exemple, entre une boule de taille 2 et une boule de taille 5 que doit-on inférer ? Une boule de taille 3 ou une boule de taille 4 ? Dans d'autres cas, l'espace discret empêche même d'inférer une structure différente de celles données. Par exemple, chaque fois qu'il faut inférer une boule intermédiaire entre une boule de taille n et une boule de taille $n+1$. Dans ce cas il n'y a pas d'intermédiaire.

L'utilisation de l'espace discret est certainement ce qui limitera le plus l'avancement de la méthode développée ici au niveau des applications directes. Ce problème d'espace discret peut malgré tout être réduit si on travaille avec des objets suffisamment grands par rapport au maillage choisi. Dans ce cas, une erreur de 1 pixel (ou de 1 voxel) ne serait pas significative donc ne poserait plus de problème.

L'utilisation d'images beaucoup plus grandes est donc certainement la solution à ce problème. Malheureusement, le traitement des images 512 x 512 (ou encore mieux 1024 x 1024) pose des problèmes de temps de calcul tant au niveau des prétraitements effectués (segmentation) que du codage (calcul du graphe de la ligne médiane, affichage, ...) et de la taille des données manipulées. De plus si on souhaite rester cohérent, les volumes utilisés doivent être eux aussi de plus grande taille (512 x 512 x 512 par exemple) et là les problèmes de temps de calcul et de volumes des données deviennent impossibles à résoudre à l'aide des machines actuellement à notre disposition.

Au niveau des résultats obtenus pour la reconstruction tridimensionnelle à partir de coupes sériées ceux-ci sont encourageants. Malheureusement, de par la méthode utilisée ils présentent souvent un aspect "marche d'escalier". Ce problème-là devrait être résolu soit l'aide d'une phase de lissage soit à l'aide d'une modélisation de la surface de l'objet par des modèles continus (splines par exemple).

7.4. Conclusions

Compte tenu des différentes remarques faites précédemment les résultats montrent la validité des méthodes développées tant dans le domaine de la squelettisation d'objets tridimensionnels que dans le domaine de la reconstruction tridimensionnelle à partir de coupes sériées.

Cependant certains points nécessitent encore des développements :

- dans certains cas, le squelette tridimensionnel filaire possède un grand nombre de barbules. En application réelle des méthodes de lissage de l'objet initial et d'ébarbulage du squelette pourraient être développées afin de résoudre définitivement ce problème.

- il serait intéressant de développer un graphe de la ligne médiane tridimensionnel, ceci afin de savoir s'il est un aussi bon outil de description, filtrage et décomposition que le graphe de la ligne médiane l'est en 2D.

- le graphe de la ligne médiane utilise la distance de l'échiquier qui est non isotrope, cela pose de nombreux problèmes. Un graphe de la ligne médiane utilisant la distance du chanfrein 3-4 est actuellement à l'étude.

- au niveau de l'inférence de graphes de la ligne médiane, une inférence d'un nombre plus important de graphes devrait pouvoir être mise en place assez facilement si, pour autant, l'espace discret ne pose pas plus de problèmes.

- au niveau de la visualisation tridimensionnelle du résultat final de la reconstruction il serait intéressant de mettre en œuvre diverses techniques de la synthèse d'images afin d'obtenir un meilleur rendu visuel. Celles-ci permettraient d'obtenir des résultats plus lissés et plus agréables à voir (ombrage par exemple). Elles devront être implantés avec le souci de ne pas trop augmenter les temps de calculs en utilisant par exemple des bibliothèques et des coprocesseurs spécifiques dédiés à la visualisation tridimensionnelle.

- enfin la méthode de reconstruction va être validée dans des cas réels plus nombreux, en trouvant ses applications en microscopie confocale (reconstruction de chromosomes, ...) ou en macroscopie (reconstruction de vertèbres, modélisation de prothèses de hanches, ...).

Conclusion

Dans cette thèse, nous nous sommes intéressés à la reconstruction tridimensionnelle à partir de coupes sériées bidimensionnelles.

Après avoir étudié les différentes méthodes de reconstruction et après avoir constaté leurs échecs relatifs, nous nous sommes attachés à développer une nouvelle méthode de reconstruction. Pour ce faire nous avons été amenés à nous pencher sur les différents modèles utilisés en analyse de formes tant dans leur aspect bidimensionnel que dans leur aspect tridimensionnel. Cette étude nous a permis de comparer les différents modes de représentation, de description et de codage. L'étude de méthodes de squelettisation tridimensionnelle nous a permis de développer un algorithme de calcul de squelette filaire. Les différents modèles de représentation bidimensionnels ayant été étudiés, c'est le graphe de la ligne médiane qui a été retenu pour l'élaboration de la nouvelle méthode de reconstruction. La méthode développée s'appuie alors sur un processus d'inférence de formes effectué à l'aide de graphes de la ligne médiane. L'inférence de graphes implique la mise en œuvre d'une phase de mise en correspondance pour laquelle diverses techniques ont été étudiées. Finalement, c'est la méthode procédant par recherche de cliques dans un graphe de mise en correspondance qui a été choisie. Différentes méthodes de recherche de cliques ont, là encore, été comparées, les méthodes heuristiques (recuit simulé et algorithmes génétiques) se sont avérées les plus efficaces. Enfin, après avoir effectué l'appariement des sommets des deux graphes de la ligne médiane, la phase d'inférence proprement dite a pu être mise en place. Celle-ci a permis de procéder à l'inférence

de formes intermédiaires qui, empilées avec les coupes données permettent d'obtenir la forme tridimensionnelle.

Les études menées au cours de ces différentes étapes nous ont permis de constater les avantages et les inconvénients des méthodes utilisées. Elles nous permettent ainsi de percevoir les limites des techniques mises en œuvre, d'en prévoir les améliorations et les développements futurs, mais aussi d'entrevoir de nouvelles applications .

Références bibliographiques

- [Ahuja 84] N. Ahuja ; W. Hoff :
Augmented medial axis transform ;
7th International Conference on Pattern Recognition. Montréal. 1984.
pp. 336-338.
- [Arcelli 72] C. Arcelli ; S. Levialdi :
Parallel shrinking in three dimensions ;
Computer Graphics & Image Processing, n° 1. 1972. pp 21-30.
- [Arcelli 81] C. Arcelli :
Pattern thinning by contour tracing ;
Computer Graphics & Image Processing, n° 17. 1981. pp 130-144.
- [Arcelli 85] C. Arcelli ; G. Sanniti di Baja :
A with-independent fast thinning algorithm ;
IEEE Transactions on Pattern Analysis and Machine Intelligence,
PAMI-7, n° 4. Juillet 1985. pp 463-474.
- [Arcelli 88] C. Arcelli ; G. Sanniti di Baja :
Finding local maxima in pseudo-euclidean distance transform ;
Computer Vision, Graphics & Image Processing, n° 43. 1988.
pp 361-367.
- [Artzy 81] E. Artzy ; G. Frieder ; G.T. Herman :
The theory, design, implementation and Evaluation of a three
dimensional surface detection algorithm ;
Comp. Graph. & Im. Proc. vol. 15. 1981. pp 1-24.

- [Ayache 88] N. Ayache :
Construction et fusion de représentations visuelles 3D - applications à la robotique mobile - ;
Thèse Université de Paris Sud, centre d'Orsay. Mai 1988.
- [Ayache 89] N. Ayache :
Vision stéréoscopique et perception multisensorielle ;
InterEditions. 1989.
- [Ballard 82] D.H. Ballard ; C.M. Brown :
Computer vision ;
Prentice-Hall. 1982.
- [Barillot 84] C. Barillot :
Méthodes de reconstruction tri-dimensionnelle et de synthèse d'images appliquées à la représentation des structures morphologiques en médecine ;
Thèse Université de Rennes I. Décembre 1984.
- [Barrow 76] H.G. Barrow ; R.M. Burstall :
Subgraph isomorphism, matching relational structures and maximal cliques ;
Information Processing Letters, vol.4, n° 4. Janvier 1976. pp 83-84.
- [Bartels 87] R.H. Bartels ; J.C. Beatty ; B.A. Barsky :
B-splines ;
Hermès. 1987.
- [Baujard 89] O. Baujard ; C. Garbay :
KISS : un système de vision multi-agents ;
7ème Congrès R.F.I.A. Paris. 1989. pp 89-98.
- [Bertrand 84] G. Bertrand :
Détermination de l'axe médian par sur-échantillonnage fictif ;
4ème Congrès R.F.I.A. Paris. Janvier 1984. pp 265-275.
- [Blum 64] H. Blum :
A transformation for extracting new descriptors of shape ;
Symp. on models for perception of speech and visual form. Boston.
Novembre 1964. MIT Press. pp 362-380.
- [Boissonnat 84] J.D. Boissonnat :
Geometric structures for three-dimensional shape representation ;
ACM Transactions on Graphics, vol. 3, n° 4. Octobre 1984.
pp 266-286.
- [Boissonnat 85] J.D. Boissonnat :
Surface reconstruction from planar cross-sections ;
IEEE Conference on Computer Vision and Pattern Recognition.
San Francisco. June 1985. pp 393-397.
- [Boissonnat 88] J.D. Boissonnat :
Shape reconstruction from planar cross-sections ;
Computer Vision, Graphics & Image Processing, 44. 1988. pp 1-29.

- [Bolles 79] R.C. Bolles :
Robust feature matching through maximal cliques ;
In Proc. SPIE Technical Symposium on Imaging Applied to Automated Industrial Inspection and Assembly, Washington DC, Avril 1979.
- [Bondy 76] J.A. Bondy ; U.S.R. Murty :
Graph theory with applications ;
North Holland. The Macmillan Press Ltd. 1976.
- [Borgefors 84] G. Borgefors :
Distance transformations in arbitrary dimensions ;
Computer Vision, Graphics & Image Processing, n° 27. 1984.
pp 321-345.
- [Borgefors 86] G. Borgefors :
Distance transformations in digital images ;
Computer Vision, Graphics & Image Processing, n° 34. 1986.
pp 344-371.
- [Borgefors 88] G. Borgefors ; G. Sanniti di Baja :
Skeletonizing the distance transform on hexagonal grid ;
9th International Conference on Pattern Recognition. Rome. Novembre 1988. pp 504-507.
- [Bron 73] C. Bron ; J. Kerbosch :
Algorithm 457 : finding all cliques of an undirected graph ;
Communications of ACM, vol. 16, n° 9. Septembre 1973. pp 575-577.
- [Burr 81] D.J. Burr :
Elastic matching of line drawings ;
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-3, n° 6. Novembre 1981. pp 708-713.
- [Calverley 88] R.K.S. Calverley ; Y. Usson :
A computer-assisted three-dimensional reconstruction package : its application to the study of synaptic plasticity ;
Internal Report n° 3 / RC 3, Octobre 1988, Neuroscience Centre, University of Otago, Dunedin, Nouvelle Zélande.
- [Carlbom 85] I. Carlbom ; I. Chakravarty ; D. Vanderschel :
A hierarchical data structure for representing the spacial decomposition of 3-D objects ;
IEEE Computer Graphics & Applications. Avril 1985. pp 24-31.
- [Chassery 91] J.M. Chassery ; A. Montanvert :
Géométrie discrète en analyse d'images ;
Hermès. 1991.
- [Chen H.H. 88] H.H. Chen ; T.S. Huang :
A survey of construction and manipulation of octrees ;
Computer Vision, Graphics & Image Processing, n° 43. 1988.
pp 409-431.

- [Chen J.S. 88] J.S. Chen ; W.C. Lin :
A new surface interpolation technique for reconstructing 3D objects from serial cross-sections ;
9th International Conference on Pattern Recognition. Rome. Novembre 1988. pp 1100-1102.
- [Chianese 89] A. Chianese ; M. De Santo ; M. Vento :
A heuristic matcher of ARG graphs for object recognition ;
5th International Conference on Image Analysis and Processing, Positano, Italy, Septembre 1989, pp 301-308.
- [Christiansen 78] H.N. Christiansen ; T.W. Sedberg :
Conversions of complex contour line definitions into polygonal element mosaics ;
Computer Graphics ACM. 1978. pp 187-192.
- [Christofides 75] N. Christofides :
Graph theory : an algorithmic approach ;
Academic Press. 1975.
- [Cook 80] P. Cook :
Three-dimensional reconstruction from serial sections for medical applications ;
Ph. D. University of Missouri-Columbia. 1980.
- [Cordella 89] L.P. Cordella ; G. Sanniti Di Baja :
Geometric properties of the union of maximal neighborhoods ;
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-11, n° 2. Février 1989. pp 214-217.
- [Das 88] P.P. Das :
A note on "Distance transformations in arbitrary dimensions" ;
Computer Vision, Graphics & Image Processing, n° 43. 1988. pp 368-385.
- [Dickinson 91] S.J. Dickinson ; A. Rosenfeld ; A.P. Pentland :
Primitive-based shape modeling and recognition ;
International Workshop on Visual Form, Plenum Press, Capri, Italie, Mai 1991.
- [Dorst 86] L. Dorst :
Pseudo-euclidean skeletons ;
8th International Conference on Pattern Recognition, Paris, 1986, pp 286-288.
- [Duda 73] R.O. Duda ; P.E. Hart :
Pattern classification and scene analysis ;
Wiley-Interscience, New-York, 1973.
- [Dyer 80] C.R. Dyer ; A. Rosenfeld ; H. Samet :
Region representation : boundary codes from quadrees ;
Communications of ACM, vol. 23, n° 3. Mars 1980. pp 171-179.

- [Eshera 84] M.A. Eshera ; K.S. Fu :
A similarity measure between attributed relational graphs for image analysis ;
7th International Conference on Pattern Recognition, Montréal, 1984, pp 75-77.
- [Freeman 61] H. Freeman :
On the encoding of arbitrary geometric configurations ;
IRE Trans. electron. Comput. vol. EC 10. Juin 1961. pp.260-268.
- [Fuchs 77] H. Fuchs ; Z.M. Kedem ; S.P. Uselton :
Optimal surface reconstruction from planar contours ;
Communications of ACM, vol.20, n° 10. Octobre 1977. pp 693-702.
- [Gagalowicz 86] A. Gagalowicz ; S.D. Ma :
Model driven synthesis of natural textures for 3D scenes ;
Computer & Graphics, vol. 10, n° 2. 1986.
- [Galil 86] Z. Galil :
Efficient algorithms for finding maximum matching in graphs ;
ACM Computing Surveys, vol. 18, n° 1. Mars 1986. pp 23-38.
- [Ganapathy 82] S. Ganapathy ; T.G Dennehy :
A new general triangulation method for planar contours ;
Computer Graphics ACM. 1982. pp 69-75.
- [Gong 90a] W.X. Gong ; G. Bertrand :
A simple parallel 3D Thinning algorithm ;
10th International Conference on Pattern Recognition, Atlantic City, Juin 1990, pp 188-190.
- [Gong 90b] W.X. Gong ; G. Bertrand :
A note on "Thinning of 3D images using the safe point thinning algorithm (SPTA)" ;
Pattern Recognition Letters, 11, 1990, pp 499-500.
- [Gordon 74] R. Gordon ; G.T. Herman :
Three-Dimensional reconstruction from projections : a review of algorithms ;
Intern. Rev. of Cytology, 38. 1974. pp 111-151.
- [Grimson 83] W.E.L. Grimson :
An implementation of a computational theory of visual surface interpolation ;
Computer Vision, Graphics & Image Processing, n° 22. 1983. pp 39-69.
- [Hafford 84] K.J. Hafford ; K. Preston Jr :
Three-dimensional skeletonization of elongated solids ;
Computer Vision, Graphics & Image Processing, n° 27. 1984. pp 77-91.

- [Haralick 85] R.M. Haralick ; L.G. Shapiro :
Image segmentation techniques ;
Computer Vision, Graphics & Image Processing, 29. 1985.
pp 100-132.
- [Hérault 90] L. Hérault ; R. Horaud ; F. Veillon ; J.J. Niez :
Symbolic image matching by simulated annealing ;
British Machine Vision Conference. Oxford. Septembre 1990.
- [Hilbert 52] D. Hilbert ; S. Cohn-Vossen :
Geometry and the imagination ;
Chelsea, New-York. 1952.
- [Hofstadter 85] D. Hofstadter :
Gödel, Escher, Bach ou les brins d'une guirlande éternelle ;
InterEditions, 1985.
- [Holland 75] J.H. Holland :
Adaptation in natural and artificial systems ;
Ann. Arbor. University of Michigan Press, 1975.
- [Horaud 89] R. Horaud ; T. Skordas :
Stereo correspondence through feature grouping and maximal cliques ;
IEEE Transactions on Pattern Analysis and Machine Intelligence,
PAMI-11, n° 11. Novembre 1989. pp 1168-1180.
- [Jackins 80] C.L. Jackins ; S.L. Tanimoto :
Octrees and their use in representing three-dimensional objets ;
Computer Graphics & Image Processing, vol. 14. 1980. pp 249-270.
- [Kaneda 88] K. Kaneda ; Y. Wakazu ; E. Nakamae ; M. Yasuda ; A.G. Sato :
Accurate reconstruction of 3D object composed of multiple surfaces ;
1^{ère} Conférence Annuelle sur l'Image Numérique à Paris. PIXIM 88.
Octobre 1988. pp 15-29.
- [Keppel 75] E. Keppel :
Approximating complex surfaces by triangulation of contour lines ;
IBM Journal of Res. Develop. Janvier 1975. pp 2-11.
- [Kirkpatrick 83] S. Kirkpatrick ; C.D. Gelatt Jr. ; M.P. Vecchi :
Optimization by simulated annealing ;
Science, vol. 220, n° 4598. 13 Mai 1983, pp 671-680.
- [Kirkpatrick 84] S. Kirkpatrick :
Optimization by simulated annealing : quantitative studies ;
Journal of Statistical Physics, vol. 34, n° 516. 1984, pp 975-987.
- [Kunii 85] T.L. Kunii ; T. Satoh ; K. Yamaguchi :
Generation of topological boundary representations from octree
encoding ;
IEEE Computer Graphics & Applications. Mars 1985. pp 29-38.
- [Laarhoven 87] P.J.M. van Laarhoven ; E.H.L. Aarts :
Simulated annealing : theory and applications ;
D. Reidel Publishing Company. 1987.

- [Laurini 88] R. Laurini :
Panorama de la modélisation géométrique basée sur les quadrees et octrees ;
Séminaire Modèles Géométriques et Images Numériques. Lyon. Avril 1988.
- [Levine 91] M.D. Levine ; R. Bergevin ; Q.L. Nguyen:
Shape description using geons as 3D primitives ;
International Workshop on Visual Form, Plenum Press, Capri, Italie, Mai 1991.
- [Leymarie 89] F. Leymarie ; M.D. Levine :
Snakes and skeletons ;
Computer Vision and Robotics Laboratory. Mac Gill Research Centre of Intelligent Machines. Mac Gill University. Montréal, Quebec, Canada. Janvier 1989.
- [Lin 89] W.C. Lin ; S.Y. Chen ; C.T. Chen :
A new surface interpolation technique for reconstructing 3D objects from serial cross-sections ;
Computer Vision, Graphics, and Image Processing, 48. 1989. pp 124-143.
- [Liu 77] H.K. Liu :
Two and three-dimensional boundary detection ;
Computer, Graphics & Image Processing, vol. 6. 1977. pp 123-134.
- [Lobregt 80] D. Lobregt ; P.W. Verbeek ; F.C. Groen :
Three-Dimensional skeletonization ; principle and algorithm ;
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-2, n° 1. Janvier 1980. pp 75-77.
- [Lux 85] A. Lux :
Algorithmique et contrôle en vision par ordinateur ;
Thèse d'état. USTMG et INPG. Grenoble. Septembre 1985.
- [Macias-Garza 89] F. Macias-Garza ; K.R. Diller ; A.C. Bovik ; S.J. Aggarwal ; J.K. Aggarwal :
Obtaining a solid model from optical serial sections ;
Pattern Recognition, vol. 22, n° 5. 1989. pp 577-586.
- [Marque 90] I. Marque :
Segmentation d'images médicales tridimensionnelles basée sur une modélisation continue du volume ;
Thèse Université Joseph Fourier GRENOBLE I. 1990.
- [Martinez 82] F. Martinez :
Vers une approche systématique de la synthèse d'images : aspects logiciel et matériel ;
Thèse de l'Institut National Polytechnique de Grenoble. 1982.
- [Martinez 86] F. Martinez :
La synthèse d'images, concepts, logiciels et matériels ;
Editions Editest. 1986.

- [Matsuyama 88] T. Matsuyama :
Expert systems for image processing - knowledge-based composition of image analysis processes ;
9th International Conference on Pattern Recognition. Rome. Novembre 1988. pp 1100-1102.
- [Mémier 91] M. Mémier :
Stéréophotogrammétrie numérique : calcul de M.N.T. par corrélation automatique d'images SPOT ;
Thèse Université Joseph Fourier GRENOBLE I. 1991.
- [Miclet 84] L. Miclet :
Méthodes structurelles pour la reconnaissance des formes ;
Eyrolles. 1984.
- [Monga 90] O. Monga ; R. Deriche ; G. Malandain ; J.P. Cocquerez :
3D edge detection by separable filtering and edge closing ;
10th International Conference on Pattern Recognition, Atlantic City, Juin 1990, pp 652-654.
- [Montanari 68] U. Montanari :
A method for obtaining skeletons using a quasi-euclidean distance ;
ACM Journal, vol.15, n° 4. 1988. pp 213-247.
- [Montanvert 85] A. Montanvert :
Obtention d'une ligne médiane par connexion de l'axe médian ;
5ème Congrès R.F.I.A. Grenoble. 1985. pp 777-785.
- [Montanvert 87a] A. Montanvert :
Graph environment from medial axis for shape manipulation ;
4th International Conference on IAPR. Cefalu (Italie). Septembre 1987.
pp 197-203
- [Montanvert 87b] A. Montanvert :
Contribution au traitement de formes discrètes - squelettes et codage par graphe de la ligne médiane ;
Thèse de l'USTMG et de l'INPG. Grenoble. Octobre 1987.
- [Montanvert 87c] A. Montanvert ; J.M. Chassery ; C. Charles :
Méthodes de décomposition d'Images binaires appliquées à la trajectologie sous-marine ;
"Traitement du Signal". Spécial Vision par Ordinateur. 1987.
pp 195-203.
- [Montanvert 87d] A. Montanvert, J.M. Chassery :
Architecture d'images binaires discrètes : outils géométriques ;
"Traitement du Signal". Spécial Vision par Ordinateur. 1987.
pp 205-216.
- [Montanvert 88] A. Montanvert :
Une interface orientée objet pour la manipulation de formes codées au moyen de régions ;
1ère Conférence Annuelle sur l'Image Numérique à Paris.PIXIM 88.
Octobre 1988. pp 331-334.

- [Montanvert 89] A. Montanvert ; D. Adelh :
Tools for shape complexity evaluation and associated decomposition processes ;
5th Int. Conf. on Image Analysis and Processing, Positano (Italie), 1989. pp 348-355.
- [Mukherjee 89] J. Mukherjee ; B.N. Chatterji ; P.P. Das :
Thinning of 3-D images using the safe point thinning algorithm (SPTA) ;
Pattern Recognition Letters, 10. 1989. pp 167-173.
- [Mukherjee 90] J. Mukherjee ; P.P. Das ; B.N. Chatterji :
On connectivity issues of ESPTA ;
Pattern Recognition Letters, 11. 1990. pp 643-648.
- [Nevatia 82] R. Nevatia :
Machine perception ;
Prentice Hall. 1982.
- [Nguyen 87] N.Q. Nguyen ; S. Hazout :
Segmentation and thinning by morphological automata - applications in cytogenetics - ;
Acta Stereologica, 6 / III. 1987. pp 675-680.
- [Okabe 83] N. Okabe ; J.i. Toriwaki ; T. Fukumura :
Paths and distance functions on three-dimensional digitalized pictures ;
Computer Vision & Pattern Recognition, Juin 83. Washington DC.
pp 384-386.
- [Ordre & Chaos] L'ordre et le chaos ;
Recueil d'articles de divers auteurs ;
Collection "Pour la Science".
- [O'Rourke 79] J. O'Rourke, N. Badler :
Decomposition of three-dimensional objets into spheres ;
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1, n° 3. Juillet 1979. pp 295-305.
- [Paugam-Moisy 90] H. Paugam-Moisy :
A spy of parallel neural networks ;
Rapport de recherche n° 90-27. LIP - Ecole Normale Supérieure de Lyon. Novembre 1990.
- [Pavlidis 74] T. Pavlidis ; S.L. Horowitz :
A thinning algorithm for discrete binary images ;
Computer, Graphics & Image Processing, n°13. 1980. pp 142-157.
- [Pavlidis 80] T. Pavlidis :
Segmentation of plane curves ;
IEEE Trans. on Comp., vol. C23, n° 8. 1974. pp 860-870.
- [Péroche 88] B. Péroche ; J. Argence ; D. Ghazanfarpour ; D. Michelucci :
La synthèse d'images ;
Hermès. 1988.

- [Pfaltz 67] J.L. Pfaltz, A. Rosenfeld :
Computer representation of planar regions by their skeletons ;
Communications of ACM, vol. 10, n° 2. Février 1967. pp 119-125.
- [Pieroni 88] G.G. Pieroni ; Y.P. Wing :
Reconstructing the activity surface for cardiac PET imagery ;
9th International Conference on Pattern Recognition. Rome. Novembre
1988. pp 125-133.
- [Pizer 87] S.M. Pizer ; R.O. William ; S.H. Bloomberg :
Hierarchical shape description via the multiresolution symmetric axis
transform ;
IEEE Transactions on Pattern Analysis and Machine Intelligence,
PAMI-9, n° 4. Juillet 1987. pp 505-511.
- [Preparata 85] F.P. Preparata ; M.I. Shamos :
Computational geometry : an introduction ;
Springer-Verlag. 1985.
- [Rao 90] A.R. Rao :
Representations for rigid solids : theory, methods and systems ;
Computing Surveys, vol. 12, n° 4. Décembre 1980. pp 437-464.
- [Requicha 80] A.A.G. Requicha :
A taxonomy for texture description and identification ;
Springer-Verlag, 1990.
- [Rogers 85] D.F. Rogers :
Procedural elements for computer graphics ;
Mac Graw Hill. 1985.
- [Rolland 89a] F. Rolland :
Interpolation de volumes à partir de coupes sériées ;
Rapport de D.E.A. d'Informatique. INPG (ENSIMAG). Grenoble.
Juin 1989.
- [Rolland 89b] F. Rolland ; A. Montanvert ; J.M. Chassery :
Reconstruction de volumes à partir de coupes sériées ;
2^{ème} Conférence Annuelle sur l'Image Numérique à Paris. PIXIM 89.
Septembre 1989. pp 511-525.
- [Rolland 90] F. Rolland :
Mise en correspondance de graphes pour l'inférence de formes codées
par leur squelette ;
Rapport de Recherche 820 -I- TIM3 - Institut IMAG - Grenoble -
Juin 1990.
- [Rolland 91a] F. Rolland ; J.M. Chassery ; A. Montanvert :
3D medial surfaces and 3D skeletons ;
International Workshop on Visual Form. Capri. Italie. 26-29 Mai
1991. à paraître.

- [Rolland 91b] F. Rolland ; A. Montanvert ; J.M. Chassery :
Graph matching for 3D reconstruction from serial cross-sections using simulated annealing ;
7th Scandinavian Conference on Image Analysis. Aalborg. Danemark. 13-16 Août 1991. à paraître.
- [Rosenfeld 68] A. Rosenfeld ; J.L. Pfaltz :
Distance functions on digital pictures ;
Pattern Recognition, vol.1. 1968. pp 33-61.
- [Rosenfeld 73] A. Rosenfeld ; E. Johnston :
Angle detection on digital curves ;
IEEE Trans. on Comp., n° 9. 1973. pp 875-878.
- [Rosenfeld 82] A. Rosenfeld ; A.C. Kak :
Digital image processing ;
Academic Press, New-York. 1982.
- [Russ 90] J.C. Russ :
Computer-assisted microscopy : the measurement and analysis of images ;
Plenum Press. 1990.
- [Sakarovitch 84] M. Sakarovitch :
Optimisation combinatoire : méthodes mathématiques et algorithmiques. Programmation discrète ;
Hermann éd.- Enseignement des Sciences - 1984.
- [Samet 80] H. Samet :
Region representation : quadtrees from binary arrays ;
Computer, Graphics & Image Processing, vol. 13. 1980. pp 88-93.
- [Samet 82] H. Samet :
Quadtrees and medial axis transforms ;
6th International Conference on Pattern Recognition. Munich. 1982. pp 184-187.
- [Samet 83] H. Samet :
A quadtree medial axis transform ;
Communications of ACM, vol. 26, n° 9. Septembre 1983. pp 680-693.
- [Samet 88] H. Samet, M. Tamminen :
Efficient component labelling of images of arbitrary dimension represented by linear bintrees ;
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-10, n° 4. Juillet 1988. pp 579-586.
- [Samet 90a] H. Samet :
The design and analysis of spatial data structures ;
Addison Wesley, 1990.
- [Samet 90b] H. Samet :
Applications of spatial data structures ;
Addison Wesley, 1990.

- [Sequeira 87] J. Sequeira :
Modélisation interactive d'objets de forme complexe à partir de données hétérogènes ;
Thèse Université de Franche-Comté Besançon. n° 223. Novembre 1987.
- [Shapiro 79] L.G. Shapiro ; R.M. Haralick :
Decomposition of two dimensional shapes by graph - theoretic clustering ;
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1, n° 1. Janvier 1979. pp 10-20.
- [Shapiro 80] L.G. Shapiro :
A structural model of shape ;
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-2, n° 2. Mars 1980. pp 111-126.
- [Shapiro 85] L.G. Shapiro ; R.M. Haralick :
A metric for comparing relational descriptions ;
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-7, n° 1. Janvier 1985. pp 90-93.
- [Shirai 87] Y. Shirai :
Three-dimensional computer vision ;
Springer-Verlag.
- [Skordas 88] T. Skordas :
Mise en correspondance et reconstruction stéréo utilisant une description structurelle des images ;
Thèse de l'Institut National Polytechnique de Grenoble. 1988.
- [Srihari 81] S.N. Srihari :
Representation of three-dimensional digital images ;
Computing Surveys, vol. 13, n° 4. Décembre 1981. pp 399-424.
- [Srisuresh 83] P. Srisuresh ; S.N. Srihari :
A shrinking algorithm for three dimensional objets ;
Computer Vision & Pattern Recognition, Juin 83, Washington DC, pp 392-393.
- [Stein 90] F. Stein ; G. Médioni :
Efficient two dimensional object recognition ;
10th International Conference on Pattern Recognition, Atlantic City, Juin 1990, pp 13-17.
- [Talbi 91] E.G. Talbi ; P.Bessière :
Genetic parallel algorithm : performances and applications ;
Int. Conf. on Novel Methods in Optimization, Copenhagen, Danemark. Février 1991. à paraître.
- [Thiel 91a] E. Thiel ; A. Montanvert :
Approximation discrète de la distance euclidienne pour l'analyse d'images : étude et amélioration des distances du chanfrein ;
Accepté pour le 8^{ème} Congrès R.F.I.A. Lyon - Villeurbanne. 25-29 Novembre 1991. à paraître.

- [Thiel 91b] E. Thiel :
Améliorations des distances du chanfrein. Construction la ligne médiane et décomposition de formes ;
Rapport de D.E.A. de Mathématiques Appliquées. Université Joseph Fourier. Grenoble I. Juin 1991.
- [Toriwaki 82] J.i. Toriwaki, S. Yokoi, T. Yonekura, T. Fukumura :
Topological properties and topology-preserving transformation of three dimensional binary pictures ;
6th International Conference on Pattern Recognition. Munich. 1982. pp 414-419.
- [Tsao 81] Y.F. Tsao ; K.S. Fu :
A parallel thinning algorithm for 3D pictures ;
Computer Graphics & Image Processing, vol. 17. 1981. pp 315-331.
- [Udupa 82] J.K. Udupa ; S.N. Srihari ; G.T. Hermann :
Boundary detection in multidimensions ;
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-4, n° 1. Janvier 1982. pp 41-50.
- [Veen 77] A. Veen ; L.D. Peachey :
TROTS : A computer graphics system for three-dimensional reconstruction from serial sections ;
Computer & Graphics, vol. 2. 1977. pp 135-150.
- [Wu 77] S.C. Wu ; J.F. Abel ; D.P. Greenberg :
An interactive computer graphics approach to surface representation ;
Communications of ACM, vol. 20, n° 10. Octobre 1977. pp 703-712.
- [Yau 84] M.M. Yau :
Generating quadtrees of cross sections from octrees ;
Computer Vision, Graphics & Image Processing, 27. 1984. pp 211-238.
- [Yokoi 75] S. Yokoi ; J.I. Toriwaki ; T. Fukumura :
An analysis of topological properties of digitized binary pictures using local features ;
Computer, Graphics & Image Processing, n° 4. 1975. pp 63-73.

Annexe

Cette annexe présente des résultats obtenus lors de l'implémentation des différentes méthodes proposées au long de ce mémoire.

Tout d'abord différents modes de squelettisation sont comparés (cf. § 3.2.3 et § 3.2.4).

Ensuite différents exemples de squelettes filaires sont présentés (cf. § 3.2.4.3).

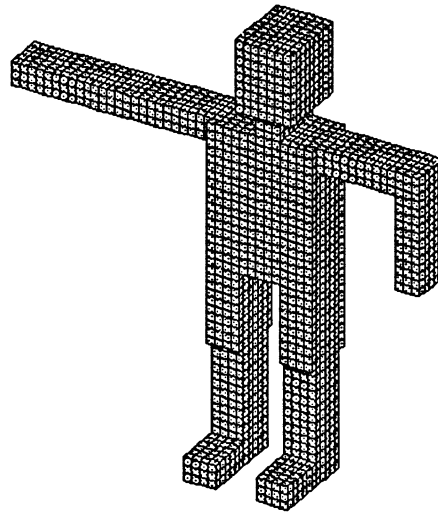
Un exemple d'utilisation du squelette filaire pour la connexion de l'axe médian est présenté ensuite (§ 3.2.4.5).

Cet exemple est suivi de la présentation de la base d'exemples utilisée pour la recherche de cliques (§ 5.2.3).

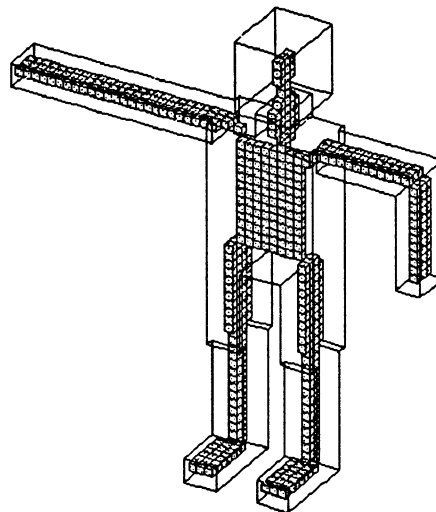
Différents résultats de mise en correspondance sont ensuite donnés (cf § 5.3 et § 6.1.2).

Enfin des résultats de reconstruction tridimensionnelle issus de l'inférence de graphes sont présentés (cf. § 6.3).

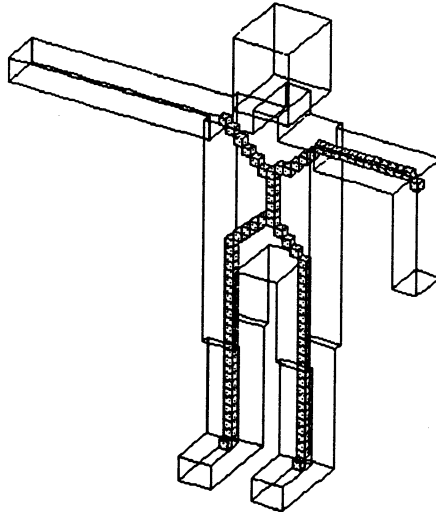
Différents squelettes



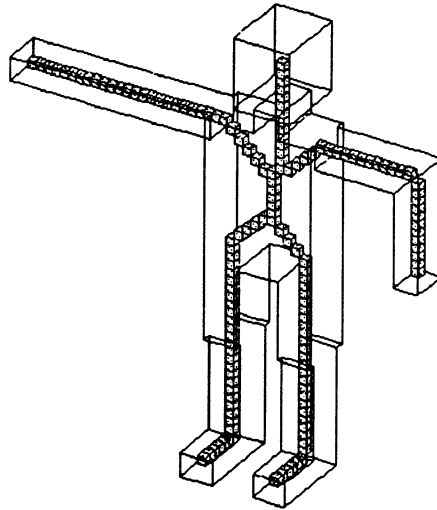
Objet initial (4322 voxels).



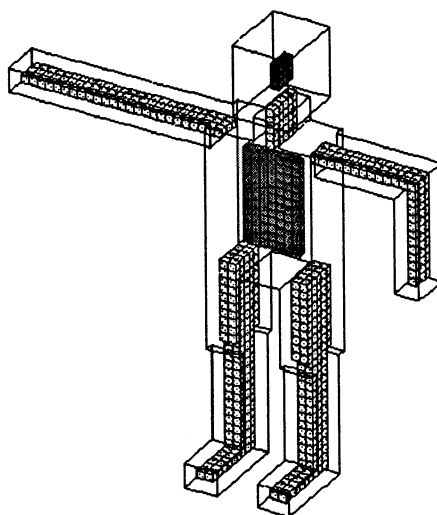
Squelette obtenu avec les conditions 2 et 3 (475 voxels).



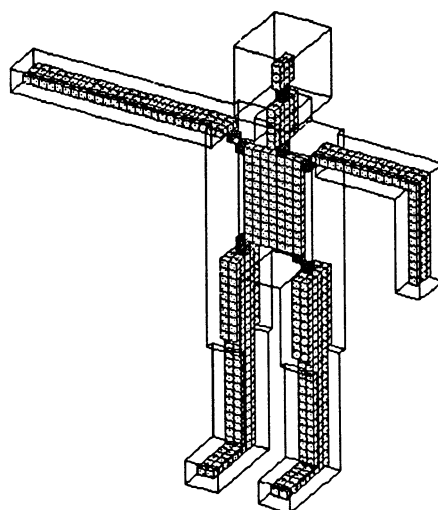
Squelette obtenu avec la condition 2 seule (90 voxels).



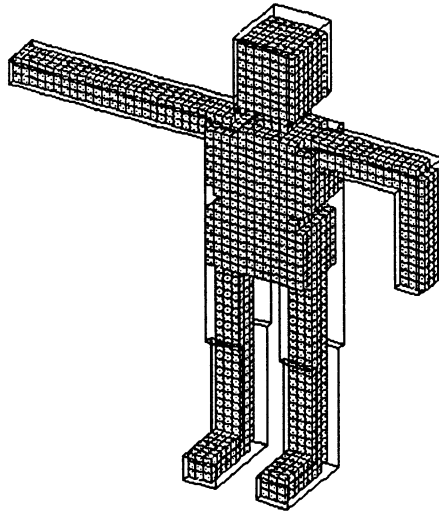
Squelette obtenu avec la condition 2 et la nouvelle condition 4 (148 voxels).



Axe médian (672 voxels).

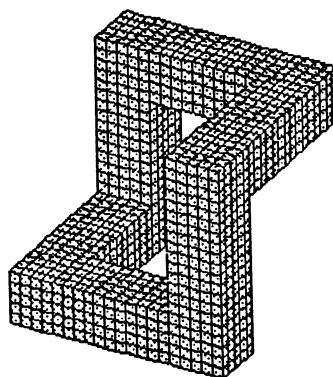


Ligne médiane obtenue avec les conditions 2 et 5 (684 voxels).

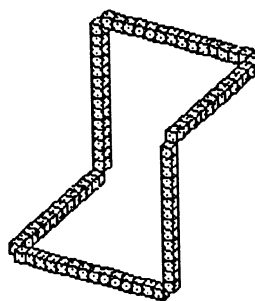


Objet reconstruit à partir du squelette filaire (conditions 2 et 4) et des pondérations de l'axe médian (2907 voxels).

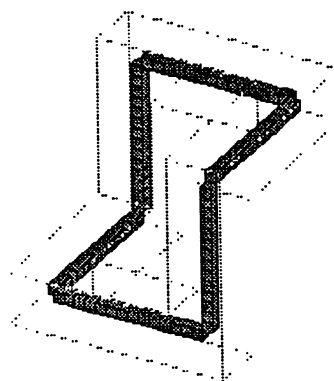
Autres exemples de squelettes d'objets de synthèse



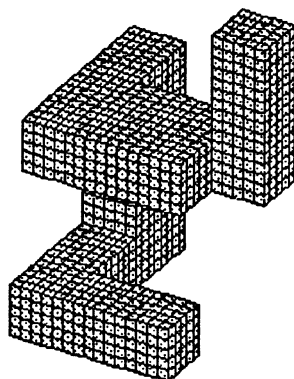
Objet initial.



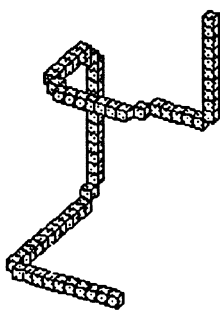
Squelette filaire.



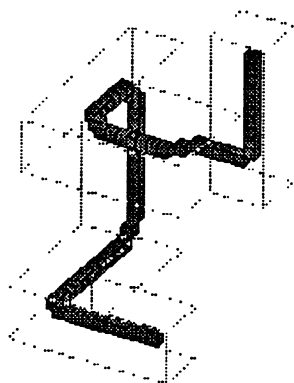
Squelette filaire comparé à la forme initiale



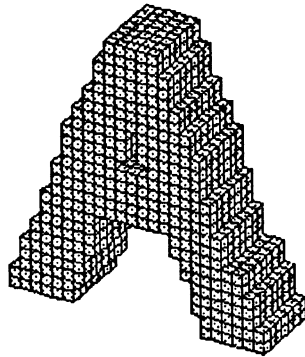
Objet initial.



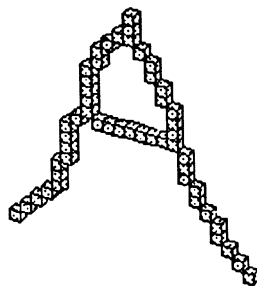
Squelette filaire.



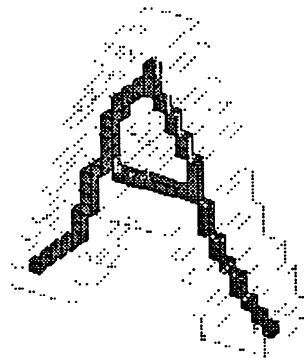
Squelette filaire comparé à la forme initiale.



Objet initial.

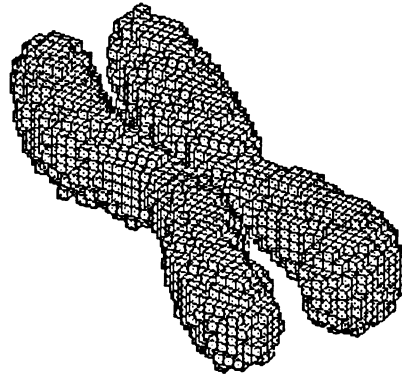


Squelette filaire.

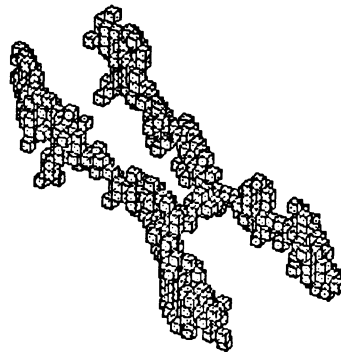


Squelette filaire comparé à la forme initiale.

**Exemple de squelette filaire d'une forme réelle acquise par
microscopie confocale**



Chromosome (4865 voxels).

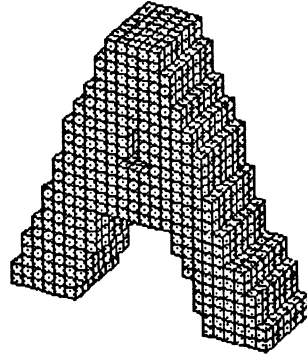


Squelette filaire du chromosome (414 voxels).

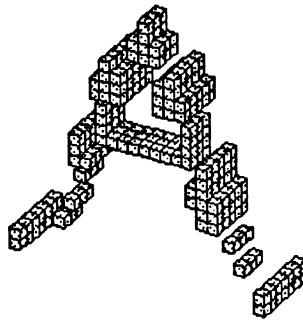


Squelette filaire comparé à la forme initiale.

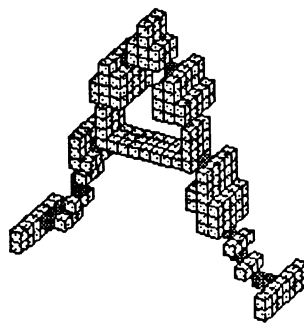
Exemple de connexion de l'axe médian



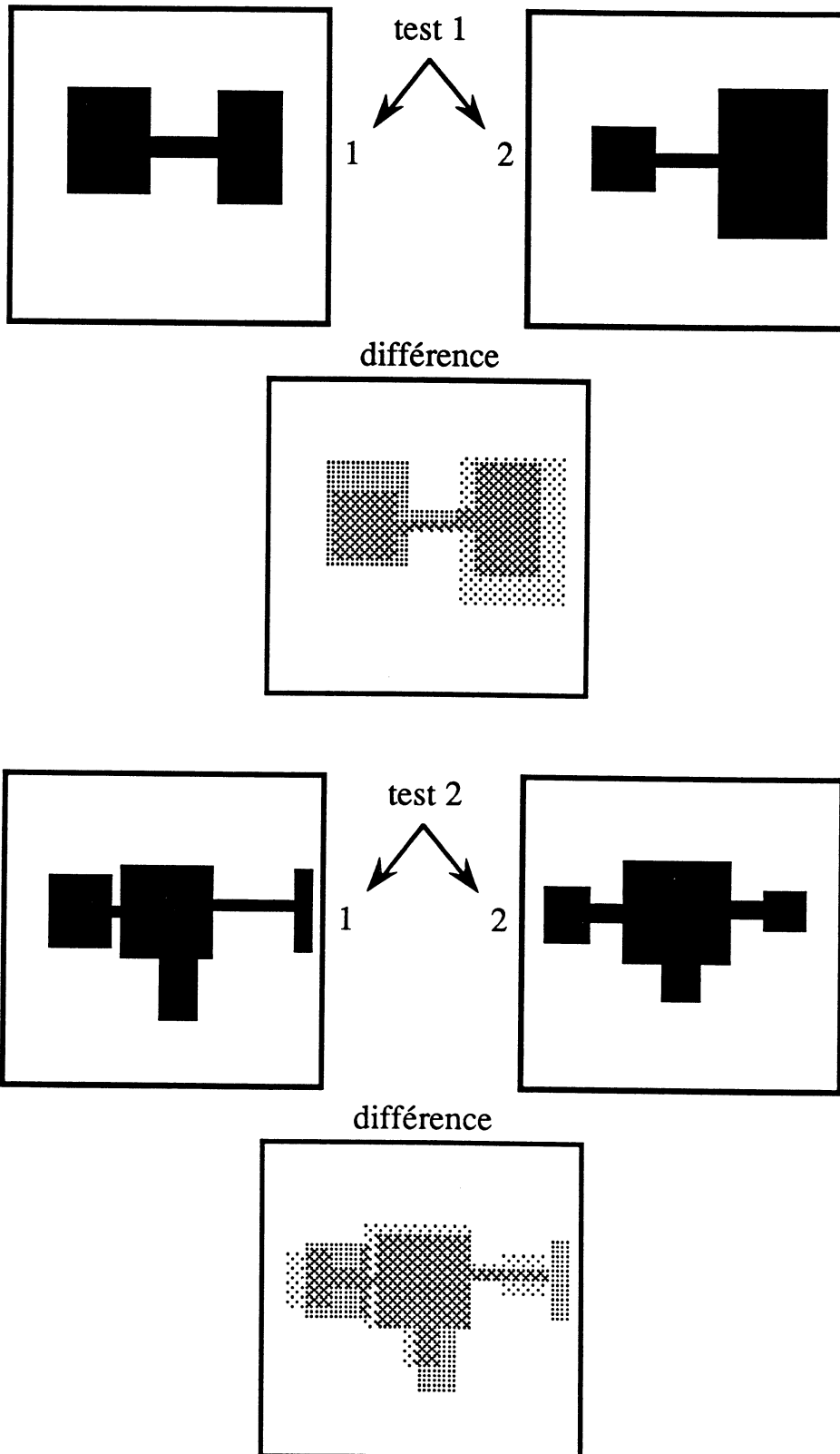
Objet initial.

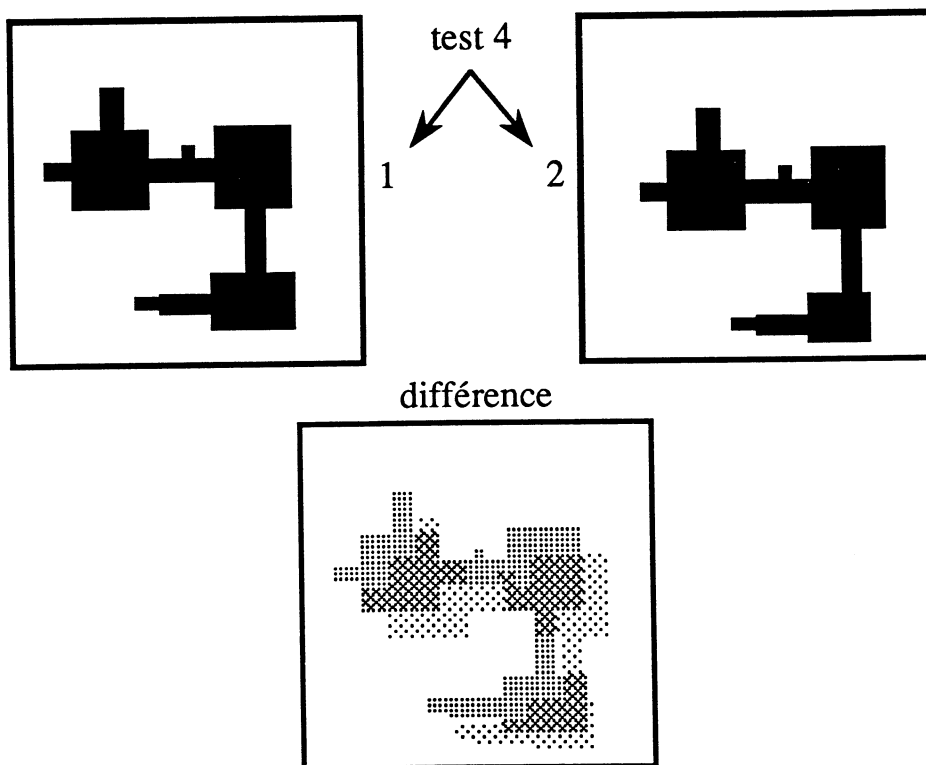
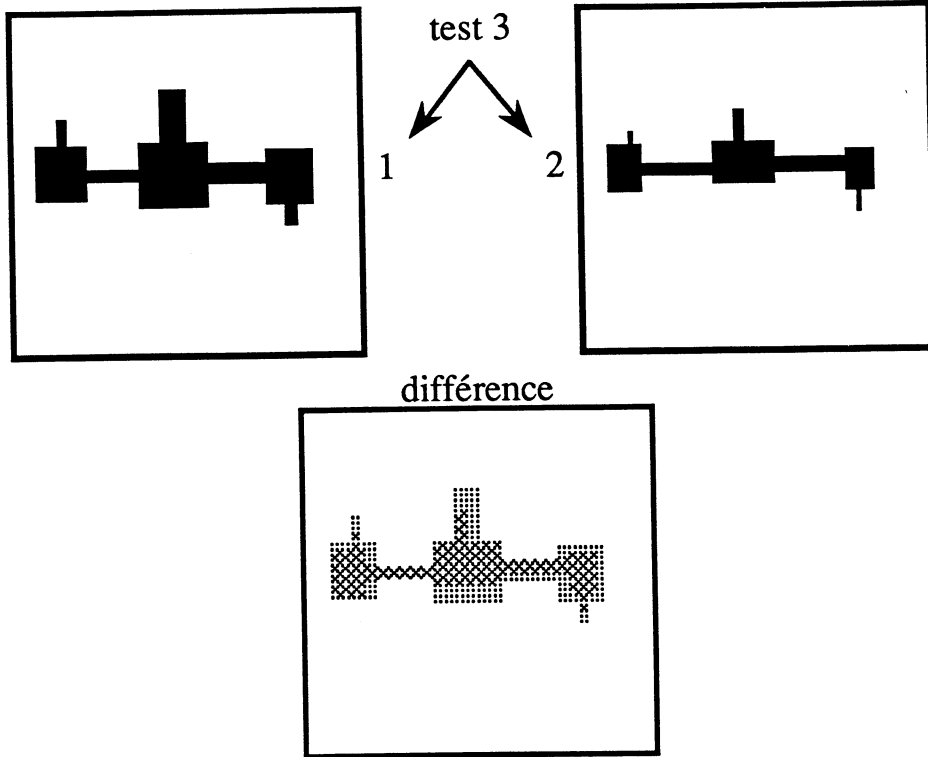


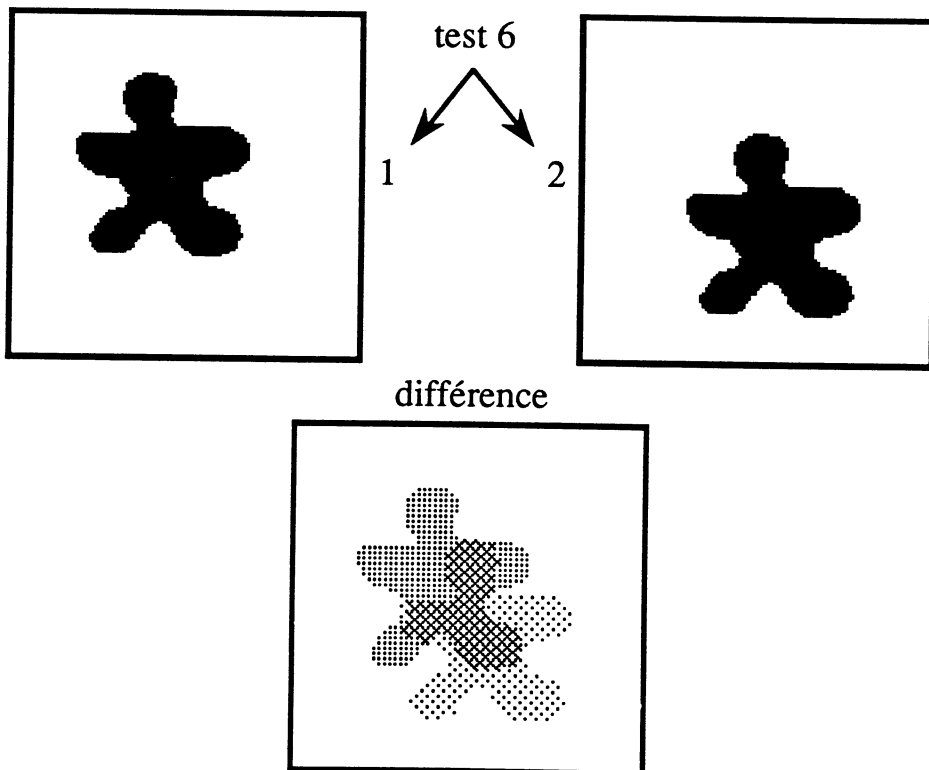
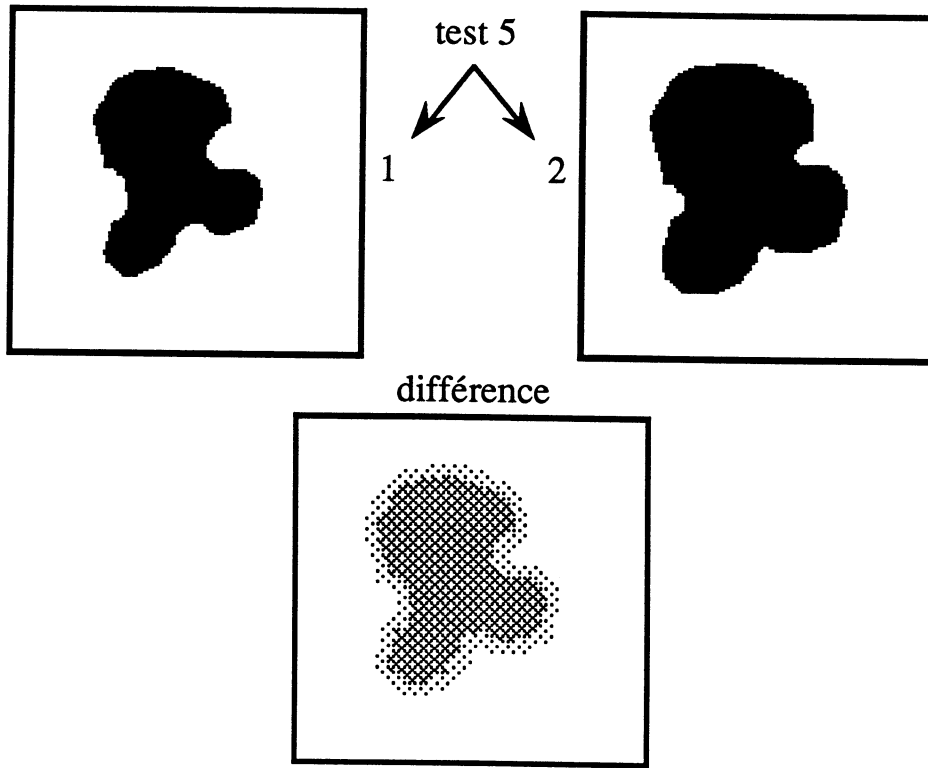
Axe médian (déconnecté).



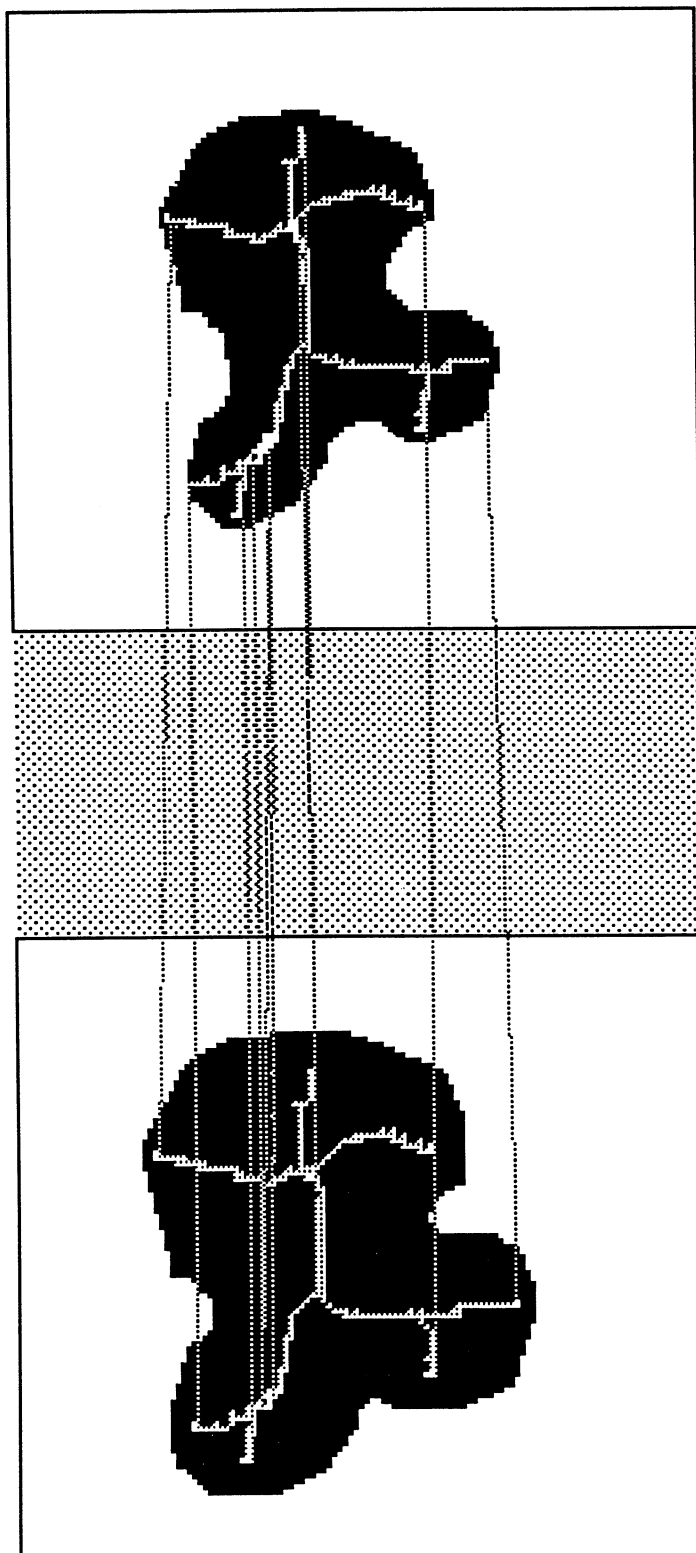
Ligne médiane (axe médian connecté).

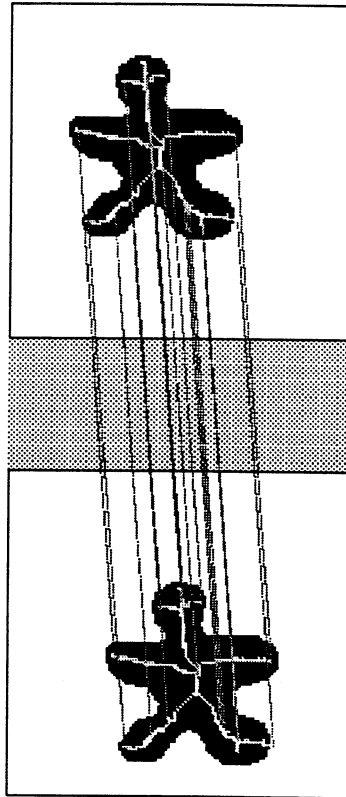
Base d'exemples utilisée pour la recherche de cliques (§ 5.2.3)



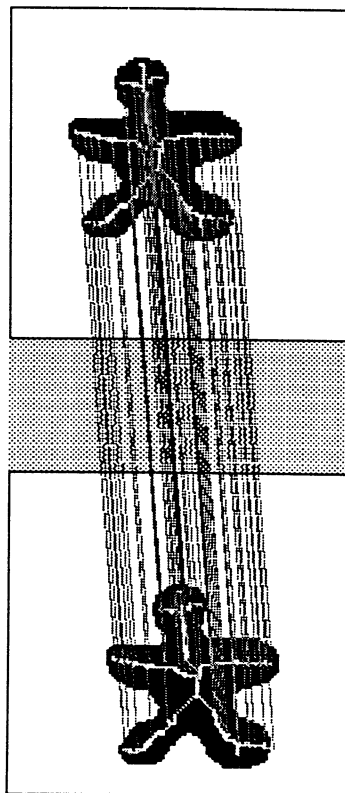


Exemples de mise en correspondance



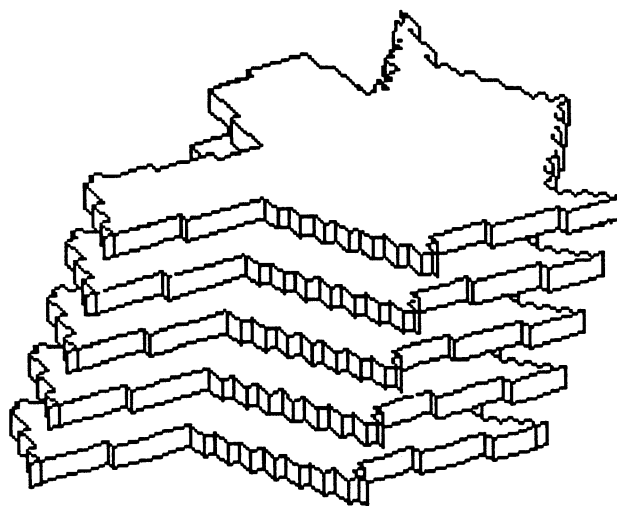


Mise en correspondance initiale.

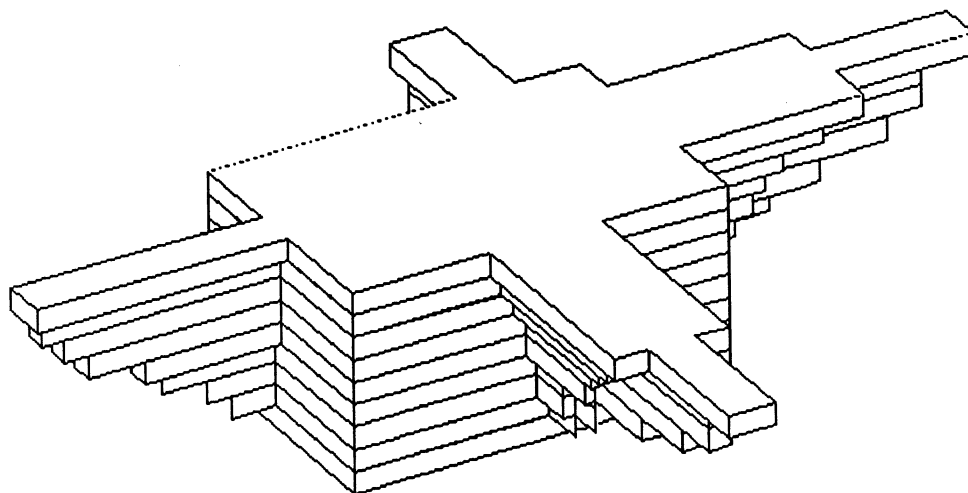


Mise en correspondance après propagation des
appariements.

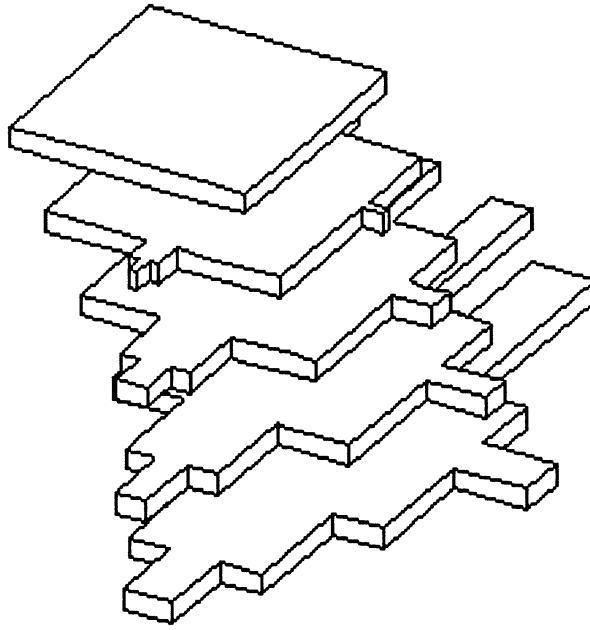
Exemples de reconstruction tridimensionnelle



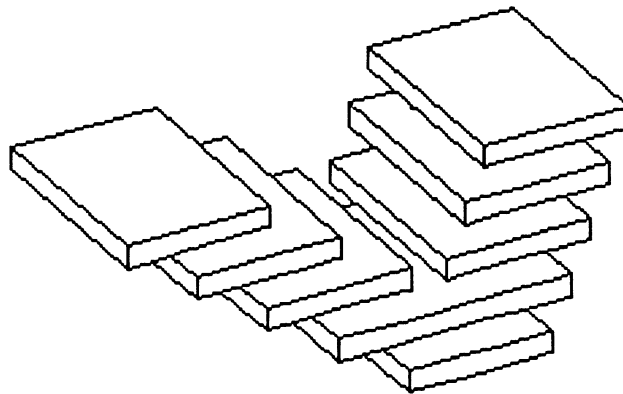
Translation.



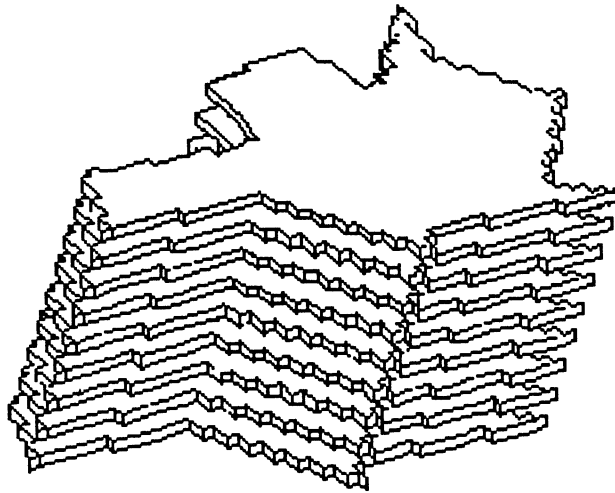
Apparition de déformations.



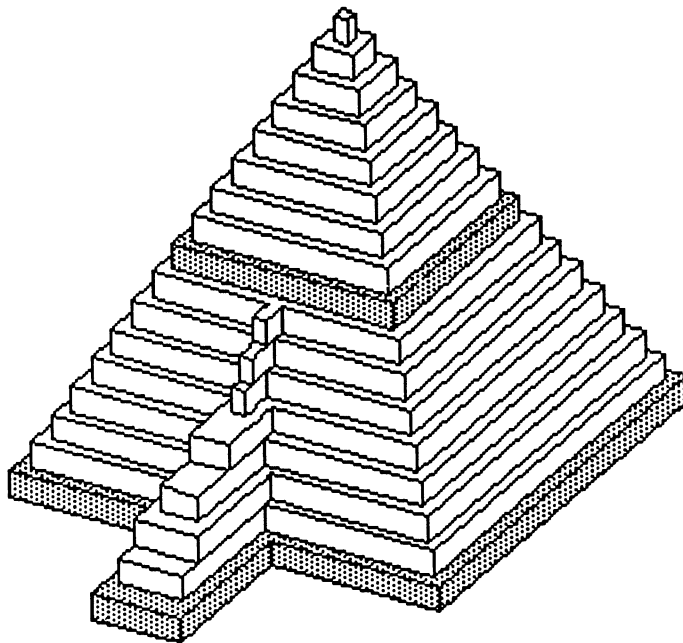
Combinaison de translation et d'apparition de déformations.



Apparition de branchements.



Pour réduire la discontinuité on peut inférer un plus grand nombre de coupes.



On peut aussi inférer des coupes à l'extérieur de celles données (en gris).

Table des figures

1.1.	Différents résultats de reconstruction	14
1.2.	Un objet et les coupes qui en sont extraites	16
1.3.	Les coupes doivent être significatives de l'objet.....	16
1.4.	Dans certains cas des ambiguïtés sont possibles.....	17
1.5.	Recalage des coupes	19
1.6.	Exemple de segmentation.....	20
1.7.	Triangulation des contours	22
1.8.	Graphe toroïde.....	25
1.9.	Trois exemples d'objets posant des problèmes	27
1.10.	Exemple de sous-objets et connexions associées.....	27
1.11.	Exemple de rapide changement de forme.....	28
1.12.	Reconstruction utilisant l'inférence de portions de contour	29
1.13.	Diagramme de Voronoï et triangulation de Delaunay.....	30
1.14.	Exemples de polygones non contenus dans la triangulation de Delaunay de leurs sommets.....	30
1.15.	Exemple d'objet construit par simple empilement	32
2.1.	Maillages carré, triangulaire et hexagonal	36
2.2.	Codage matriciel d'une image.....	36
2.3.	$B_4 (P , 1)$ et propagation de la distance d_4	38
2.4.	$B_8 (P , 1)$ et propagation de la distance d_8	39
2.5.	Masque du chanfrein 3-4 et une boule associée.....	40

2.6.	Premier octant du masque du chanfrein 11 x 11.....	41
2.7.	Exemples des boules pour les distances du chanfrein.....	41
2.8.	Chemin connexe selon d_4 et selon d_8	42
2.9.	Courbes.....	42
2.10.	Composantes connexes.....	42
2.11.	Une image est composée d'objets, du fond et de trous.....	43
2.12.	Illustration du théorème de Jordan.....	44
2.13.	Contours.....	44
2.14.	Différents contours.....	45
2.15.	Codage des contours par la méthode de Freeman.....	46
2.16.	Un objet, son partitionnement et le <i>quadtree</i> correspondant.....	47
2.17.	Non invariance en translation du <i>quadtree</i>	48
2.18.	<i>Quadtree Medial Axis</i>	49
2.19.	Arbre binaire.....	49
2.20.	Squelette par propagation d'un feu de brousse.....	50
2.21.	Sensibilité au bruit du squelette.....	51
2.22.	Exemple d'axe médian en d_8	52
2.23.	Non connexité de l'axe médian.....	53
2.24.	Ligne médiane.....	53
2.25.	Graphe de la ligne médiane.....	54
2.26.	Graphe de la ligne médiane.....	55
3.1.	Boule euclidienne discrète.....	59
3.2.	6-voisinage d'un point et boule discrète selon d_6	60
3.3.	Masques aller et retour pour d_6	60
3.4.	26-voisinage d'un point et boule discrète selon d_{26}	61
3.5.	Masques aller et retour pour d_{26}	62
3.6.	18-voisinage d'un point et boule discrète selon d_{18}	62
3.7.	Masques aller et retour pour d_{18}	63
3.8.	Masques aller et retour pour la distance du chanfrein 3-4-5.....	63
3.9.	Différentes boules : pour les distances octogonales 6/18, 6/26, 18/28 et pour la distance du chanfrein 3-4-5.....	64
3.10.	Les 6 directions de l'espace.....	65
3.11.	Objets 3D : objet fermé ; objet avec anse ; objet avec cavité ; invariants topologiques.....	66
3.12.	Illusions.....	67
3.13.	Un objet et l'arbre CSG correspondant.....	69

3.14. Un objet, les codes pour l'obtention de l' <i>octree</i> correspondant.....	70
3.15. Un objet et sa décomposition pour construire l'arbre binaire correspondant	71
3.16. Cellule coin, cellule angle et cellule surface	71
3.17. Axe médian d'un parallélépipède	72
3.18. L'axe médian est déconnecté.....	73
3.19. A partir d'un parallélépipède veut-on obtenir un squelette surfacique ou un squelette filaire.....	75
3.20. L'objet initial et le squelette surfacique	77
3.21. Evaluation des conditions (2) et (3).....	78
3.22. Squelette obtenu avec la condition (2) seule.....	78
3.23. Objet tridimensionnel et le squelette obtenu en utilisant les conditions (2) et (4).....	79
3.24. Squelette filaire	80
3.25. Parcours d'un volume à l'aide du parcours de son squelette surfacique défini lui-même par le parcours du squelette filaire	81
3.26. Connexion de l'axe médian pour obtenir une ligne médiane	82
3.27. Exemple de ligne médiane un peu trop longue.....	83
3.28. Axe médian et ligne médiane.....	84
3.29. Différentes méthodes de codage appliquées à un objet composé de 4322 voxels.....	85
3.30. Squelettes de coupes 2D et coupes 2D de squelette 3D	87
3.31. La coupe 2D du squelette d'un objet peut être différente du squelette 2D de cette même coupe.....	88
4.1. Exemple de résultats obtenus par assortiment élastique.....	94
4.2. Différentes approches par les contours	95
4.3. Approche par les régions choisie	96
5.1. Mise en correspondance et isomorphismes.....	101
5.2. Méthode de recherche par retour arrière.....	106
5.3. Graphe de mise en correspondance	108
5.4. Filtrage du graphe de la ligne médiane.....	112
5.5. Compatibilité des appariements.....	113
5.6. Table des pondérations liées aux types de sommets.....	115
5.7. Pondération selon la taille et le nombre de boules codées	116

5.9.	Calcul des degrés.....	117
5.10.	Table des pondérations liées à la configuration des graphes de la ligne médiane au niveau des sommets appariés	118
5.11.	Pondération liée à la position de la composante codée par rapport à la forme globale	118
5.12.	Deux formes et le graphe de mise en correspondance construit.....	120
5.13.	Optimums locaux et “barrières de potentiel”	126
5.14.	Exemple d'évolution du coût quand la température décroît.....	128
5.15.	Opérateurs génétiques : crossing-over, mutation	130
5.16.	Exemple d'évaluation de la fonction coût avec les générations	131
5.17.	Exemple de résultat de recherche de cliques.....	134
5.18.	Recherche de cliques.....	135
6.1.	Appariement de proéminences larges : cas simple.....	138
6.2.	Appariement de proéminence larges pour des nombres de proéminences larges différents.....	139
6.3.	Propagation des appariements : appariements multiples	140
6.4.	Propagation des appariements : appariements multiples avec chemins de longueurs différentes.....	140
6.5.	Propagation des appariements : cas plus complexe.....	141
6.6.	Appariement des proéminences strictes peu déformantes.....	142
6.7.	Situation paradoxale lors de l'appariement des proéminences strictes peu déformantes	142
6.8.	Mise en correspondance	143
6.9.	Inférence dans le cas d'un appariement : sommet principal ↔ sommet principal	145
6.10.	Inférence dans le cas d'un appariement : sommet principal ↔ sommet de proéminence stricte.....	146
6.11.	Inférence dans le cas d'un appariement : sommet principal ↔ sommet de proéminence large	147
6.12.	Inférence dans le cas d'un appariement : sommet de proéminence stricte ↔ sommet de proéminence stricte	148
6.13.	Inférence dans le cas d'un appariement : sommet de proéminence large ↔ sommet de proéminence large.....	149
6.14.	Inférence dans le cas d'un appariement : sommet de proéminence stricte ↔ sommet de proéminence large.....	150

6.15. Inférence dans le cas d'un appariement : sommet de rétrécissement \leftrightarrow sommet de rétrécissement	151
6.16. Inférence dans le cas d'un appariement : sommet de rétrécissement \leftrightarrow sommet principal	152
6.17. Inférence dans le cas d'un appariement : sommet de rétrécissement \leftrightarrow sommet de proéminence stricte	153
6.18. Inférence dans le cas d'un appariement : sommet de rétrécissement \leftrightarrow sommet de proéminence large.....	154
6.19. Inférence dans le cas d'un appariement multiple : sommet principal \leftrightarrow sommets principaux.....	155
6.20. Inférence dans le cas d'un appariement multiple : sommet principal \leftrightarrow sommets principaux, sommets de rétrécissement.....	156
6.21. Résultat de l'inférence dans le cas de l'appariement : sommet principal \leftrightarrow sommet principal	158
6.22. Résultat de l'inférence dans le cas de l'appariement : sommet principal \leftrightarrow sommet de proéminence stricte	158
6.23. Résultat de l'inférence dans le cas de l'appariement : sommet principal \leftrightarrow sommet de proéminence large.....	159
6.24. Résultat de l'inférence dans le cas de l'appariement : sommet de proéminence stricte \leftrightarrow sommet de proéminence stricte.....	159
6.25. Résultat de l'inférence dans le cas de l'appariement : sommet de proéminence large \leftrightarrow sommet de proéminence large	159
6.26. Résultat de l'inférence dans le cas de l'appariement : sommet de proéminence stricte \leftrightarrow sommet de proéminence large	160
6.27. Résultat de l'inférence dans le cas de l'appariement : sommet de rétrécissement \leftrightarrow sommet de rétrécissement	160
6.28. Résultat de l'inférence dans le cas de l'appariement : sommet de rétrécissement \leftrightarrow sommet principal.....	160
6.29. Résultat de l'inférence dans le cas de l'appariement : sommet de rétrécissement \leftrightarrow sommet de proéminence stricte.....	161
6.30. Résultat de l'inférence dans le cas de l'appariement : sommet de rétrécissement \leftrightarrow sommet de proéminence large	161
6.31. Résultat de l'inférence dans le cas de l'appariement : sommets principaux \leftrightarrow sommet principal	161
7.1. Décomposition en éléments convexes.....	167

AUTORISATION DE SOUTENANCE

DOCTORAT DE L'UNIVERSITE JOSEPH FOURIER - GRENOBLE 1

Vu les dispositions de l'Arrêté du 5 juillet 1984,

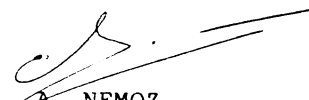
Vu les rapports de **Monsieur. Philippe. CINQUIN**

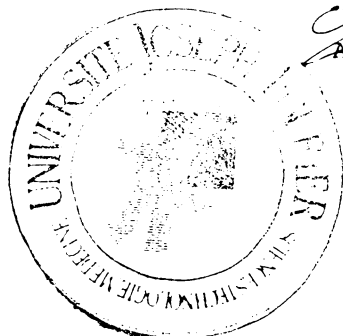
Monsieur. Francis. SCHMITT

Monsieur. Franck. ROLLANDest autorisé(e)
à présenter une thèse en vue de l'obtention du **doctorat de l'Université**....
Joseph Fourier GRENOBLE I

Grenoble, le

Le Président de l'Université
Joseph Fourier - Grenoble 1


A. NEMOZ



Résumé

L'objectif des travaux présentés dans ce mémoire est la représentation de formes tridimensionnelles ainsi que la reconstruction tridimensionnelle à partir de coupes sériées.

Après avoir défini le cadre exact de la reconstruction tridimensionnelle à partir de coupes sériées, le premier chapitre analyse les différentes étapes et les méthodes généralement employées pour effectuer cette reconstruction. Les chapitres 2 et 3 s'attachent à la description et à la représentation de formes tant bidimensionnelles que tridimensionnelles. Le chapitre 3 développe ainsi une méthode de squelettisation tridimensionnelle permettant d'obtenir un squelette filaire. Le chapitre 4 est consacré à la description de deux méthodes de reconstruction tridimensionnelle opérant par inférence de formes. La méthode utilisant l'inférence structurelle de graphes de la ligne médiane est ensuite développée. Elle passe par une nécessaire étape, chapitre 5, de mise en correspondance de graphes de la ligne médiane. Différentes méthodes de mise en correspondance sont présentées, finalement la méthode procédant par recherche de cliques dans un graphe est développée. La construction du graphe de mise en correspondance est tout d'abord présentée, ensuite la recherche de cliques est abordée à l'aide de méthodes heuristiques de parcours (recuit simulé et algorithmes génétiques). Le chapitre 6 développe le processus d'inférence de graphes de la ligne médiane mis en œuvre, il permet d'inférer des coupes intermédiaires qui sont empilées pour finalement donner le volume tridimensionnel recherché. Le chapitre 7 est consacré à la présentation des divers résultats et de leurs conséquences, il présente aussi d'autres applications possibles des différentes techniques développées.

Mots Clés : Analyse d'images - Reconstruction 3D à partir de coupes 2D - Squelettes 3D - Mise en correspondance de graphes - Recherche de cliques - Recuit simulé - Algorithmes génétiques - Inférence structurelle.

