



**HAL**  
open science

# Modèles déformables surfaciques, implicites et volumiques, pour l'imagerie médicale

Eric Bittar

► **To cite this version:**

Eric Bittar. Modèles déformables surfaciques, implicites et volumiques, pour l'imagerie médicale. Autre [cs.OH]. Université Joseph-Fourier - Grenoble I, 1998. Français. NNT: . tel-00004869

**HAL Id: tel-00004869**

**<https://theses.hal.science/tel-00004869v1>**

Submitted on 19 Feb 2004

**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.

# THESE

présentée par

**Eric BITTAR**

pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITE GRENOBLE I**

*Discipline : Informatique*

## **Modèles déformables surfaceutiques, implicites et volumiques, pour l'imagerie médicale**

Soutenue le

4 Mars 1998

devant le jury composé de

**Christian Roux** Président

**Isabelle Bloch**  
**Olivier Monga** Rapporteurs

**Philippe Cinquin**  
**Claude Puech** Directeurs

Thèse préparée au laboratoire **TIMC-IMAG**



---

# Remerciements

Je tiens à remercier Isabelle Bloch et Olivier Monga pour avoir accepté d'être rapporteurs de cette thèse. Je remercie Christian Roux d'avoir bien voulu participer au jury.

Je remercie Claude Puech d'avoir co-dirigé ce travail, et de m'avoir fait profiter de sa hauteur de vue scientifique. Je remercie particulièrement Philippe Cinquin pour ses encouragements dans cette aventure tumultueuse, et pour sa devise *ama et fac quod vis*.

Je remercie Marie-Paule Cani-Gascuel pour sa gentillesse et sa grande connaissance du domaine scientifique et implicite, et Nicolas Tsingos pour sa vitalité et son travail. Soyez tous deux remerciés pour cette enrichissante collaboration sur les surfaces implicites.

Un remerciement spécialement chaleureux pour Stéphane Lavallée, grâce à la disponibilité, la patience et les conseils duquel le travail sur le modèle volumique déformable a pu voir le jour.

La nature du travail de la thèse sert de révélateur aux petites et aux grandes amitiés. Je remercie tous ceux qui m'ont soutenu pendant ces années, Catherine Barbe-Zoppis, Yann Menguy, Agnès Poyet, Mahnu Promayon, Yves Delnondedieu. Je remercie également pour leurs coups de main et leur personnalité sympathique Ali Hamadeh, Delphine Henry, Benoit Mollard, Matthieu Chabanas et Markus Fleute et aussi les piliers de l'équipe GMCAO, Jocelyne Troccaz, Laurent Desbat, Guillaume Champleboux et bien sûr Maribel Chenin.

Je remercie encore les artisans de l'équilibre, professionnels de la relation authentique ou de la joie et de l'amitié : Hervé, Annie et Jean-Dominique, Daniel, Elsa et Nathalie, Christian Deville-Cavellin et Jean Collas.

Je remercie enfin celle qui est la plus proche de moi, pour son amour et sa compréhension, ma femme Marie-Laure.



# Table des matières

<b>Remerciements</b>	<b>3</b>
<b>Présentation</b>	<b>9</b>
Introduction . . . . .	9
Applications . . . . .	10
Organisation du document . . . . .	12
<b>I Etude bibliographique</b>	<b>13</b>
<b>I•1 Formalismes</b>	<b>15</b>
I•1.1 Modélisation des modèles déformables . . . . .	15
I•1.2 Le formalisme des snakes . . . . .	18
I•1.3 Modèles déformables et recalage non-rigide . . . . .	19
I•1.4 Chaîne de traitement de l'information . . . . .	19
<b>I•2 Caractéristiques de liaison</b>	<b>21</b>
I•2.1 Caractéristiques de bas niveau . . . . .	21
I•2.2 Fonction de liaison . . . . .	26
<b>I•3 Représentation géométrique</b>	<b>31</b>
I•3.1 Modèles surfaciques . . . . .	31
I•3.2 Modèles implicites . . . . .	34
I•3.3 Modèles volumiques . . . . .	37
I•3.4 Discussion . . . . .	39
<b>I•4 Déformation et interactivité</b>	<b>41</b>
I•4.1 Déformations indirectes . . . . .	41
I•4.2 Déformations directes . . . . .	44
I•4.3 Combinaison de déformations . . . . .	47
I•4.4 Interactivité . . . . .	47
I•4.5 Discussion . . . . .	49
<b>I•5 Déformabilité et contrôle</b>	<b>51</b>
I•5.1 Déformabilité . . . . .	51
I•5.2 Contrôle . . . . .	52
I•5.3 Optimisation d'une fonctionnelle . . . . .	52

<b>I●6</b>	<b>Discussion et conclusion</b>	<b>55</b>
<b>II</b>	<b>Un modèle surfacique déformable</b>	<b>57</b>
<b>II●1</b>	<b>Présentation</b>	<b>59</b>
II●1.1	Le modèle des $\delta$ -snakes . . . . .	59
II●1.2	Déformation . . . . .	60
II●1.3	Champ potentiel . . . . .	61
II●1.4	Apport de l'interactivité . . . . .	63
<b>II●2</b>	<b>Interactivite</b>	<b>66</b>
II●2.1	Modélisation de l'environnement . . . . .	66
II●2.2	Déformation interactive . . . . .	67
II●2.3	Comparaison . . . . .	68
<b>II●3</b>	<b>Resultats</b>	<b>70</b>
II●3.1	Présentation de l'application . . . . .	70
II●3.2	Resultats . . . . .	70
<b>II●4</b>	<b>Conclusion de la partie II</b>	<b>74</b>
<b>III</b>	<b>Un modèle implicite déformable</b>	<b>75</b>
<b>III●1</b>	<b>Introduction</b>	<b>77</b>
III●1.1	Reconstruction de la surface d'un objet . . . . .	77
III●1.2	Reconstruction de forme à l'aide de primitives implicites . . . . .	78
III●1.3	Plan de la partie III . . . . .	79
<b>III●2</b>	<b>Première approche</b>	<b>80</b>
III●2.1	Définition des fonctions de champ locales . . . . .	80
III●2.2	Bases de la reconstruction semi-automatique . . . . .	81
III●2.3	Un algorithme de subdivision efficace . . . . .	81
III●2.4	Fenêtres de reconstruction . . . . .	82
<b>III●3</b>	<b>Interactivité et premiers résultats</b>	<b>84</b>
III●3.1	Approche semi-automatique . . . . .	84
III●3.2	Resultats . . . . .	84
<b>III●4</b>	<b>Construction de l'axe médian</b>	<b>89</b>
III●4.1	Présentation . . . . .	89
III●4.2	Carte de distance . . . . .	90
III●4.3	Extraction de l'axe médian . . . . .	91
III●4.4	Ajuster la résolution de l'axe médian en fonction des données . . . . .	91
<b>III●5</b>	<b>2ème approche : reconstruction automatique</b>	<b>93</b>
III●5.1	D'un axe médian à une surface implicite . . . . .	93
III●5.2	Création et affinage de la surface implicite . . . . .	94
<b>III●6</b>	<b>Résultats de l'approche automatique</b>	<b>97</b>
III●6.1	Résultats expérimentaux . . . . .	97

<b>III●7 Conclusion de la partie III</b>	<b>101</b>
III●7.1 Conclusion . . . . .	101
III●7.2 Perspectives . . . . .	101
<b>IV Un modèle volumique déformable</b>	<b>103</b>
<b>IV●1 Présentation</b>	<b>105</b>
IV●1.1 Formulation de la méthode . . . . .	105
IV●1.2 Notion de points aberrants . . . . .	107
IV●1.3 Segmentation par inférence . . . . .	107
IV●1.4 Organisation de cette partie . . . . .	107
<b>IV●2 Caractéristiques de bas niveau</b>	<b>109</b>
IV●2.1 Introduction . . . . .	109
IV●2.2 Choix du vecteur gradient . . . . .	109
IV●2.3 Calcul du gradient 3D . . . . .	109
IV●2.4 Extraction des extrema . . . . .	110
IV●2.5 Seuillage par hystérésis . . . . .	111
IV●2.6 Influence du paramètre $\alpha$ . . . . .	111
<b>IV●3 Distance</b>	<b>113</b>
IV●3.1 Utilisation optimisée des arbres k-d . . . . .	113
IV●3.2 Nature de la distance . . . . .	118
IV●3.3 Transformation des caractéristiques . . . . .	119
<b>IV●4 Déformation et déformabilité</b>	<b>120</b>
IV●4.1 Types de déformations . . . . .	120
IV●4.2 Déformabilité . . . . .	121
IV●4.3 Déformations hiérarchiques par octree spline . . . . .	122
IV●4.4 Déformation et rééchantillonnage . . . . .	123
IV●4.5 Conclusion . . . . .	125
<b>IV●5 Contrôle : Minimisation</b>	<b>126</b>
IV●5.1 Minimisation aux moindres carrés . . . . .	126
IV●5.2 Outliers . . . . .	127
IV●5.3 Minima locaux . . . . .	127
<b>IV●6 Applications</b>	<b>128</b>
IV●6.1 Validation sur des données synthétiques . . . . .	128
IV●6.2 Segmentation d'une vertèbre dans une image TDM . . . . .	129
IV●6.3 Echographie virtuelle . . . . .	137
IV●6.4 Correction des distorsion en IRM . . . . .	138
<b>IV●7 Discussion et Conclusion</b>	<b>142</b>
IV●7.1 Discussion . . . . .	142
IV●7.2 Conclusion . . . . .	143



---

<b>Conclusion</b>	<b>147</b>
Bilan . . . . .	147
Perspectives . . . . .	148
<b>Bibliographie</b>	<b>149</b>
<b>Abstract</b>	<b>159</b>
<b>Résumé</b>	<b>160</b>

## Introduction

### Voir l'intérieur du corps humain

Les dispositifs d'imagerie médicale donnent aux médecins des moyens de visualiser l'intérieur du corps humain. Malheureusement dans la plupart des modalités d'imagerie, le pouvoir de voir est accompagné de la malédiction de nuire. Au cours des décennies, la vision de l'homme au travers du corps de son semblable a suivi une double évolution vers plus de détails et moins de nocivité. Dans les examens irradiants, la sensibilité des détecteurs permet de réduire les doses reçues par le patient, et d'augmenter le nombre de coupes successives, pour obtenir une acquisition réellement tridimensionnelle des organes, c'est-à-dire un ensemble d'informations homogène dans toutes des directions de l'espace, et non plus une série de tranches bidimensionnelles.

Ces images nécessitent de nouveaux dispositifs de traitement, adaptés à l'organisation spatiale 3D des données, mais également à la grande quantité d'information à traiter. Le résultat d'un examen en tomographie par rayon X (scanner X ou TDM) ou en imagerie par résonance magnétique (IRM) peut typiquement représenter un volume de  $256 \times 256 \times 128$  éléments, soit plus de huit million trois cent mille de ces éléments, les voxels.

### Utilisation de la connaissance a priori

Au vu de cette quantité d'information, nous avons décidé de nous appuyer sur de la connaissance a priori pour traiter efficacement les images 3D. Cette connaissance est liée d'une part à la nature de l'image, aux particularités de chaque modalité d'imagerie, mais il s'agit surtout d'une information de haut niveau, structurellement dépendante de la nature du résultat désiré.

Une façon d'intégrer de la connaissance a priori est d'employer des modèles déformables, qui ont la propriété de s'appuyer sur des caractéristiques de bas niveau des images pour guider l'évolution d'une forme de haut niveau dans l'image. Nous nous sommes donc proposé d'étudier ces modèles déformables et leurs applications en traitement d'images médicales.

---

## Modèles déformables tridimensionnels

Les modèles déformables ont fait leurs preuves dans des tâches de segmentation, de mise en correspondance, de suivi de données, de simulation. En médecine, ils s'intègrent dans des outils de visualisation de structures anatomiques, mais aussi d'aide au diagnostic, aide à la thérapeutique ou pour la formation des médecins.

Les premiers modèles déformables étaient des courbes qui évoluaient dans des images 2D. Ils sont connus sous le nom de contours actifs. La connaissance a priori qu'ils contenaient était très primitive, comme par exemple "L'objet représenté par l'image est homotopique à un contour fermé". Cela correspond uniquement à un critère de déformabilité très général. Depuis, les modèles déformables ont été amenés à évoluer dans des images 3D. Ils sont devenus des surfaces, puis des volumes déformables, en prenant en compte l'évolution de la puissance de calcul des ordinateurs.

Notre étude de la littérature des modèles déformables nous a conduit à proposer une modélisation de ces modèles. Nous distinguons cinq composantes. Les **caractéristiques de liaison modèle-données** correspondent à l'information de bas niveau du modèle et de l'image, et à la mesure calculée à partir de ces caractéristiques de l'éloignement entre le modèle et les données. Nous distinguons ensuite la **représentation géométrique**, qui correspond à la forme, la **déformation** que subit cette forme, l'ensemble des contraintes de déformation - que nous appellerons **déformabilité** - qui détermine l'ensemble des formes admissibles, et la méthode de **contrôle** qui fixe l'évolution de la forme en fonction des contraintes.

Grâce à notre modélisation, nous établirons une classification des modèles déformables de la littérature, en nous attachant à relever les choix possibles pour chacune des cinq composantes. Cette analyse théorique permettra de retenir les modèles déformables qui sont pertinents pour une application donnée. Elle sera poursuivie par un approfondissement pratique de ces modèles, que nous avons choisi de conduire dans les trois grandes classes de représentation géométrique des modèles déformables : les modèles surfaciques, implicites et volumiques.

A travers cette démarche, l'objectif de notre travail thèse est d'obtenir une modélisation de haut niveau de données géométriques non organisées (nuages de points ou voxels) en déformant un modèle pour le faire coïncider avec ces données.

## Applications

Notre objectif de modélisation de données non organisées se décline en fonction des applications que nous envisageons. Nous avons choisi deux types d'applications particulières ; la reconstruction d'objets à partir de données de leur surface, et la segmentation d'images. Nous montrerons à travers notre travail comment les modèles déformables permettent une représentation pertinente de l'information. Nous avons également eu l'occasion d'employer les modèles de notre étude pour d'autres applications : création d'un modèle, correction des déformations de certaines modalités d'imagerie, et création d'images virtuelles.

Présentons plus précisément ces cinq classes d'application.

---

## Reconstruction

Nous appellerons reconstruction le problème qui se pose lorsque l'on a des données éparées sur la surface d'un objet, et que l'on désire retrouver la surface elle-même. Les applications se trouvent dans différents domaines. En vision par ordinateur, ils s'agit de modéliser une scène observée à partir d'une ou de plusieurs caméras. Des dispositifs de numérisation tridimensionnels permettent d'acquérir des points à la surface d'objets du monde réel. Nous avons par exemple utilisé un capteur actif composé d'une caméra vidéo et d'un plan laser pour numériser la surface d'un mannequin de 30 centimètres.

De plus, un ensemble de points épars n'est pas toujours suffisant pour visualiser un objet, ou pour en calculer des propriétés. Par exemple, si l'on désire appliquer des déformations correspondant à un modèle physique en utilisant la méthode des éléments finis, on a besoin d'un maillage de la surface de l'objet.

## Segmentation

Nous choisissons comme définition de la segmentation d'une image sa partition en autant de régions qu'il y a d'objets distincts dans l'image, les régions correspondant aux objets. La segmentation de structures médicales est importante à la fois pour établir un diagnostic médical et pour réaliser de façon optimale des interventions médicales assistées par ordinateur [Lavallée et al. 1997]. Les régions peuvent être extraites directement, ou bien être définies par leurs contours. Nous avons principalement inscrit notre travail dans cette deuxième démarche, en construisant un modèle déformable fondé sur des caractéristiques de contours extraites des images.

## Création d'un modèle déformable

Certains modèles déformables se déforment sans contrainte particulière de forme, guidés par leurs caractéristiques de bas niveau. D'autres modèles au contraire ont une forme propre, et toutes leurs déformations s'effectuent relativement à leur forme de base. On peut ainsi citer les modèles qui se déforment en déplaçant des points à partir d'une position initiale, ces points étant attachés à leur position initiale par un ressort. De cette manière, la déformation est contrôlée, et la forme du modèle sert de référence.

Pour de tels modèles, la forme initiale est très importante et il est nécessaire de la modéliser avec soin. On peut pour cela partir d'un modèle déformable dont la forme initiale est un objet géométrique simple comme une sphère ou un tore. Ainsi la forme obtenue par un premier modèle déformable sert de forme de référence à un deuxième modèle déformable.

## Correction d'images

L'IRM possède les nombreux avantages d'un examen non invasif et peu traumatisant, ainsi qu'un bon contraste entre les tissus mous. Mais ce type d'images présente des distorsions géométriques gênantes, qui proviennent essentiellement de l'hétérogénéité du champ magnétique principal, du fait des différences de susceptibilité magnétique des différents tissus [Sumanaweera et al. 1993]. Ces différences de susceptibilité se traduisent par le déplacement des interfaces entre tissus, principalement aux frontières air/tissus. La correction de ces distorsions par déformation d'un modèle TDM supposé sans déformation

---

permet de s'assurer qu'une planification chirurgicale établie à partir des images d'IRM est valide.

## Réalité virtuelle

Notre modèle déformable a également été employé dans une étude dont le but est la construction d'un simulateur d'examens échographiques. L'échographie est une modalité d'imagerie dynamique très utilisée, qui occupe une place privilégiée dans l'élaboration de nombreux diagnostics, notamment dans l'exploration non invasive des vaisseaux. La création d'un simulateur permettra à terme d'entraîner les échographistes en formation sur des cas qui pourront être répertoriés en fonction de la difficulté d'interprétation des images. L'étude de Delphine Henry [Henry 1997] a porté plus particulièrement sur la détection de thromboses veineuses profondes des membres inférieurs.

## Organisation du document

Ce mémoire est divisé en quatre parties, numérotées en chiffres romains.

- La partie I propose une modélisation des modèles déformables, et une étude des modèles de la littérature selon cet éclairage particulier. Nous nous sommes particulièrement intéressés à comparer les représentations géométriques et les déformations.
- La partie II relate notre utilisation d'un modèle surfacique déformable, et notre mise en œuvre d'outils d'interactivité.
- La partie III présente notre approche des modèles déformables implicites générés par des primitives, pour laquelle notre étude nous a amené à utiliser un squelette de l'objet pour créer le modèle déformable.
- Dans la dernière partie, IV, nous décrivons un modèle volumique déformable qui contient des méthodes élaborées de calcul de distance et de déformation.

Les chapitres sont désignés par le chiffre de la partie dans laquelle ils se trouvent, associé avec un numéro d'ordre au sein de la partie. Par exemple I●1 désigne le premier chapitre de la première partie. Les chapitres sont divisés en sections, qui sont numérotées en reprenant le numéro de la partie et du chapitre (par exemple I●1.1). Les subdivisions suivantes ne reprennent pas le numéro de la partie (par exemple 1.1.1).

L'organisation des chapitres suit notre modélisation du processus de création et d'utilisation de modèles déformables. Nous présentons cette modélisation dans la partie suivante.

Première partie  
Etude bibliographique



Nous allons étudier particulièrement les modèles déformables qui nous paraissent utiles pour atteindre notre objectif. Nous ne prétendons pas à une étude exhaustive. Notre désir est d'examiner les propriétés des modèles selon les critères de notre modélisation. L'originalité de notre démarche est de proposer ces différentes composantes qui peuvent être étudiées de manière indépendante, et dont les combinaisons pourront donner des idées de nouveaux modèles déformables. Des revues plus conventionnelles se trouvent dans la littérature, comme celle de [McInerney et Terzopoulos 1996].

### I●1.1 Modélisation des modèles déformables

Nous présentons dans cette section les cinq composantes qui à notre sens représentent les différents aspects des modèles déformables. Les notions propres à l'animation et au modelage par ordinateur se retrouvent dans [Wilhelms 1987] et [Bainville 1996].

Le modèle déformable est paramétré par un vecteur  $\mathbf{e}$  de paramètres, lequel est formé par l'union des paramètres des différentes composantes. Ce vecteur est appelé **état** du modèle déformable.

#### 1.1.1 Caractéristiques de liaison modèle-données

Les caractéristiques de liaison sont constituées d'une part d'éléments de bas niveau et d'autre part d'une fonction de liaison. Les caractéristiques de bas niveau sont les éléments qui vont induire l'évolution du modèle dans l'image. Cette composante n'est habituellement pas identifiée comme telle, alors que le choix de ces caractéristiques en fonction du type d'images et de l'application envisagée peut être déterminant.

Prenons l'image d'un filet de pêcheur lesté de poids. Les poids qui entraînent les mailles du filet dans l'eau (Figure I●1.1 (a)) sont les caractéristiques de bas niveau qui induisent la déformation d'un modèle déformable.

La fonction de liaison, quant à elle, établit le lien entre le modèle et les données. Elle est calculée à partir des caractéristiques de bas niveau, et peut par exemple entrer dans le calcul d'une énergie. La minimisation de cette énergie (cf. 1.1.5) mène alors à la correspondance entre le modèle et les données.

En poursuivant l'image du filet, la fonction de liaison serait composée à partir de la distance de chacun des poids au sol. Le filet suivra la forme du sol (correspondance entre le modèle et les données) lorsque toutes les distances seront nulles (Figure I●1.1 (b)).



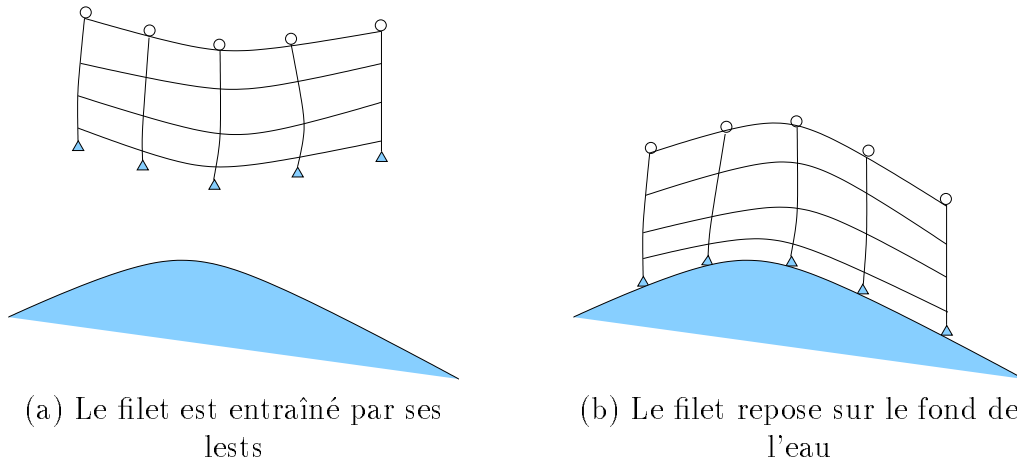


FIG. I.1.1: Image du filet de pêcheur

Nous verrons dans le chapitre I.2 que la répartition des caractéristiques de liaison entre le modèle et les données varie suivant la deuxième composante des modèles déformables : le type de représentation géométrique.

### 1.1.2 Représentation géométrique

Il s'agit des éléments géométriques constitutifs de la forme du modèle. Par exemple la surface du modèle ou bien la représentation de son volume intérieur. La représentation géométrique est dite continue si elle est connue de manière analytique, et discrète dans le cas contraire. On a alors un maillage, qui est généralement obtenu par discrétisation par différences finies d'une surface ou d'un volume. L'information est alors portée par les seuls nœuds de ce maillage, ce qui réduit la complexité algorithmique. La représentation géométrique est dans tous les cas déterminée par un nombre fini de paramètres réels.

### 1.1.3 Déformation

La déformation peut être liée à la représentation géométrique du modèle ou lui être indépendante. Dans le premier cas nous parlerons de **déformation directe** et dans le deuxième de **déformation indirecte**. La déformation est paramétrée par un nombre fini de variables. La figure I.1.2 (a) montre comment les vecteurs qui matérialisent la déformation sont portés par les points de la représentation géométrique, alors qu'en (b) ils sont portés par un maillage qui en est indépendant.

La déformation indirecte est une application  $\mathbf{T}$ , paramétrée par un vecteur de paramètres  $\mathbf{p}$  qui associe à tout point du modèle dans une certaine configuration sa position courante dans l'espace. Cette configuration donnée, que nous appellerons **configuration de base** peut être la configuration précédente ou une configuration de référence. Nous appellerons  $\text{Ref}_{\text{modèle}}$  le système de coordonnées du modèle dans cette configuration. Le modèle déformé se trouve dans l'espace de l'image à traiter, que nous appellerons **image des données**, avec son système de coordonnées  $\text{Ref}_{\text{données}}$ . La déformation indirecte relie donc  $\text{Ref}_{\text{données}}$  à  $\text{Ref}_{\text{modèle}}$ .

La déformation indirecte doit être injective, préserver l'orientation de l'espace, et être suffisamment régulière pour que les grandeurs qui en sont dérivées soient correctement

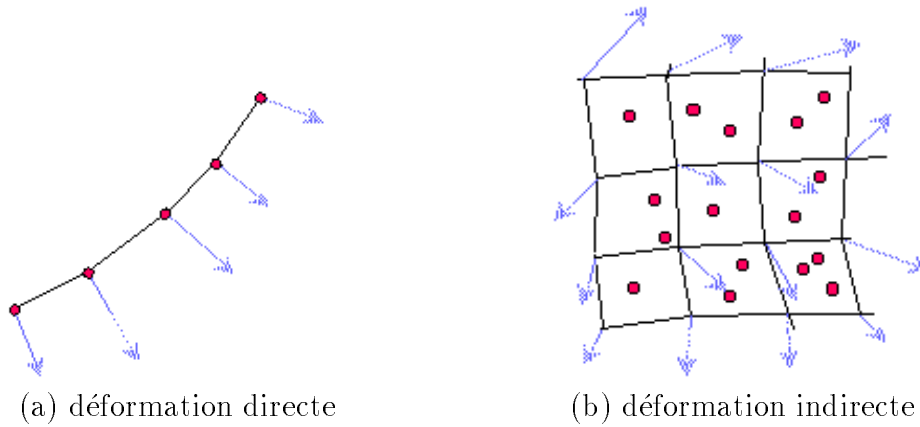


FIG. I.1.2: illustration de la relation déformation/représentation géométrique

définies.

### 1.1.4 Déformabilité

Nous appelons déformabilité l'ensemble des contraintes qui permettent de restreindre le domaine de déformation afin de s'assurer que la forme du modèle soit admissible.

Nous avons imaginé ajouter des segments rigides au filet du pêcheur, pour empêcher qu'il ne s'emmêle (Figure I.1.3).

La déformabilité est une manière d'utiliser de la connaissance a priori sur l'objet à traiter. Un modèle de ballon de baudruche, par exemple, devra maintenir son volume constant au cours de ses déformations.

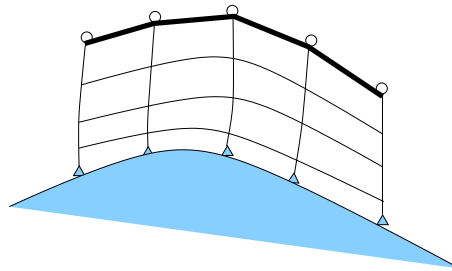


FIG. I.1.3: Le filet de pêcheur est muni de segments rigides

### 1.1.5 Contrôle

Le contrôle fixe l'évolution de l'état du modèle déformable, en tenant compte des contraintes. Il peut s'agir d'optimisation sous contraintes, ou bien d'intégration des équations différentielles du comportement.

L'évolution des lests du filet de pêcheur, par exemple, se fait selon une équation qui prend en compte la gravitation, l'interaction avec l'eau, et l'inertie du filet. Cette équation peut être intégrée par une méthode classique comme Runge-Kutta, pour établir la position des poids à un temps donné, connaissant la position et la vitesse au temps précédent.

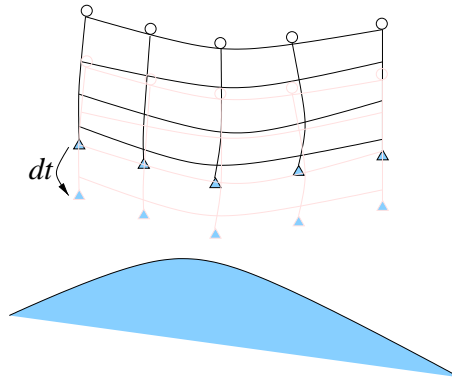


FIG. I•1.4: Le contrôle fixe l'évolution du filet de pêcheur

## I•1.2 Le formalisme des snakes

C'est le formalisme des *snakes* (serpents en anglais) qui est généralement reconnu à l'origine des développements des modèles déformables. Il faut cependant citer des travaux plus anciens, ceux de [Fischler et Elschlager 1973] sur les *templates déformables* et ceux de [Widrow 1973]. Kass, Witkin et Terzopoulos [Kass et al. 1988] ont introduit les snakes il y a dix ans, des courbes qui se déforment et se déplacent dans un champ potentiel afin de minimiser une énergie. L'idée a été largement reprise et développée [Terzopoulos et al. 1988], et les contours actifs sont devenus des surfaces actives, tout en gardant le principe de la déformation qui doit minimiser une énergie.

$$\mathcal{E}(\mathbf{e}) = \mathcal{E}_{interne}(\mathbf{p}) + \mathcal{E}_{externe}(\mathbf{p}) \quad (\text{I•1.1})$$

L'énergie est composée de deux termes. Le premier  $\mathcal{E}_{interne}(\mathbf{p})$  impose des contraintes à la forme du modèle, qui s'expriment par exemple en termes de rigidité, de tension, ou de régularisation de la déformation du modèle. Ces contraintes peuvent avoir des composantes dynamiques, comme une force de pression interne [Cohen et Cohen 1992]. On écrira

$$\mathcal{E}_{interne}(\mathbf{p}) = \mathcal{E}_{interne}^{forme}(\mathbf{e}) + \mathcal{E}_{interne}^{dynamique}(\mathbf{e}) \quad (\text{I•1.2})$$

Le deuxième terme d'énergie,  $\mathcal{E}_{externe}(\mathbf{p})$ , comprend d'une part les contraintes qui déforment le modèle en fonction des données sur lesquelles il s'applique, et d'autre part les contraintes particulières que l'utilisateur décide d'imposer.

$$\mathcal{E}_{externe}(\mathbf{p}) = \mathcal{E}_{externe}^{données}(\mathbf{e}) + \mathcal{E}_{externe}^{utilisateur}(\mathbf{e}) \quad (\text{I•1.3})$$

Les modèles snakes initiaux, ainsi que les multiples modèles de courbes déformables qui ont été utilisés pour le traitement d'images 2D avaient des caractéristiques communes. Ils devaient être initialisés au voisinage du modèle recherché, et pouvaient être ajustés interactivement. Leur emploi pour le traitement d'images 3D était relativement laborieux. Le modèle était propagé de coupe en coupe, et finalement, les contours pouvaient éventuellement être connectés pour construire une surface. Les surfaces actives sont l'évolution dans un espace 3D des contours actifs 2D. Elles permettent de traiter une image volumique directement, ce qui permet de tenir compte des trois composantes spatiales de la fonction d'intensité, et non plus seulement de deux.

---

Nous utiliserons cette propriété dans nos modèles, mais tout d'abord, nous présentons la chaîne de traitement des images qui intègre ce formalisme énergétique.

### **I•1.3 Modèles déformables et recalage non-rigide**

Nous présentons dans ce document sous l'angle des modèles déformables un paradigme général, dans lequel nous englobons le recalage non-rigide.

Le recalage non-rigide est la mise en correspondance de deux images qui contiennent des données similaires, mais non semblables, même à une transformation rigide près. L'application d'une déformation non-rigide à l'une des deux images la fait correspondre à l'autre image.

C'est dans ce cadre que nous avons choisi d'assimiler l'image déformée à un modèle déformable volumique, et de ne pas utiliser de terminologie différente pour le recalage non-rigide.

### **I•1.4 Chaîne de traitement de l'information**

Nous avons modélisé la chaîne de traitement des données comme illustré par la figure I•1.5. Les données 3D subissent un pré-traitement pour en extraire des caractéristiques particulières. Le modèle est déformé, puis la distance entre le modèle déformé et les données est calculée, et on optimise une énergie composée de deux termes, l'énergie interne qui dépend du modèle déformé, et l'énergie externe, qui dépend de la distance entre le modèle et les données.

Nous allons détailler dans les chapitres suivants les composants de cette chaîne.

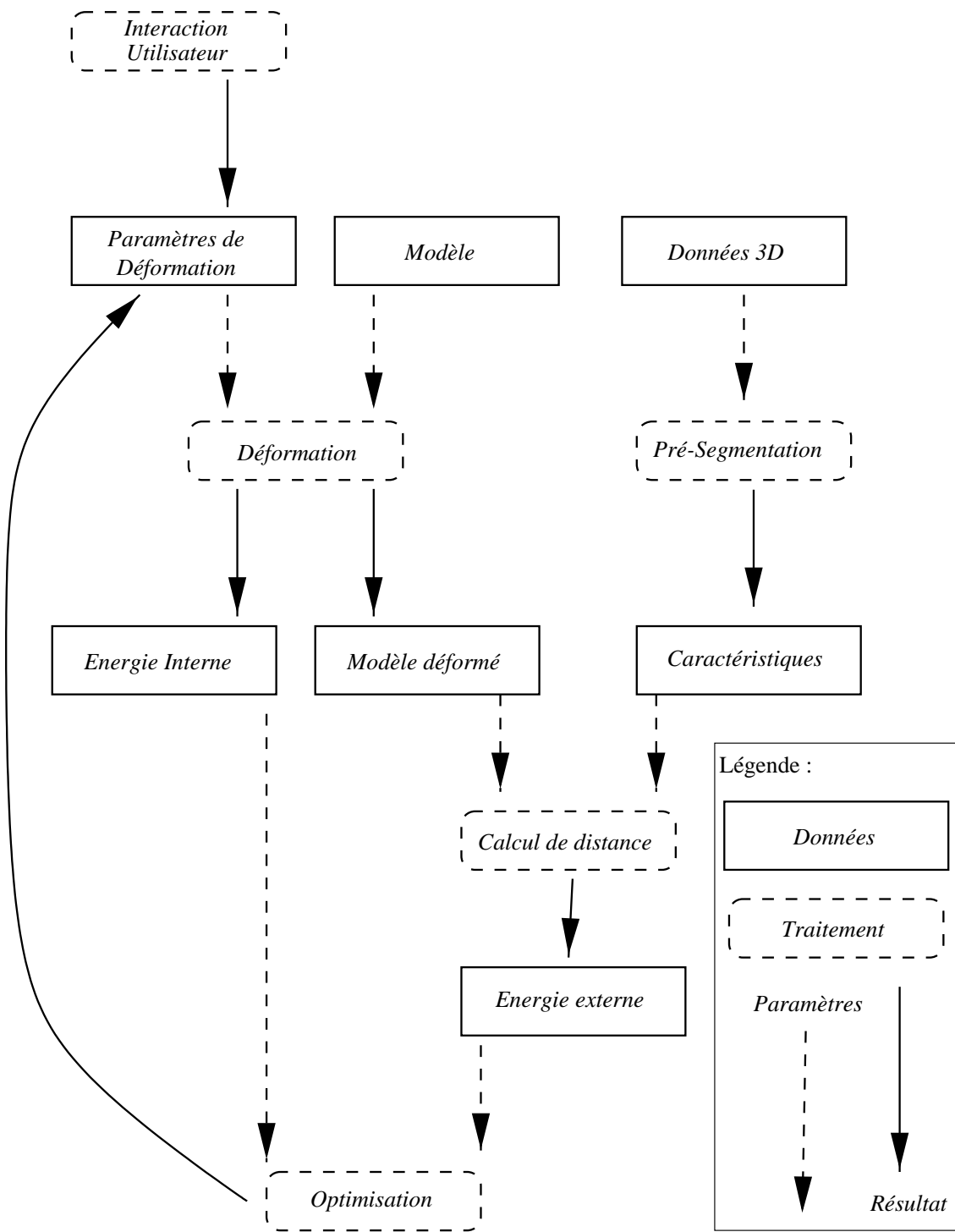


FIG. I.1.5: Traitement de l'information

# Chapitre I•2

## Caractéristiques de liaison

Comment établir le calcul de l'énergie externe, en sachant qu'elle doit décroître lorsque le modèle se rapproche des données ? L'énergie externe est calculée à partir de la fonction de liaison entre le modèle et les données qui, elle-même, utilise les caractéristiques de bas niveau. Nous allons nous employer dans ce chapitre à détailler cette affirmation très générale.

### I•2.1 Caractéristiques de bas niveau

La plupart du temps, on applique aux images un pré-traitement qui s'apparente à un filtrage. Son but est d'extraire des informations de bas-niveau, afin de réduire l'énorme quantité d'information présente dans les images volumiques pour n'en garder que l'essentiel. Ces informations, que l'on appelle **caractéristiques de bas niveau**, sont les éléments qui vont induire la déformation du modèle. Selon le type de représentation géométrique, elles seront calculées sur le modèle, les données, ou sur les deux. Nous réservons la discussion de ces trois cas à la section I•3.4, car les différents types de caractéristiques de bas niveau que nous allons maintenant présenter ne sont pas des types spécifiques du modèle ou des données.

Le choix d'une ou plusieurs caractéristiques dépendra de la modalité d'imagerie, de la qualité des images, et de l'application que l'on envisage.

#### 2.1.1 Types de caractéristiques

Certains auteurs désignent interactivement des points caractéristiques dans le modèle et l'image [Evans et al. 1989], [Bookstein et Green 1992] ou bien ceux-ci sont extraits semi-automatiquement [Rohr et al. 1996]. Mais la plupart du temps des méthodes automatiques sont appliquées, pour réduire l'intervention de l'utilisateur.

Une manière classique de sélectionner automatiquement l'essentiel est d'extraire des images les **contours** des objets, ou les **régions** qu'ils composent. D'autres approches exploitent d'autres **structures remarquables** des images. Nous allons détailler ces notions dans cette section, présentons-les d'abord rapidement.

Les contours sont généralement détectés comme des variations d'intensité dans les images, soit comme des extrema locaux de la norme de la dérivée première (gradient) soit comme des passages à zéro du laplacien. Les points détectés par ces deux opérateurs peuvent ne pas coïncider, car les maxima du gradient ne sont pas forcément des zéros du laplacien, même pour des fonctions  $C^\infty$ . D'autre part, les filtres utilisés en pratique sont

---

des approximations discrètes des fonctions gradient et laplacien.

Les régions sont extraites en utilisant un critère d'homogénéité ou de texture.

Parmi les autres structures remarquables des images, on compte les lignes de crête, les points extrémaux et les crêtes du relief. Les lignes de crête sont les lignes saillantes des surfaces d'iso-intensité. Les points extrémaux sont un sous ensemble des lignes de crête. Les crêtes du relief [van den Elsen et al. 1995], qui sont souvent confondues avec les lignes de crête, n'ont pas d'interprétation géométrique aussi simple que ces dernières.

Ces approches ont en commun de nécessiter un lissage préalable de l'image. On utilise pour cela le plus souvent une convolution avec les dérivées d'une gaussienne dont la taille  $\sigma$  détermine l'échelle à laquelle les caractéristiques sont obtenues, et a donc une grande influence sur le résultat de la pré-segmentation.

### 2.1.1.1 Valeurs des voxels

Certaines approches, ne réduisent pas la quantité d'information, et utilisent les valeurs de tous les voxels. Pour diminuer quand même le temps de calcul, elles appliquent généralement le cadre du multi-échelles (voir 2.1.2).

### 2.1.1.2 Contours

**Gradient** De nombreux auteurs caractérisent les contours par le maximum de la norme du gradient dans la direction du gradient. L'extremum est caractérisé par un passage par zéro de la dérivée directionnelle de la norme du gradient :  $\nabla(\|\nabla f\| \cdot \nabla f) = 0$ .

Pour traiter un volume  $3D$ , le calcul de gradient était opéré en  $2D$  sur chacune des coupes successives des données, mais l'information sur le troisième axe n'était pas utilisée. Le traitement s'opère actuellement en  $3D$ , mais cela pose cependant des problèmes si le volume n'est pas isotrope. Il faut alors interpoler les données, et si cela n'est pas fait correctement, l'information sur le troisième axe perd son sens.

Nous avons choisi une approche classique de la segmentation par contour qui est l'utilisation du filtre séparable optimal de Canny-Deriche-Monga [Canny 1986 ; Deriche 1987, 1990 ; Horaud et Monga 1993], qui est optimisé pour détecter les variations d'intensité de type marche d'escalier.

**Laplacien** Le calcul du laplacien de l'image s'obtient également à l'aide d'un filtre séparable [Deriche 1990]. Du fait de l'utilisation d'une approximation discrète du laplacien, ce ne sont pas les passages à zéro, mais les changements de signe du laplacien qui sont détectés.

**Etape de nettoyage** Les contours ainsi extraits doivent encore être nettoyés pour éliminer les points parasites, qui correspondent uniquement au bruit de l'image. On pourra par exemple extraire les maxima locaux dans la direction du gradient, et effectuer ensuite un seuillage par hystérésis. Les points des contours pourront aussi être chaînés, pour éliminer les petits contours, ou combler de petits trous. Les outils de morphologie mathématique permettent également de nettoyer les contours obtenus. Une opération d'érosion/dilatation sur l'image, par exemple permet d'éliminer des contours parasites.

### 2.1.1.3 Régions

La segmentation de l'image en différentes zones permet un étiquetage des voxels. De nombreuses méthodes de classification sont utilisées pour segmenter totalement l'image. Elles reposent sur des théories statistiques comme la théorie bayésienne de la décision. Le contexte spatial des voxels peut être modélisé d'une manière locale dans un cadre markovien [Aurdal et al. 1995 ; Géraud et al. 1995]. La théorie des ensembles flous est aussi utilisée [Bezdek et al. 1993 ; Boujemaa et al. 1994], ainsi que la théorie des croyances de Dempster-Shafer [Bloch 1996]. Nous considérerons ces méthodes comme des moyens d'extraire des caractéristiques pour les modèles déformables. [Dellepiane et al. 1992] ont été les pionniers de ce genre d'approche.

### 2.1.1.4 Autres structures remarquables

**Lignes de crête** Historiquement, cette notion a été définie d'après des observations de paysages, en repérant la ligne de crête des montagnes et le chemin du fond de la vallée (*Thalweg*). Ces deux exemples sont des lignes de crête, au sens mathématique du terme. La formalisation mathématique de cette notion fait intervenir les extrema locaux de la courbure. Les lignes de crête sont constituées des extrema locaux de la courbure principale maximale en valeur absolue, dans la direction principale correspondante. Elles peuvent être extraites par l'algorithme des "marching lines" [Thirion et al. 1992].

**Points extrémaux** Thirion extrait des points spécifiques des lignes de crête, les points extrémaux [Thirion 1994]. Ce sont les points extrémaux pour les deux courbures principales. Le fait de réduire l'information pour obtenir des points à la place de lignes permet un traitement plus rapide des données, mais nécessite que les images soient de bonne qualité, afin d'assurer la fiabilité du calcul des courbures.

**Crêtes du relief** Elles nécessitent de considérer l'image comme une carte d'élévation  $i = f(x, y, z)$ , une hyper-surface de  $\mathcal{R}^4$ . Mais dans ce formalisme, la dimension d'intensité joue un rôle particulier, et une simple transformation linéaire de l'intensité va déplacer les points extrémaux dans l'espace. Pour des images 3D, les crêtes du relief sont des morceaux de surface 3D, et non plus des lignes. L'information est donc moins réduite, ce qui diminue l'intérêt des crêtes du relief. De plus, une étude de [Maintz et al. 1996] comparant différentes caractéristiques pour recaler par corrélation des images scanner et IRM a obtenu de meilleurs résultats avec la norme du gradient qu'avec les crêtes du relief.

## 2.1.2 Extraction multi-échelles

Une image multi-échelles est un ensemble d'images construit par applications successives de filtres passe-bas sur une image. Cette structure peut-être représentée comme une pyramide d'images, avec l'image initiale au-dessous, et les images lissées aux niveaux suivants. Un choix habituel est d'utiliser un filtre gaussien, et de diminuer la résolution suivant chaque dimension de l'image d'un facteur de deux en montant d'un niveau [Jolion et Rosenfeld 1991]. Les caractéristiques sont alors calculées pour chaque niveau de la pyramide. Si l'on prend l'exemple du calcul du gradient, à un niveau élevé, on obtient les points de fort gradient, mais avec une grande incertitude, et à un niveau plus



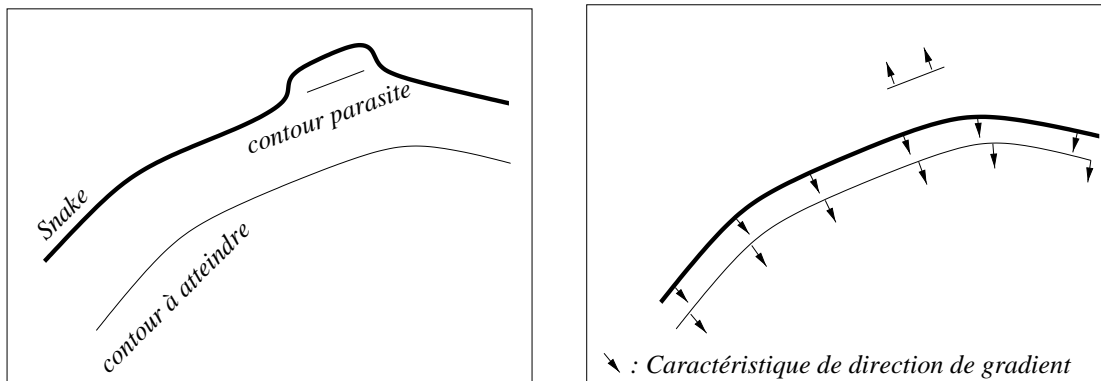
fin on a une meilleure précision spatiale, mais de nombreux petits contours parasites. La déformation s'opère en considérant le niveau le plus élevé - qui contient le moins de voxels - et en descendant dans la pyramide d'images au fur et à mesure que plus de précision est nécessaire.

Une autre façon de procéder est de considérer que le lissage confère une dimension supplémentaire à l'image, qui détermine l'échelle d'analyse, au lieu de fixer certaines valeurs arbitraires pour le lissage comme dans la pyramide classique. Ainsi [Fidrich et Thirion 1994] convoluent leurs images avec une gaussienne de paramètre  $\sigma$ , et étudient les lignes de crête et les points extrémaux en tenant compte de cette nouvelle dimension  $\sigma$ .

### 2.1.3 Associations de caractéristiques

L'extraction de points de contours n'est pas suffisante pour guider la déformation d'un modèle dans l'objet. Prenons l'exemple d'un contour actif qui est attiré par les points de gradient fort dans une image 2D. Son évolution peut être gênée par la présence de contours parasites, comme l'illustre la figure I●2.1. Si au contraire les contours contiennent une information supplémentaire, comme la direction du gradient, par exemple, et le modèle lui aussi possède cette information, il sera plus sélectif dans son évolution, et pourra éviter des contours qui ne correspondront pas aux caractéristiques qu'il porte. Nous appellerons un modèle portant de l'information sur les caractéristiques un **modèle enrichi**.

Nous présentons dans cette section différentes caractéristiques qui peuvent enrichir les points de contours.



(a) Le snake est stoppé par un contour parasite

(b) Le snake enrichi atteint son but

FIG. I●2.1: Importance de l'association de caractéristiques

#### 2.1.3.1 Direction du gradient

C'est une caractéristique dont la pertinence s'illustre bien dans la situation de faire la différence entre le bord intérieur et le bord extérieur d'une paroi épaisse, ou plus généralement entre deux contours proches dont les gradients ont des directions différentes.

Il faut noter que si la structure qui porte cette caractéristique se déforme, la direction du gradient doit être recalculée.

### 2.1.3.2 Norme du gradient

La norme est utile pour discriminer les contours les plus marqués des artefacts du filtrage.

### 2.1.3.3 Région et contours

Les informations de type région et contours peuvent aussi être combinées, pour une meilleure segmentation. [Cohen et al. 1991] par exemple interdisent le regroupement de régions adjacentes si le contour qui les sépare est très marqué.

### 2.1.3.4 Courbure

La courbure s'extrait à l'aide d'un calcul qui fait intervenir les dérivées partielles d'ordre deux de l'image. Ce calcul donne deux valeurs. Si l'on a choisi d'utiliser la courbure principale, il s'agit alors de la distinguer de la courbure secondaire. Un autre choix est de ne considérer que la moyenne des deux courbures.

A l'instar de la direction du gradient, la courbure doit être recalculée si elle est portée par une structure qui se déforme.

### 2.1.3.5 Crédibilité interne

[Hamadeh 1997] a introduit la notion de crédibilité interne, sur une idée de Brunie. Cette notion distingue les pixels qui sont les plus susceptibles d'appartenir à une structure anatomique d'intérêt. Pour cela, il sélectionne les pixels qui sont à la fois un extremum local du gradient, et un point de passage par zéro du laplacien. Il supprime les pixels dont la norme du gradient est inférieure à un seuil bas, et regroupe les pixels restants en composantes connexes. Les composantes connexes ne contenant aucun pixel dont la norme du gradient est supérieure à un seuil haut sont également supprimées. La crédibilité interne  $c$  est non nulle pour les pixels restants. Elle est définie comme une pondération de la norme du gradient,  $\|\mathbf{G}\|$ , et de la longueur de la composante connexe à laquelle appartient le pixel,  $l$ . Cette longueur peut être définie en fonction du nombre de pixels (en 2D) ou de voxels (en 3D) de la composante.

$$c = \alpha\|\mathbf{G}\| + (1 - \alpha)l, \quad \alpha \in [0, 1] \quad (\text{I}\bullet\text{2.1})$$

Ali Hamadeh a choisi de donner plus d'importance à la valeur de la norme du gradient qu'à la longueur de la composante connexe, il prend  $\alpha = 0.8$ .

### 2.1.3.6 Profil des niveaux de gris

Pour mieux caractériser la fonction de niveaux de gris aux points de contour, on peut calculer le profil de niveaux de gris dans une direction particulière, qui peut être par exemple celle du gradient. Ce profil est composé d'un certain nombre de valeurs, centrées sur le point. [Cootes et al. 1994] échantillonnent la dérivée des niveaux de gris, et normalisent les valeurs obtenues, afin d'obtenir une invariance par rapport à la mise à l'échelle uniforme des niveaux de gris, et à l'addition d'une constante.

### 2.1.3.7 Gradient et courbure

Le gradient et la courbure sont également utilisés, on les trouve même combinés, sous la forme du produit de la norme du gradient par la courbure moyenne [Collins et al. 1996], ou pour former une distance généralisée [Feldmar 1995], ou encore par combinaison d'une énergie de courbure et d'une énergie de gradient [Benayoun 1994]. Dans cette dernière méthode, les coefficients de la combinaison varient au cours du temps pour que la courbure soit initialement prépondérante, et que le gradient ait plus d'importance ensuite.

### 2.1.4 Discussion

Le choix d'une caractéristique dépend des images que l'on veut traiter et de l'application envisagée. Si l'image a une bonne résolution et une bonne qualité, les caractéristiques différentielles comme le gradient, la courbure ou les lignes de crête pourront être calculées avec une précision suffisante pour être utilisées. Au contraire si la résolution de l'image est faible, ces caractéristiques seront bruitées, et donc difficiles à exploiter. C'est également le cas pour des images de mauvaise qualité comme les images échographiques.

Pour une application de segmentation, toutes les caractéristiques présentées, ainsi que leurs combinaisons sont envisageables. Si l'on cherche à reconstruire la surface d'un objet, le choix des caractéristiques sera limité à celles qu'il est possible de calculer. En effet, si les données se présentent sous la forme d'un nuage de points et non pas sous la forme d'une image volumique, aucune des caractéristiques qui font intervenir les valeurs des voxels ne peuvent être calculées.

Plus l'ordre de la dérivation à appliquer sur l'image pour calculer la caractéristique est important, plus cela nécessite de temps. Si la rapidité du traitement est un critère important, il sera alors sage d'évaluer ce qu'apporte l'utilisation d'une caractéristique par rapport au temps requis pour son calcul.

C'est ainsi que nous avons raisonné en ce qui concerne nos applications. Nous avons choisi des caractéristiques ponctuelles pour la reconstruction, étant donné que les données sont sous forme de nuages de points. Pour la segmentation, nous avons cherché un type de caractéristique qui soit rapide à calculer, et qui permette de discriminer dans une image un objet de ses voisins proches. C'est pourquoi nous avons retenu la combinaison des points de contour et du vecteur gradient de ces points.

Il faut prendre en compte l'utilisation qui va être faite des caractéristiques retenues. Si elles sont utilisées par une fonction de liaison, la dimension des caractéristiques influera sur le temps de calcul de cette fonction.

## I●2.2 Fonction de liaison

L'énergie externe du modèle le relie aux données à traiter. Elle peut dériver d'un champ de potentiel, ou bien être déterminée par une fonction de similarité, ou une fonction de distance.

### 2.2.1 Potentiel

Une solution adaptée à la segmentation d'images simples est de ne pas calculer de caractéristiques particulières pour le modèle, et de construire uniquement un champ po-

---

tentiel à partir de l'image à traiter. Le modèle évoluera vers un minimum de ce champ. Les premiers modèles de contours actifs évoluaient ainsi vers les pixels de gradient fort de l'image. Cette solution n'utilise que l'information contenue dans l'image à traiter, et n'ajoute pas d'information a priori dans le modèle.

Cette solution est également adaptée pour la reconstruction d'un objet à partir de points de cette surface.

## 2.2.2 Fonction de similarité

Lorsque la valeur des voxels est utilisée comme caractéristique de bas niveau (voir 2.1.1.1), c'est la similarité entre régions ou entre voxels, sous forme d'information mutuelle [Studholme et al. 1996 ; Kim et al. 1996] ou de corrélation [Bajcsy et Kovacic 1989], qui est à la base du calcul de l'énergie externe [Viola et Wells 1995 ; Wells et al. 1995, 1996].

## 2.2.3 Calcul de la distance généralisée

Nous nous plaçons à présent dans le cas où l'on a choisi de faire porter au modèle des caractéristiques de même nature que celles que l'on extrait sur l'image à traiter, appelée image des données,  $D$ . La fonction de liaison est alors la distance entre les caractéristiques du modèle et celles des données. Pour chaque point de  $D$ , il faut calculer à chaque itération la distance au modèle. C'est la distance au point le plus proche de  $D$  parmi les  $n$  points du modèle. A chaque point de coordonnées  $x, y, z$  sont associées ses caractéristiques de bas niveau. Chaque point a ainsi  $k$  composantes, on parle de point en  $k$ -d.

Nous allons présenter dans cette section les différents moyens de calculer la distance  $k$ -d à un ensemble de points.

Posons le problème de manière générale. Il s'agit d'étudier les moyens de calculer rapidement la distance d'un point  $k$ -d à un ensemble de points  $k$ -d. Nous allons passer en revue différentes approches de la littérature, que nous avons regroupées suivant les classes de solutions standard de l'algorithmique [Aho et al. 1994]. Celles-ci comprennent d'une part plusieurs méthodes permettant d'accélérer la recherche du point le plus proche dans un ensemble de points (diviser-pour-régner, programmation dynamique, méthode gloutonne, recherche conduite localement ou arbre de jeu), et d'autre part la détermination de solutions approchées du problème, qui peuvent être satisfaisantes, si l'on sait borner l'erreur ou garantir que l'approximation est compatible avec l'utilisation recherchée du résultat.

Ces méthodes sont caractérisées par trois facteurs : l'efficacité de la recherche, le coût du stockage de la structure de données utilisée, et la complexité de la phase d'apprentissage. Nous établirons un classement des différentes approches suivant ces critères.

### 2.2.3.1 Approches exactes

Les méthodes que nous exposons ici ont toutes en commun de calculer le point le plus proche de manière exacte. Pour ce faire, elles exploitent chacune un principe d'algorithmique différent. Nous allons les passer en revue.

**Diviser-pour-régner** Le principe est de diviser un problème de taille  $n$  en sous-problèmes plus petits, de manière que la solution de chaque sous-problème facilite la construction

de la solution du problème entier. Cette approche est d'autant plus efficace que les sous-problèmes sont de tailles sensiblement égales.

Une manière d'appliquer ce principe à notre problème est de partager le nuage de  $n$  points en deux nuages de taille moitié, de rechercher le point le plus proche de  $D$  dans chacun des deux nuages, et de garder le point le plus proche de  $D$  parmi ces deux points. Cet algorithme ne sera efficace que s'il existe un moyen d'éviter de chercher systématiquement le point le plus proche dans les deux sous nuages. Nous verrons une application de ce principe avec les arbres k-d.

**Programmation dynamique** Il faut prendre garde en morcelant un problème en de nombreux sous-problèmes de ne pas créer et recalculer plusieurs fois des sous-problèmes identiques. La programmation dynamique propose de construire une table des sous-problèmes rencontrés pour éviter des calculs redondants.

Nous reviendrons à cette idée lors du calcul du point le plus proche itéré.

**Algorithmes gloutons** Cette approche utilise une heuristique qui permet à chaque étape de l'algorithme de faire un choix localement optimal, selon certains critères. Cette stratégie de choix locaux successifs n'a aucune garantie d'optimalité globale, mais on l'utilise quand la probabilité d'obtenir une solution fiable est assez élevée, si le seul vrai moyen d'avoir une solution optimale est d'utiliser une technique de recherche exhaustive et prohibitive.

Nous n'avons pas trouvé d'auteurs utilisant cette approche, car lorsqu'il s'agit de calcul de distance, il est difficile de trouver un critère heuristique localement optimal.

**Arbres de jeu** Si l'on doit effectuer une recherche exhaustive pour trouver une solution optimale, on pourra utiliser la recherche avec retour arrière (backtracking) et l'élagage  $\alpha - \beta$  pour réduire l'espace de recherche.

Une recherche heuristique de type séparation-évaluation, ou branch-and-bound (aiguiller-et-borner) permet d'imposer un ensemble de contraintes pour réduire le parcours de l'arbre. En tout nœud auquel on aboutit, on essaye d'aiguiller (branch) la suite de la recherche en direction d'une ramification particulière de l'arbre, après avoir évalué un coût minorant, donc une borne locale (bound), et l'avoir comparée à la borne globale obtenue jusqu'à ce stade pour le problème.

Nous présenterons dans le chapitre IV•3 un algorithme de recherche dans un arbre k-d, qui utilise la méthode aiguiller-et-borner.

Fukunaga et Narendra [Fukunaga et Narendra 1975] ont proposé une autre approche dans laquelle, grâce à la méthode aiguiller-et-borner, ils réduisent considérablement le nombre de points examinés. Ils représentent l'espace Euclidien des points k-d par un arbre dont les nœuds sont des hypersphères. Chaque hypersphère est elle-même décomposée en petites hypersphères. Le processus de décomposition s'arrête lorsque l'hypersphère contient un nombre minimal de points. Chaque nœud de l'arbre est caractérisé par le centre de la sphère, son rayon et le nombre des fils. Les feuilles de l'arbre sont constituées d'un ou plusieurs points. Les centres des hypersphères sont obtenus par un algorithme des k-moyennes. Cet algorithme adopte le principe des recherches locales, que nous décrirons dans le paragraphe suivant.

Lockwook [Lockwook 1986] a proposé de remplacer les centres des hypersphères de la méthode de Fukunaga par des points k-d de l'ensemble de données. Cette modification

permet de gagner de la place au niveau du stockage de la structure de données, mais dégrade la performance de la recherche. Il y remédie par la suite [Lubiarz et Lockwook 1997].

Kim et Park [Kim et Park 1986] proposent en 1986 un algorithme qui effectue une recherche de type aiguiller-et-borner sur un partitionnement ordonné des points. Après avoir trié les points suivant leurs projections sur chacun des axes, ils créent une structure d'arbre qui permet une recherche en un temps constant.

### 2.2.3.2 Les solutions approchées

On trouve dans la littérature deux types de solutions approchées. Le premier calcule une carte de distance, le deuxième une carte du plus proche voisin. En effet, il y a une solution théorique exacte au problème du point le plus proche. Les  $n$  points définissent une partition de l'espace en  $n$  régions de Voronoi. Pour un k-d point quelconque dans l'espace, il suffit de connaître la région de Voronoi dans laquelle il se trouve pour connaître son point le plus proche. Mais cette région est difficilement calculable exactement en dimension supérieure à trois. C'est pourquoi une solution est de calculer une approximation de ces régions.

**Recherches locales** Dans certaines situations, la stratégie suivante permettra d'atteindre la solution optimale d'un problème : Commencer par une solution aléatoire. Appliquer à la solution courante un élément d'un ensemble de transformations donné, dans le but de l'améliorer. La version améliorée devient la nouvelle solution courante. Répéter jusqu'à ce qu'aucune amélioration de la solution courante ne soit plus possible.

Si les transformations considérées ne comprennent pas toutes les solutions remplaçant une solution par n'importe quelle autre, la solution optimale ne sera pas forcément trouvée. On parle alors de recherche locale.

Les algorithmes de recherche locale ont une efficacité maximum quand ils servent d'heuristique pour la solution de problèmes qui ne sont exactement solubles que moyennant une complexité exponentielle.

La partition d'un ensemble de  $n$  points en  $k$  sous-ensembles se fait classiquement par un algorithme des k-moyennes qui utilise une recherche locale.

**Carte de distance octree-spline** Au lieu de rechercher le point le plus proche de  $D$  dans un ensemble de  $n$  points, dans le but de calculer la distance à l'ensemble de points, et le gradient de cette distance, on peut calculer une approximation de la distance et du gradient. C'est la solution choisie par Brunie, Lavallée et Szeliski [Brunie 1992 ; Lavallée et al. 1996], qui créent une carte de distance sous la forme d'un octree-spline adapté au modèle. Ils pré-calculent la distance au modèle en chaque sommet des cubes de l'octree. Ensuite la distance d'un point quelconque au modèle est approchée par l'interpolation linéaire des distances des 8 sommets du cube auquel appartient le point. Cette structure est très efficace en termes de calcul de distance, puisque le coût ne dépend pas du nombre de points du modèle, mais uniquement de la profondeur de l'octree.

**Carte de distance discrète** La distance à un ensemble de points peut être calculée de manière discrète, grâce à la distance du chanfrein par exemple. Cette distance se calcule habituellement sur une grille régulière, et son stockage nécessite le stockage de cette grille,

---

ce qui est coûteux. Ce coût diminue en utilisant une distance du chanfrein adaptative (voir par exemple [Mangin 1995]).

**Le Voronoi-bucket** [Ramasubramanian et Paliwal 1992] insistent sur l'importance de l'optimisation de la construction de l'arbre k-d. En effet la complexité de la recherche dans l'arbre doit être minimale. Ils proposent de combiner un arbre k-d avec le Voronoi-bucket.

L'idée des auteurs est de construire un arbre k-d en utilisant l'information donnée par le diagramme de Voronoi. Les  $n$  vecteurs définissent une partition de l'espace en  $n$  régions distinctes appelées régions de Voronoi. Ces régions sont chacune associées à un point, elles représentent la partie de l'espace pour laquelle le point le plus proche est le point associé à la région. Une région de Voronoi  $V_j$  associée au point  $c_j$  est la région convexe définie par l'intersection des demi-espaces  $H(c_i, c_j), i = 1, \dots, n$  où  $H(c_j, c_i)$  est le demi-espace des points plus proches de  $c_j$  que de  $c_i$ .

Le Voronoi-bucket est donc un arbre dont les nœuds terminaux sont des compartiments (buckets) dans lesquels on range les points dont la région de Voronoi intersecte la région définie par le compartiment. La recherche dans une telle structure se fait en deux temps : la localisation du compartiment contenant le point dont on cherche le plus proche voisin, et le calcul de la distance aux points associés à ce compartiment.

Si les compartiments sont suffisamment petits, ils intersecteront peu de régions de Voronoi, et l'on aura par point peu de calculs de distances à effectuer pour trouver le point le plus proche. On comprend que le point délicat de cette approche est la localisation des hyperplans du diagramme de Voronoi.

## 2.2.4 Discussion

Le choix du type de fonction à retenir dépend des caractéristiques de bas niveau choisies. Si l'on choisit les mêmes caractéristiques dans le modèle et les données, l'emploi d'une fonction de distance s'impose. Elle doit être rapide à calculer et suffisamment précise pour ne pas introduire d'erreur d'ordre supérieure à celle sur les caractéristiques. On peut donc préconiser un choix de distance approchée de type octree-spline pour une distance 3D. Pour une distance en dimension supérieure à trois, les arbres k-d fournissent une bonne base algorithmique, qui nécessite cependant des aménagements pour autoriser une approximation de la distance. C'est ce que nous présenterons dans le chapitre IV•3.

Dans le chapitre suivant de cette partie, nous allons étudier la composante **représentation géométrique** des modèles déformables. Nous pourrions ensuite discuter du calcul des caractéristiques sur le modèle ou les données, en fonction du choix établi pour cette composante.

# Chapitre I•3

## Représentation géométrique

Les représentations géométriques des modèles que nous avons étudiés se répartissent en trois catégories : surfaciques, implicites et volumiques. Ces catégories fournissent une base pour le classement original des différents types de représentations géométriques que nous proposons.

### I•3.1 Modèles surfaciques

Les modèles surfaciques sont appelés ainsi car leur représentation géométrique est une surface. Nous les avons classés suivant la manière dont la surface est obtenue, en distinguant les modèles polygonaux, les modèles construits sur des bases de fonctions, et les modèles à base de particules.

#### 3.1.1 Modèles polygonaux

Ce sont des modèles discrets qui sont intéressants par leur simplicité géométrique, et leur souplesse de contrôle local de la forme.

[Miller et al. 1991] proposent un *modèle déformable géométriquement*, qui est construit à partir de facettes triangulaires. Le modèle cherche à minimiser une fonction de coût, qui est composée de trois termes : le premier résulte d'une force d'expansion, le deuxième est déterminé par l'image et le dernier maintient la topologie. C'est le formalisme classique des snakes. Le terme dépendant de l'image est très simple, il nécessite que l'intérieur et l'extérieur de l'objet à reconstruire soient définis. Un rééchantillonnage dans les régions en expansion permet d'assurer la cohérence du modèle pendant sa croissance.

[Bainville 1992] a proposé un modèle similaire appelé  $\delta$ -snake, composé de triangles réguliers. Son modèle gère non seulement l'accroissement de la surface, mais aussi sa diminution, en partageant en deux ou en supprimant des arêtes, pour qu'elles aient toutes une longueur proche du paramètre  $\delta$ . Le modèle du  $\delta$ -snake a été étendu [Lachaud et Bainville 1994] pour détecter les auto-intersections et les étranglements de sa surface, et modifier en conséquence sa topologie.

Le maillage simplexe est une représentation de surface introduite par [Delingette 1994], dans laquelle la propriété caractéristique est que le degré de chaque sommet est constant. Pour les  $\delta$ -snakes, le nombre de sommets de chaque facette est constant, et le nombre de voisins est libre. Pour les maillages simplexes, c'est l'inverse, chaque sommet a exactement trois voisins et le nombre de sommets d'une facette n'est pas fixé (voir un exemple figure I•3.1). La taille des facettes n'est pas imposée non plus, la surface peut ainsi être



représentée de manière adaptative, avec peu de sommets dans les zones de faible courbure. Les maillages simplexes servent ainsi par exemple pour décimer le maillage d'une isosurface [Delingette 1997]. Deux morceaux de surface simplexe sont aisément ajustables pour créer des surfaces de topologie complexe.

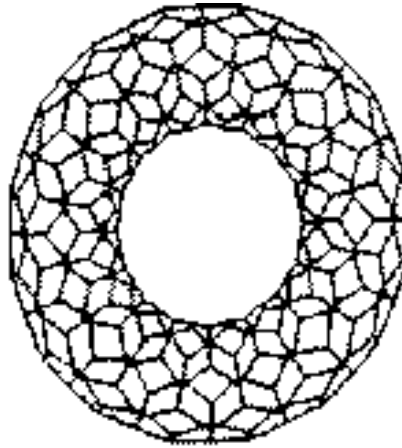


FIG. I●3.1: exemple de maillage simplexe, d'après [Delingette 1994]

Une approche est de créer des modèles fondés sur la physique, dans lesquels les sommets des polygones portent des masses. La déformation élastique d'une telle structure se fait en utilisant les équations de la dynamique. [Nastar et Ayache 1994] se servent d'un tel modèle pour comparer et classifier des déformations. Promayon [Promayon et al. 1996, 1997] propose un modèle dynamique qui permet d'intégrer des contraintes globales comme la conservation du volume.

Les *formes actives* de [Cootes et al. 1995] utilisent une représentation surfacique dans laquelle ce sont les sommets des facettes qui sont importants. En effet ce modèle utilise un moyen de déformation original que nous décrirons au chapitre I●4.

[DeCarlo et Metaxas 1994] ont eu l'idée de réaliser un modèle déformable par mélange de deux modèles. Ils interpolent deux formes paramétrisées pour créer un nouveau modèle. L'interpolation d'une sphère et un tore, par exemple permet de créer une nouvelle forme qui pourra changer de genre. Pour créer leurs formes de bases, ils utilisent des objets géométriques simples comme des sphères ou des cylindres, ou bien des superquadriques (voir 3.2.1).

### 3.1.2 Modèles construits sur des bases de fonctions

Ce sont des modèles continus qui sont décrits par leur paramétrisation sur un ensemble de fonctions. Plusieurs sortes de fonctions ont été employées, les plus utilisées étant les fonctions splines.

#### 3.1.2.1 Surfaces-splines

Plusieurs auteurs ont choisi une représentation de modèle surfacique à l'aide splines, comme Cohen [Cohen et Cohen 1993], McInerney et Terzopoulos [McInerney et Terzopoulos 1993], ou Leitner [Leitner 1993].

Une surface spline est obtenue par le raccordement continu de plusieurs morceaux de surfaces ou patches. Chaque patch est caractérisé par la position de points de contrôle, répartis sur un polygone. La donnée des points du maillage de contrôle, lequel peut être régulier ou non [Loop 1994], définit la surface spline. Le modèle de Leitner, qui utilise une spline adaptative et des patches à  $n$  côtés (figure I●3.2), permet de gérer les changements de topologie de la surface.

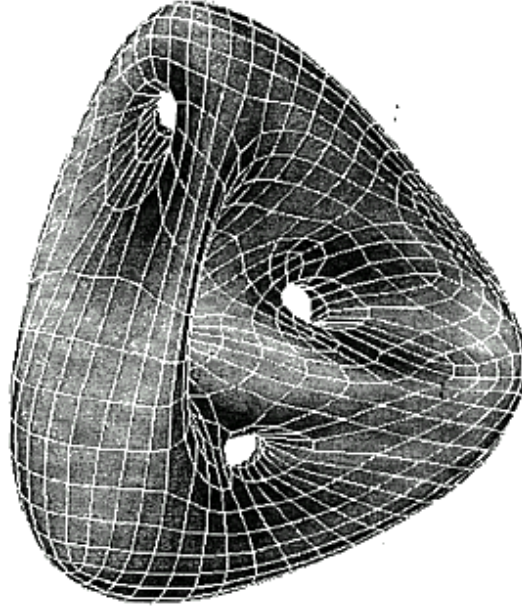


FIG. I●3.2: Exemple de surface de topologie complexe, d'après [Leitner 1993]

### 3.1.2.2 Surfaces de Fourier

Staib et Duncan [Staib et Duncan 1996] ont décrit un modèle qui utilise une paramétrisation de Fourier, qui décompose la surface en une somme pondérée de fonctions de base sinusoïdales. Ils modélisent quatre types de surfaces, des tores, des tubes, des surfaces ouvertes et des surfaces fermées. L'idée de la décomposition s'explique facilement en 2D, car une courbe fermée  $f(t)$  est périodique et peut s'exprimer dans la base :

$$B_{F,t} = 1, \cos(2\pi t), \sin(2\pi t), \cos(4\pi t), \sin(4\pi t), \dots$$

Pour une courbe ouverte, le domaine de définition sera parcouru deux fois, une fois dans le sens direct et une fois dans le sens indirect, afin d'obtenir une fonction paire et périodique, décomposable en série infinie dans la base :

$$B_{O,t} = 1, \cos(2\pi t), \cos(4\pi t), \cos(6\pi t), \dots$$

Pour une surface 3D, la base de décomposition dépend du type de la surface recherchée.

La série est tronquée, ce qui a comme effet de lisser la surface obtenue, et de limiter le nombre de paramètres nécessaires à la représentation.

### 3.1.2.3 Harmoniques sphériques

Ce sont les solutions de l'équation de Laplace exprimée dans un système de coordonnées sphériques. Elles constituent une base de fonctions orthogonales, orientées selon les fréquences spatiales, ce qui permet d'approximer une forme en choisissant le niveau de décomposition. Cette modélisation est limitée aux formes en étoile [Lefaix et al. 1997].

### 3.1.3 Systèmes à base de particules

Des techniques locales, comme dans les modèles à base de particules sont une manière de ne pas être limité par une seule topologie. Ces modèles sont des modèles discrets. Les particules se répartissent à la surface du modèle, peuvent porter des caractéristiques, et sont liées entre elles par des forces d'interaction. Szeliski reconstruit des objets en utilisant des particules [Szeliski et Tonnesen 1992] comportant une masse et un vecteur normal, qui interagissent selon un modèle physique. Lombardo [Lombardo et Puech 1995] a créé un modèle similaire, qui possède en plus une mémoire de forme. [Wu et al. 1995] utilisent les systèmes de particules pour la modélisation et l'animation.

## I•3.2 Modèles implicites

Ce sont des modèles dont la représentation géométrique est une surface définie par une équation implicite du type  $f(x, y, z) = iso$ . Nous les avons séparés des modèles surfaciques car il y a plus d'informations dans une équation implicite que dans une surface. En particulier les modèles implicites donnent la possibilité pour tous les points de l'espace de savoir s'ils sont ou non à l'intérieur du modèle. De plus la fonction  $f$  permet de calculer une approximation de la distance à la surface.

### 3.2.1 Superquadriques

Les superquadriques sont une extension des quadriques élémentaires qui sont utilisées comme représentation géométrique de modèles déformables car elles permettent de construire une surface déformable à l'aide d'un très petit nombre de paramètre [Sclaroff S. 1991 ; Terzopoulos et Metaxas 1991 ; Bardinet 1995]. Il y a quatre types de superquadriques, superellipsoïde, superhyperboloïde à une ou deux nappes, supertore. Le plus généralement, c'est le superellipsoïde qui est choisi (pour l'utilisation du supertore, voir [DeCarlo et Metaxas 1994]). La modélisation est alors limitée aux objets de topologie sphérique. L'équation implicite associée à un superellipsoïde s'écrit  $f(x, y, z) = 1$ , avec

$$f(x, y, z) = \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_1}{\epsilon_2}} + \left( \frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}$$

L'objet est ensuite approximé par une paramétrisation, qui répartit les points le plus régulièrement possible sur la surface [Bardinet 1995].

## 3.2.2 Surfaces implicites générées par des primitives

### 3.2.2.1 Présentation

Une surface implicite peut être définie directement par une équation analytique comme le sont les superquadriques, ou comme le mélange de plusieurs objets implicites, chacun étant généré par une primitive simple. Les surfaces implicites générées par des primitives ont été principalement développées comme des outils de modélisation de formes libres “free-form” [Wyvill et al. 1986 ; Wyvill et Wyvill 1989 ; Bloomenthal et Wyvill 1990 ; Bloomenthal et Shoemake 1991]. Elle connaissent un grand succès en modelage par ordinateur sous le nom de *metaballs*, que nous traduirons par métaboules. L'utilisation de primitives facilite la modélisation de la forme, et permet un stockage compact de formes complexes. L'ouvrage de [Bloomenthal 1997] propose un excellent tour d'horizon des différentes formes de surfaces implicites.

Une surface implicite  $S$  générée par un ensemble de primitives  $P_i (i = 1..n)$  auxquelles sont associées des fonctions potentielles  $f_i$  est définie par :

$$S = \{M \in \mathbb{R}^3 / f(M) = iso\} \text{ où } f(M) = \sum_{i=1}^n f_i(M)$$

- $iso$  est une valeur réelle correspondant à l'iso-valeur.
- Les primitives  $P_i$  peuvent être n'importe quelle primitive géométrique admettant une fonction de distance bien définie.
- Les fonctions potentielles  $f_i$  sont des fonctions monotones décroissantes qui dépendent de la distance à la primitive associée.

La surface délimite un volume défini par  $f(M) \geq iso$ , et les vecteurs normaux sont dirigés le long du gradient de  $f$ .

### 3.2.2.2 Fonctions potentielles

Elles peuvent être définies, par exemple par des fonctions exponentielles [Blinn 1982], ou être polynomiales par morceaux [Nishimura et al. 1985 ; Wyvill et al. 1986 ; Blanc et Schlick 1995], ou être des fonctions procédurales paramétrées [Kacic-Alesic et Wyvill 1991]. Elles peuvent même ne pas dépendre de manière isotrope de la distance à la primitive associée [Blanc et Schlick 1995].

Les exemples de surfaces implicites des figures I●3.4 et I●3.3 montrent comment l'utilisation de deux fonctions potentielles différentes influe sur la manière dont le mélange s'opère. Plus la décroissance de la fonction potentielle avec la distance est rapide (ce que nous appellerons une fonction potentielle raide), plus la surface générée par plusieurs primitives se rapproche de l'union des surfaces générées par les primitives considérées comme seules dans l'espace. Dans les deux figures, les surfaces implicites sont définies par deux primitives ponctuelles associées à des fonctions potentielles identiques. La pente de ces fonctions est dix fois supérieure dans la figure I●3.4 à celle de la figure I●3.3.

## 3.2.3 Utilisation des métaboules

Muraki [Muraki 1991] est le premier à avoir utilisé des surfaces implicites générées par des primitives pour reconstruire des formes. Ces surfaces ont l'avantage de modéliser les

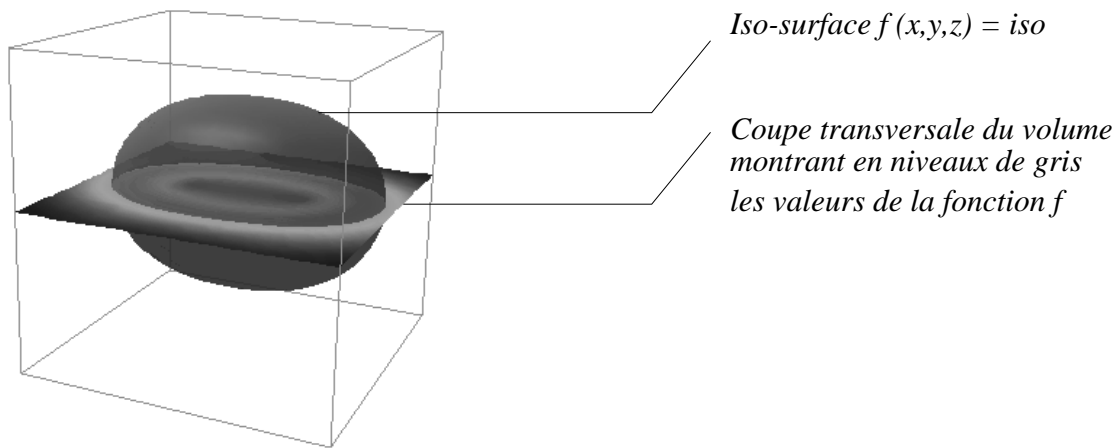


FIG. I●3.3: fonction potentielle douce

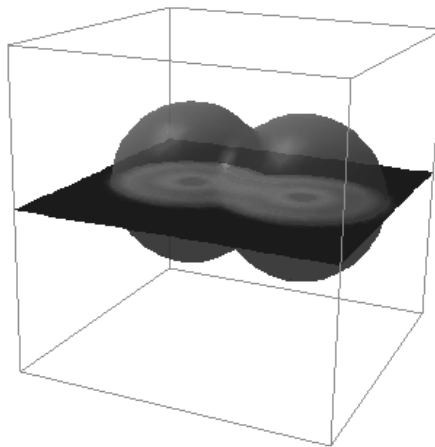


FIG. I●3.4: fonction potentielle raide

formes “libres” d’une manière très compacte, et de donner en même temps une “distance” à l’ensemble des points des données qui peut être utilisée pour la minimisation de l’énergie.

Muraki a présenté une méthode automatique pour générer une forme implicite à partir de données d’un capteur de distance, qui comprennent à la fois des points et les vecteurs normaux à la surface correspondants. Le modèle utilisé pour la reconstruction est le “Blobby Model” de Blinn [Blinn 1982]. Il est composé de primitives ponctuelles  $P_i = (x_i, y_i, z_i)$ , associées à des fonctions potentielles qui dépendent exponentiellement de la distance  $d(M, P_i)$  :

$$f_i(M) = b_i e^{-a_i d(M, P_i)}$$

les  $b_i$  peuvent être positifs ou négatifs. Une “primitive négative”, c’est-à-dire une primitive dont le potentiel est négatif permet de réduire le volume de l’objet implicite, et une primitive positive accroît ce volume.

Le principe de l’algorithme est de minimiser une énergie qui représente une “distance” entre la surface implicite courante et les points des données. La forme reconstruite est affinée de manière itérative en subdivisant une primitive en deux nouvelles primitives, et en optimisant les paramètres de ces dernières pour minimiser la fonction d’énergie.

Cependant, Muraki ne propose pas de méthode satisfaisante pour sélectionner la primi-

tive à subdiviser, ce qui conduit à des calculs très longs, empêchant d'utiliser cette méthode pour des objets détaillés. De plus, le choix d'une fonction exponentielle ne permet pas de traiter localement la question de la reconstruction. Chaque primitive a une influence sur toute la surface reconstruite, même sur les zones de la surface les plus éloignées de la primitive. L'ajout de nouvelles primitives modifie la forme entière, y compris les parties qui sont déjà correctement reconstruites. Enfin, Muraki propose d'initialiser la forme avec une seule primitive, placée au barycentre des points des données, ce qui n'est pas acceptable pour les objets de topologie complexe comme les objets comportant des trous.

Les metaboules sont aussi utilisées dans le domaine de l'animation, comme dans le système LEMAN [Turner 1995].

### 3.2.4 Approche par ensembles de niveau

Cette approche permet de modéliser des formes complexes, sans connaissance a priori de leur topologie. Elle a été proposée par Malladi, Sethian et Vemuri [Malladi et al. 1995]. Un ensemble de niveau (*level set*) peut librement se diviser pour représenter plusieurs objets. La dénomination est générique. En 2D, il s'agit de courbes de niveau, en 3D de surfaces de niveau. La surface de l'objet est modélisée grâce à la propagation d'une interface solide/liquide dont la vitesse dépend de la courbure. Le front de l'interface coule suivant son champ de gradient. Il se déplace au cours du temps en résolvant une équation de type Hamilton-Jacobi qui est écrite pour une fonction  $\psi$  dont l'interface est un ensemble de niveau particulier ( $\psi = 0$ ). Dans le cas de traitement de données 2D l'interface est une courbe fermée de niveau d'une surface  $\psi(x, y)$  (figure I●3.5). Pour des données 3D, l'interface est une surface fermée de niveau d'une hypersurface  $\psi(x, y, z)$ .

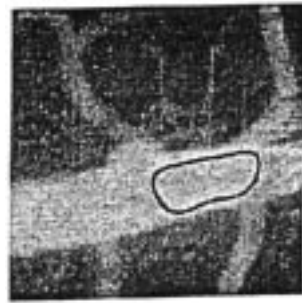
[Whitaker 1995] propose une méthode de résolution de l'équation Hamilton-Jacobi adaptée à un champ peu dense, qui mène à un ensemble de niveau discret. Le temps de calcul est réduit, car seuls les voxels qui se trouvent dans le voisinage immédiat des voxels de la surface de niveau sont pris en compte pour l'évolution du modèle. Un exemple d'évolution d'un cube vers un tore prend toutefois une minute sur une SPARCstation 20 pour une image  $32 \times 32 \times 32$ .

## I●3.3 Modèles volumiques

Nous appellerons modèle volumique un modèle déformable dont les éléments de la représentation géométrique sont répartis dans l'espace ou une portion de l'espace. Détaillons les modèles volumiques selon la nature de ces éléments.

### 3.3.1 Représentation géométrique constituée par des voxels

De nombreux modèles déformables sont composés de tous les voxels d'une image 3D [Bajcsy et Kovacic 1989 ; Studholme et al. 1996 ; Wells et al. 1996]. D'autres modèles ne contiennent que les voxels d'un objet qui évolue dans l'image. Citons le modèle de [Mangin 1995] qui est composé de régions déformables pour lesquelles on possède des connaissances topologiques a priori, et le modèle de croissance d'os de Bro-Nielsen [Bro-Nielsen et al. 1997], qui évolue par agrégation de nouveaux voxels. La figure I●3.6 montre sur une image simple la différence entre le cas où tous les voxels de forment le modèle, et le cas où le modèle est strictement inclus dans l'image.



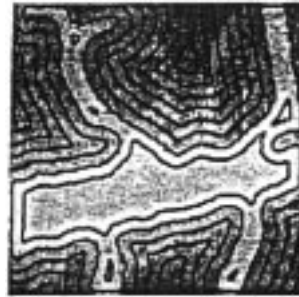
(a) courbe initiale,  
t=0.000



(b)  $\psi(x, 0.000)$



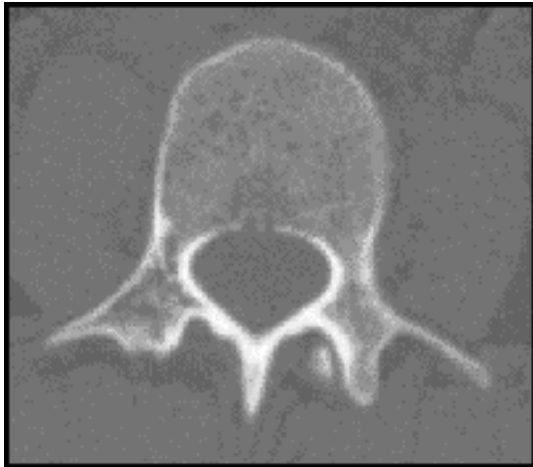
(c) courbe finale,  
t=0.375



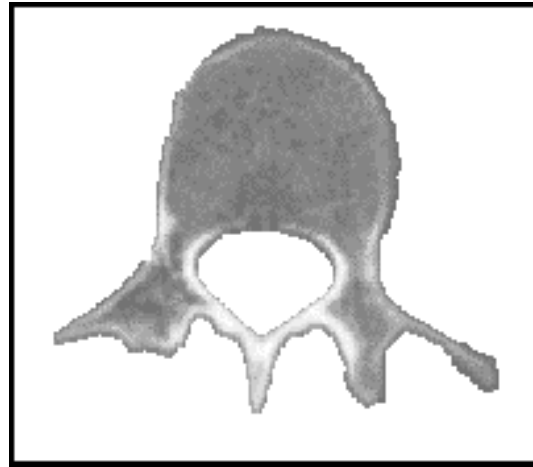
(d)  $\psi(x, 0.375)$

FIG. I●3.5: Modèle par ensembles de niveau, d'après [Malladi et al. 1995]

La colonne de gauche montre l'évolution du modèle, celle de droite les ensembles de niveau de la fonction  $\psi$  associée.



(a) tous les voxels de l'image forment le modèle



(b) le modèle est strictement inclus dans l'image

FIG. I●3.6: Représentation géométrique constituée de voxels

### 3.3.2 Ensemble de particules

Dans cette approche originale, les éléments géométriques sont répartis dans l'espace sans avoir de lien explicite entre eux. C'est le choix de la déformation qui garantit la

cohérence du modèle. Le représentant de cette catégorie est le modèle proposé par [Szeliski et Lavallée 1996]. Ce modèle volumique est appelé octree-spline, d'après la méthode de déformation utilisée. La représentation géométrique est constituée d'un ensemble de particules réparties dans l'espace. Elles peuvent être de simples points ou comprendre d'autres caractéristiques comme nous le verrons dans la partie IV .

### 3.3.3 Maillage d'un objet

Dans les modèles de cette section, l'information est portée par les nœuds du maillage volumique. Le maillage recouvre uniquement un objet qui se déforme dans les deux premiers types de modèles, le maillage simplexe, et les réseaux explicites d'éléments ayant une masse. Dans le dernier modèle, les pyramides actives, c'est une image 3D toute entière qui est maillée.

#### 3.3.3.1 Maillage simplexe

Les maillages simplexes surfaciques se généralisent en maillages simplexes volumiques. Ce sont des tétraédrisations d'objets volumiques. Ils ont été utilisés pour des applications de simulation [Cotin et al. 1996].

#### 3.3.3.2 Réseaux explicites d'éléments ayant une masse

Nous avons donné dans la section 3.1.1 des exemples de modèles déformables surfaciques pour lesquels les sommets des polygones portent une masse. On trouve de même un ensemble de modèles dans lesquels un réseau de masses discrétisent des objets volumiques. On notera par exemple les travaux de [Luciani et Cadoz 1986] et [Gascuel 1990] dans le domaine de l'animation, et ceux de [Jimenez 1993] et [Joukhadar 1997] en modélisation. La voie est ouverte à ce type de modèle dans le domaine du traitement d'images médicales.

#### 3.3.3.3 Pyramide active

[Magnin et Reissman 1996] ont présenté une représentation des données par une pyramide active. Chaque niveau de la pyramide est un graphe régulier construit récursivement sur une image originelle convoluée par un filtre passe-bas. Les cellules de ce graphe s'adaptent au contenu local des images. L'adaptation du graphe est obtenue en minimisant une fonctionnelle énergétique fondée sur le gradient de l'image et la déformation locale des cellules. Les éléments qui portent l'information de bas niveau sont les cellules du graphe. Les auteurs utilisent ces pyramides actives en imagerie dynamique d'objets subissant des transformations élastiques.

## I•3.4 Discussion

Les trois types d'approche, surfacique, implicite et volumique se distinguent par des options différentes en termes de caractéristiques de liaison. Dans les modèles surfaciques, la fonction de liaison est un attribut de l'image des données. On calcule la valeur de cette fonction pour les caractéristiques du modèle. Dans les modèles implicites, c'est la



---

fonction qui définit l'équation implicite qui constitue le cœur de la fonction de liaison. Il s'agit donc d'un attribut du modèle. La valeur de cette fonction est calculée pour les points des données. Dans les modèles volumiques, la symétrie entre le modèle et les données permet de choisir dans quel sens calculer la fonction de liaison.

# Chapitre I•4

## Déformation et interactivité

Nous allons étudier dans ce chapitre les différents moyens de déformation, directes et indirectes, des modèles déformables. Nous nous intéresserons au caractère global ou local de chaque déformation. Nous aborderons également la possibilité de déformer ces modèles interactivement.

### I•4.1 Déformations indirectes

Comme nous l'avons vu dans les définitions du paragraphe 1.1.3 du chapitre I•1, les déformations indirectes sont des transformations non-rigides de l'espace.

#### 4.1.1 Déformations globales

Les déformations globales sont définies par un petit nombre de paramètres, et ne permettent donc de réaliser qu'un domaine limité de transformations.

##### 4.1.1.1 Déformations affines

La déformation affine est une des transformations non-rigides les plus simples. Toute transformation affine s'écrit de manière unique comme la composition d'un déplacement rigide, d'un changement d'échelle indépendant dans la direction de chacun des axes, et d'un autre déplacement rigide.

$$\mathbf{T}_p^g(\mathbf{d}_i) = \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \end{bmatrix} \begin{bmatrix} 1 \\ x_i \\ y_i \\ z_i \end{bmatrix} \quad (\text{I•4.1})$$

Les  $\mathbf{d}_i = (x_i, y_i, z_i)$  sont les coordonnées des points  $i$  dans  $\text{Ref}_{\text{modèle}}$ .

Une déformation globale affine n'est généralement pas suffisante pour mettre en correspondance les images de deux personnes différentes [Feldmar 1995]. Elle peut par contre être utilisée pour mettre en correspondance les images CT et IRM d'un même patient [Studholme et al. 1996].

##### 4.1.1.2 Déformations polynomiales globales

Une transformation trilineaire permet 24 degrés de liberté. Elle peut aussi être paramétrée par la position des huit sommets d'un cube englobant le modèle, mais les équations

sont plus compliquées. C'est la déformation qui a été choisie par [Jacq et Roux 1995 ; Rouet et al. 1997].

$$\mathbf{T}_{\mathbf{p}}^g(\mathbf{d}_i) = \begin{bmatrix} p_{00} & p_{01} & & p_{07} \\ p_{10} & p_{11} & \cdots & p_{17} \\ p_{20} & p_{21} & & p_{27} \end{bmatrix} \begin{bmatrix} 1 & x_i & y_i & z_i & x_i y_i & y_i z_i & z_i x_i & x_i y_i z_i \end{bmatrix}^T \quad (\text{I}\bullet 4.2)$$

On obtient de façon similaire les 30 paramètres de la famille des transformations quadratiques.

$$\mathbf{T}_{\mathbf{p}}^g(\mathbf{d}_i) = \begin{bmatrix} p_{00} & p_{01} & & p_{09} \\ p_{10} & p_{11} & \cdots & p_{19} \\ p_{20} & p_{21} & & p_{29} \end{bmatrix} \begin{bmatrix} 1 & x_i & y_i & z_i & x_i y_i & y_i z_i & z_i x_i & x_i^2 & y_i^2 & z_i^2 \end{bmatrix}^T \quad (\text{I}\bullet 4.3)$$

Toutes ces transformations ont en commun la propriété que la position du point transformé  $\mathbf{r}_i = \mathbf{T}_{\mathbf{p}}^g(\mathbf{d}_i)$  est une fonction *linéaire* du paramètre  $\mathbf{p}$ . C'est à dire que la transformation peut s'écrire  $\mathbf{r}_i = \mathbf{M}(\mathbf{d}_i) \cdot \mathbf{p} = \mathbf{M}_i \cdot \mathbf{p}$  [Witkin et Welch 1990]. Par exemple la transformation affine peut être représentée par

$$\mathbf{M}_i = \begin{bmatrix} 1 & 0 & 0 & x_i & 0 & 0 & y_i & 0 & 0 & z_i & 0 & 0 \\ 0 & 1 & 0 & 0 & x_i & 0 & 0 & y_i & 0 & 0 & z_i & 0 \\ 0 & 0 & 1 & 0 & 0 & x_i & 0 & 0 & y_i & 0 & 0 & z_i \end{bmatrix},$$

avec le vecteur des paramètres  $\mathbf{p} = [p_{00} \ p_{10} \ \dots \ p_{23}]^T$ .

Au fur et à mesure que l'on ajoute des paramètres globaux au modèle de transformation, la possibilité de déterminer leurs valeurs de façon stable se détériore.

## 4.1.2 Déformations locales

Les déformations locales sont dotées d'un nombre plus grand de paramètres, qui leur permet de réaliser un domaine plus important de déformations que les déformations globales. Cette définition reste empirique, mais elle est utile à la classification. Nous y reviendrons lorsque nous aurons détaillé les différentes déformations locales. Celles-ci sont généralement calculées sur une base spline ou bien sur une grille de vecteurs.

### 4.1.2.1 Déformations splines

La transformation peut être calculée sur une base spline. De nombreux auteurs utilisent les *splines plaques-minces* de Duchon [Bookstein 1989 ; Champeboux 1991 ; Kim et al. 1996 ; Evans et al. 1996 ; Rohr et al. 1996]. Ces splines ont été créés initialement pour de la conception et fabrication assistée par ordinateur, il s'agissait de calculer la déformation d'une plaque de métal mince comme celles qui recouvrent les ailes d'avions lorsqu'on contraint un certain nombre de points  $(x, y)$  à des déplacements  $z$ . La formalisation mathématique de ces splines a conduit à la définition de la famille des *splines généralisées* à laquelle appartiennent les splines plaques-minces. Un autre membre de cette famille est la *spline volumique*. [Nielson 1993a,b] suggère qu'elle est mieux adaptée aux données 3D que la spline plaque-mince.

[Davis et al. 1997] ont proposé une déformation indirecte qu'ils appellent *corps élastique spline* (elastic body spline). Il s'agit d'une spline 3-D qui est fondée sur le modèle physique d'un matériau élastique 3D homogène et isotrope, et qui repose sur les équations différentielles partielles de Navier. Le corps élastique spline est adapté au cas particulier où un seul objet qui se déforme est présent dans deux images, comme dans l'exemple des IRM du sein qui illustre l'article, il permet d'inférer avec plus de précision la déformation générale à partir de l'appariement de points de contrôle que ne le font les splines plaque minces ou les splines volumiques. Mais le corps élastique spline nécessite plus de calcul que ces deux dernières splines.

#### 4.1.2.2 Déformations nodales

Les déformations nodales sont construites à partir des déplacements des nœuds d'un maillage.

**Déformations de forme libre** Il s'agit de déformer une portion de l'espace, afin d'induire les déformations des objets qui y sont inclus. On compare souvent cette approche à la déformation d'un bloc de gélatine, qui permet de déformer les objets qui seraient pris dans ce bloc (Figure I●4.1).

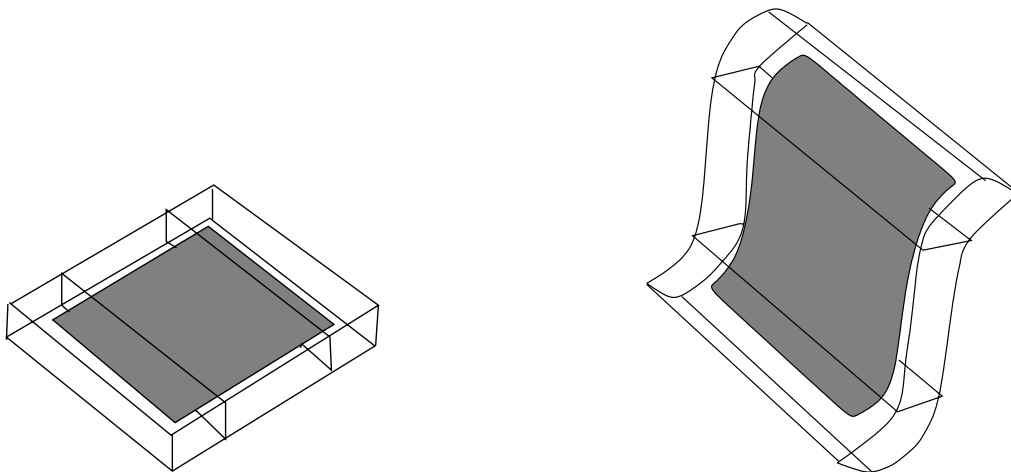


FIG. I●4.1: Déformations de forme libre d'un morceau de surface

La déformation est contrôlée par le treillis régulier des points de contrôle d'une fonction d'interpolation tridimensionnelle de type spline. On l'appelle déformation de forme libre (Free Form Deformation ou FFD), ou déformation de forme libre étendue (Extended Free Form Deformation, EFFD) lorsque le treillis utilisé n'est pas régulier, mais adapté à l'objet à déformer [Coquillard 1990].

**FFD classique** Cette approche a d'abord été utilisée dans le domaine de la modélisation et de l'animation d'objets de synthèse [Sederberg et Parry 1986 ; Chadwick et al. 1989 ; Witkin et Welch 1990], puis également pour l'imagerie médicale [Bardinet et al. 1994 ; Bardinet 1995 ; Robert et al. 1995], avec des treillis réguliers.

**FFD hiérarchique** [Szeliski et Lavallée 1994 ; Lavallée et al. 1996] augmentent l'efficacité de cette approche en utilisant un *octree spline de déformation*, qui est une représentation multirésolution de la spline de déformation [Forsey et Bartels 1988], fondée sur le concept de fonctions de base hiérarchiques [Szeliski 1990]. Les éléments géométriques représentent l'information qui guide la déformation. La résolution de l'octree-spline augmente au voisinage de ces éléments, ce qui permet d'obtenir une déformation précise là où l'information est disponible. La déformation du reste du volume est inférée par l'intermédiaire de cubes de l'octree-spline d'une taille plus grande (Figure I●4.2).

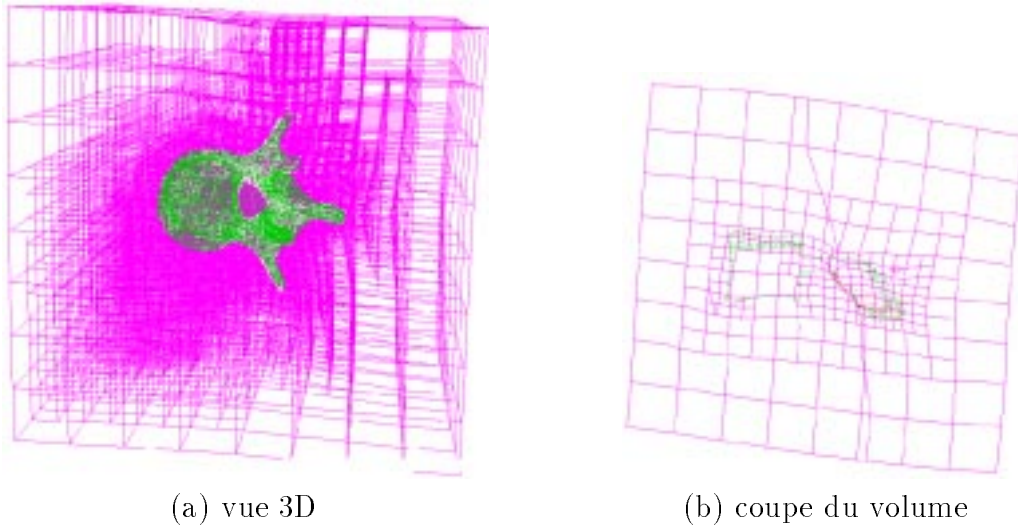


FIG. I●4.2: Avec l'octree spline de déformation, le maillage est raffiné au voisinage de la représentation géométrique.

**Maillage structuré adaptatif** Une autre manière d'obtenir un maillage plus dense au voisinage des éléments géométriques est de partir d'un maillage régulier, et de le déformer en attirant les nœuds vers ces éléments. Cette méthode, classique en dynamique des fluides, a été reprise par [Benayoun 1994] pour calculer un champ de déplacement sur des séquences d'images.

## I●4.2 Déformations directes

Parmi les déformations directes, pour lesquelles la déformation est par essence liée à la représentation géométrique, nous ferons la distinction entre les **déplacements explicites** des éléments géométriques, et les **déplacements paramétriques**.

### 4.2.1 Déformations directes explicites

Ces déformations sont l'apanage des modèles dont la représentation géométrique est maillée, que ce soient des modèles surfaciques ou volumiques. Les nœuds du maillage contiennent l'information de bas niveau, et leur position courante est donnée en appliquant un vecteur de déplacement à une certaine position, qui peut être la position précédente ou une position de référence. Nous appellerons cette position **position de base** à l'instar de la configuration de base des déformations indirectes.

Les modèles dont la déformation est directe explicite sont les modèles surfaciques polygonaux - exception faite des formes actives - et les objets volumiques maillés. Nous les avons décrits dans le chapitre I•3, aux paragraphes 3.1.1 et 3.3.3.

#### **4.2.1.1 Déformations fluides**

Elles sont basées sur la déformation linéaire élastique du champ de vitesse d'un fluide visqueux. Une telle approche permet des déformations importantes, tout en garantissant la conservation de la topologie du modèle. Mais les calculs sont très longs. [Christensen et al. 1996] traitent une image 3D en 2 à 6 heures sur une machine parallèle DECmpp 128x64 MasPar. [Bro-Nielsen et Gramkow 1996] accélèrent les calculs en utilisant des filtres de convolution, ils obtiennent tout de même des temps de calcul du même ordre, sur une station de travail, toutefois.

### **4.2.2 Déformations directes paramétriques**

La représentation géométrique des modèles qui composent cette catégorie est obtenue à partir d'une fonction, soit de manière explicite pour les modèles construits sur des bases de fonctions (3.1.2), soit de manière implicite, pour les modèles implicites (I•3.2). La déformation du modèle est obtenue en modifiant les paramètres de cette fonction, c'est pourquoi nous l'avons nommée paramétrique. Pour représenter un objet donné à partir d'un modèle de ce genre, une inversion de la fonction est nécessaire afin de trouver les paramètres qui induiront la forme recherchée.

En plus des modèles cités (ceux construits sur des bases de fonctions, et les implicites), on trouve dans cette catégorie les modèles à déformations modales, et le cas de la déformation localement affine que nous allons maintenant présenter.

#### **4.2.2.1 Déformation localement affine**

[Feldmar 1995] a mis au point une méthode pour recaler deux ensembles de points, en utilisant les appariements de points. Sa déformation est localement affine dans le sens que chaque point à déformer est doté d'une déformation affine. Un processus de régularisation assure que ces déformations ne varient pas trop entre un point et son voisin. Chaque paire de points détermine un déplacement, qui est celui qui réalise cet appariement. Feldmar calcule pour chaque point de son modèle les déplacements appariant les voisins de ce point. Il utilise une pondération de ces déplacements pour calculer la transformation affine associée à ce point. Nous avons classé ce mode de déformation comme directe car chaque point est associé à sa transformation affine locale.

#### **4.2.2.2 Déformations modales physiques**

Pentland et Williams [Pentland et Williams 1989] ont les premiers proposé des déformations modales. Ils se sont inspirés des modes de vibrations libres d'objets physiques. Pentland a par la suite développé ce modèle avec l'aide de Sclaroff [Pentland et Sclaroff 1991 ; Sclaroff et Pentland 1994]. Ils ont utilisé une approche fondée sur la méthode des éléments finis (MEF), et la modélisation de solides paramétriques à l'aide de fonctions implicites. Dans cette formulation, les degrés de liberté sont orthogonaux, et donc découplés quand on pose les équations de la dynamique en termes de vecteurs propres de

la MEF. Ces vecteurs propres sont les modes de déformations libres de l'objet, ou modes de déformation. Ils forment une base orthogonale ordonnée en fréquences, qui est analogue à la transformée de Fourier. Les modes de fréquences les plus basses correspondent toujours aux modes rigides de translation et de rotation.

#### 4.2.2.3 Déformations modales statistiques

Décrivons plus en détail le cas des formes actives [Cootes et al. 1994, 1995]. Nous l'avons brièvement évoqué au paragraphe 3.1.1 du chapitre I●3, ce modèle représente les objets par des ensembles de points placés à la main sur une partie spécifique de l'objet. Cette approche nécessite des données d'apprentissage, un ensemble d'images semblables à celles qui devront être traitées par le modèle. Elles sont recalées entre elles afin d'aligner les points équivalents des différentes images. Cela se fait par rotation, translation et mise à l'échelle des formes apprises, en minimisant une somme pondérée du carré des distances entre points équivalents.

La difficulté de la méthode réside en ce que les points doivent être placés de façon très précise dans les images d'apprentissage, afin d'établir la correspondance entre les points qui se situent aux mêmes endroits de ces images. Pour cela, l'alignement des images doit également être très précis. L'analyse de la répartition statistique des points permet de créer un "Modèle de Distribution des Points" (Point Distribution Model, en anglais). Ce modèle comprend la position moyenne des points, et une description des principaux modes de variation trouvés dans la phase d'apprentissage. Les modes de variation sont obtenus grâce à une analyse en composantes principales : ce sont les vecteurs propres unitaires de la matrice de covariance. Plus la valeur propre associée est grande, plus le mode est significatif. La déformation du modèle est généralement obtenue à l'aide d'un petit nombre de modes. La figure I●4.3 montre l'influence des deux premiers modes obtenus pour un exemple de forme active.

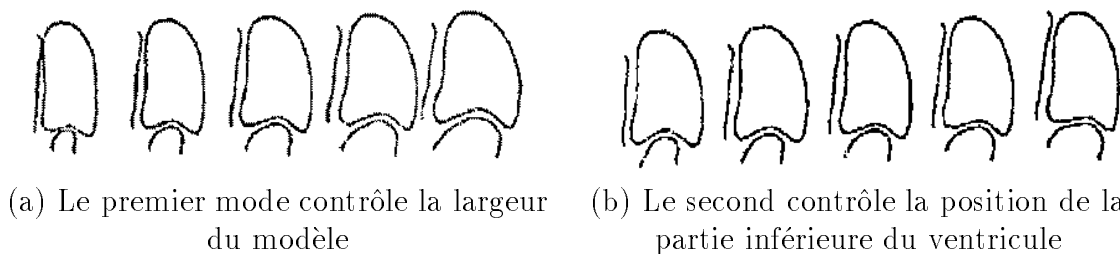


FIG. I●4.3: Influence des variations individuelles des deux premiers paramètres d'un Modèle de Distribution des Points d'un ventricule gauche comprenant 96 points, d'après [Cootes et al. 1994]

La déformation est directe et paramétrique, car pour ajuster les positions des points, une fois trouvé le déplacement désiré pour chacun de ces points, il faut calculer par une étape de minimisation la variation des paramètres de forme qui permettra d'obtenir ces déplacements.

## I●4.3 Combinaison de déformations

Certains auteurs utilisent conjointement deux modèles de déformation, afin de bénéficier du faible nombre de paramètres qui permettent de définir une déformation globale d'une part, et de la précision qu'offre une déformation locale. Les paramètres de définition de la déformation locale ne sont alors ajoutés qu'aux endroits précis où ils sont nécessaires.

### 4.3.1 Superquadriques et déplacements locaux

[Sclaroff et Pentland 1994] déforment leurs superquadriques en utilisant conjointement des déformations modales et une carte de déplacements, qui décrit les détails de la surface à l'aide d'ondelettes multiéchelles. [Terzopoulos et Metaxas 1991] utilisent une approche fondée sur la physique pour déformer des superquadriques de manière locale et globale. [DeCarlo et Metaxas 1994] se servent de déformations globales, puis des paramètres intrinsèques de leur modèle (paramètres de formes et paramètres de mélange), et ajoutent des déplacements locaux.

### 4.3.2 Superquadriques, éléments finis et ondelettes

[Vemuri et Radisavljevic 1994 ; Vemuri et al. 1996] ont introduit un modèle qui comprend des paramètres de forme globaux et locaux. Au niveau global, la forme est déterminée par une superquadrique, et au niveau local par une discrétisation en éléments finis. Une base orthonormale d'ondelettes sert à lisser la déformation pour créer des niveaux intermédiaires. Les auteurs appellent leur modèle *modèle hybride multirésolution*.

### 4.3.3 Déformations globales et octree-spline

[Szeliski et Lavallée 1994] utilisent également les combinaisons de déformation pour obtenir une meilleure convergence de leur algorithme. Ils opèrent d'abord une déformation globale quadratique, avant de calculer les déformations locales de l'octree-spline.

## I●4.4 Interactivité

L'interactivité est capitale quand on ne peut pas faire du tout automatique. Il est plus facile de passer du tout à la main au semi-automatique, avec une contribution de l'utilisateur bien gérée (facilitée en termes d'Interface Homme-Machine), que d'atteindre le tout automatique.

### 4.4.1 Modification de la géométrie du modèle

Considérons que cette déformation est constituée par le déplacement dans l'espace d'un point d'un objet. Ce point peut-être un point de contrôle de la géométrie de l'objet, ou directement un point de cette géométrie [Hsu et al. 1992].

La modélisation interactive existe également dans d'autres domaines que le domaine médical, comme l'animation de personnages [Turner 1995].



## 4.4.2 Dispositifs de Réalité Virtuelle

La possibilité pour l'utilisateur de déformer interactivement un objet est rendue plus intuitive avec les dispositifs de réalité virtuelle. Nous présentons dans cette section les paradigmes de ce domaine, et donnons des exemples d'utilisation de ces dispositifs.

La manipulation d'objets 3D dans le but de les observer s'est bien accommodée des limitations de la souris 2D. A travers des concepts comme la boule virtuelle [AVS 1994] les rotations et translations d'un objet deviennent simples à effectuer avec une souris ordinaire.

Pour ce qui est de la déformation interactive d'objets, la question est plus complexe. En effet, elle met en jeu à la fois la définition du point de l'objet à déformer, et la position finale de ce point, en même temps que la visualisation de la déformation générée, qui permet par un effet de rétroaction de corriger la déformation au moment même du déplacement du point.

Trois familles d'outils de réalité virtuelle permettent de résoudre cette question [Burdea et Coiffet 1993].

### 4.4.2.1 Repérage 6D du pointeur

Une gamme de capteurs existe, pour repérer la position et l'orientation du pointeur dans l'espace. Les capteurs mécaniques sont constitués d'un bras articulé, muni de codeurs, qui porte le pointeur. Les capteurs magnétiques permettent un grand volume de travail, mais sont perturbés par les objets métalliques. Les capteurs à ultrasons ou à infrarouges nécessitent une visibilité directe entre l'émetteur et le récepteur.

### 4.4.2.2 Vision stéréoscopique

La vision d'un objet par l'œil droit et par l'œil gauche n'est pas la même. En présentant à l'utilisateur une image différente pour ses deux yeux il est possible de recréer l'illusion du relief. Outre le fait qu'il faut calculer deux fois plus d'images, il faut utiliser un dispositif adapté pour afficher les images. Ces dispositifs se classent en deux catégories, selon qu'ils coupent l'utilisateur ou non du monde réel.

**Immersion totale** Le procédé d'immersion totale plonge l'utilisateur totalement dans un monde virtuel.

Les casques de visualisation sont portés sur la tête et placent devant les yeux deux mini écrans. Les écrans peuvent être à cristaux liquides, mais ils ont une faible résolution, ou bien être composés de tubes cathodiques miniatures placés sur les côtés de la tête, et réfléchis par des miroirs. Cette dernière technologie est plus chère, mais permet une meilleure résolution.

Les moniteurs binoculaires omni-directionnels (BOOMS) sont placés au bout d'un bras articulé, et en évitant la contrainte de porter les écrans sur la tête, proposent un champ de vision plus important que les casques, car ils peuvent inclure des écrans cathodiques de plus grandes dimensions. Ces dispositifs sont cependant chers et encombrants.

L'inconvénient principal de l'immersion totale est le mal des simulateurs. Ce mal apparaît lorsque la perception visuelle du monde virtuelle entre en conflit avec la perception physique du monde réel par l'oreille interne.

---

**Immersion partielle** Les dispositifs d'immersion partielle n'induisent pas ce mal. Ils montrent le monde virtuel sans priver l'utilisateur de la vue du monde réel.

Les lunettes semi-transparentes sont composées d'écrans à cristaux liquides semi-transparentes.

Les lunettes stéréo fonctionnent en obstruant alternativement chaque œil par des cristaux liquides, à une fréquence synchronisée avec celle de l'écran qui affiche en alternance les images pour chaque œil. Plusieurs personnes peuvent ainsi partager la même image stéréo.

Des écrans stéréo existent également, qui utilisent le multiplexage spatial. Les écrans micropolaires sont une autre sorte d'écrans que l'on regarde avec des lunettes polarisées passives.

#### 4.4.2.3 Repérage 6D du point de vue

Les mêmes techniques de repérage dans l'espace que dans le paragraphe 4.4.2.1 sont utilisées. L'application est différente : la connaissance temps réel de la position de la tête de l'utilisateur permet d'inférer la localisation de chacun des yeux, afin de calculer le point de vue de l'utilisateur. Il est alors possible de simuler la présence d'un objet fixe autour duquel l'on pourrait se déplacer. La perception de la profondeur est ainsi encore améliorée.

#### 4.4.2.4 Utilisation de ces dispositifs

Le système HoloSketch de Deering [Deering 1995] offre un dispositif de suivi de la main et de la tête pour améliorer la modélisation des formes. Turner et col. [Turner et al. 1996] utilisent à la fois un dispositif de repérage 3D du pointeur, le suivi de la tête de l'utilisateur et des lunettes semi-transparentes CrystalEyes pour offrir un système de manipulation et déformation interactif d'objets de synthèse. Leurs applications sont dans le domaine de la construction de personnages animés. Grâce au repérage de la position et de l'orientation de la tête de l'utilisateur, leur système une fois calibré mélange exactement le monde virtuel et le monde réel : l'utilisateur voit dans le prolongement précis de son pointeur, qui est bien réel, un outil virtuel, par lequel il peut déformer les objets virtuels.

## I●4.5 Discussion

Nous avons vu que les méthodes de déformation directes et indirectes des modèles déformables sont variées. Il est important de disposer de méthodes de déformation globales, pour coder de manière économique les paramètres généraux d'une forme. Mais ces méthodes sont insuffisantes pour prendre en compte les détails. Il faut également disposer de méthodes de déformation locales.

Si l'on dispose d'un ensemble significatif d'images pour créer le modèle, l'approche par déformations modales statistiques s'impose parce qu'elle permet des déformations globales et locales dans un même contexte. Il faut cependant pour cette approche être capable de positionner de manière précise et répétable les points de caractéristiques de bas niveau dans les images d'entraînement. Cette approche également comme point fort de distinguer les déformations qui sont acceptables de celles qui ne le sont pas. Nous verrons dans le chapitre suivant d'autres moyens de coder cette déformabilité.

---

La déformation peut concerner une surface ou un volume, le choix se fera en fonction de l'application et du type des données. Pour la reconstruction d'une surface, une déformation surfacique convient. Pour la segmentation d'images volumiques, les deux types de déformation sont à considérer. Nous privilégierons la déformation volumique, car elle peut être appliquée sur une représentation géométrique également volumique, utilisant ainsi plus d'informations.

Si la déformation automatique n'est pas satisfaisante, le recours à une déformation en partie interactive permet de corriger le modèle selon la convenance de l'utilisateur.

Nous allons voir dans le chapitre suivant les contraintes à appliquer sur les déformations pour qu'elles restent satisfaisantes, et les algorithmes de contrôle qui déterminent les paramètres de la déformation quand elle est automatiquement calculée.

### I•5.1 Déformabilité

L'inconvénient des déformations globales est d'apporter trop peu de degrés de liberté, et donc de ne pas déformer le modèle d'une manière suffisante pour épouser la forme des données. Les déformations locales, au contraire, fournissent tellement de degrés de liberté qu'elles peuvent modifier totalement la forme du modèle, dénaturant la démarche de l'utilisation des modèles déformables, qui est de se servir de la forme du modèle pour traiter une image.

C'est l'introduction de contraintes qui permet de s'assurer que la forme du modèle est admissible. La formulation de la contrainte est dite **forte** si le modèle est forcé de vérifier cette contrainte à tout instant. Si au contraire le modèle essaye de vérifier la contrainte sans y être forcé, la formulation est dite **faible** [Bainville 1996]. Dans ce cas, la contrainte est introduite à travers un terme comme une énergie. Plus ce terme est faible, mieux la contrainte est vérifiée. Lorsqu'un modèle doit satisfaire à de nombreuses contraintes qui ne peuvent pas être exactement vérifiées simultanément, la formulation faible est utilisée afin de vérifier chacune au mieux.

Voyons maintenant des exemples de ces contraintes.

#### 5.1.1 Contrainte globale

[Promayon et al. 1997] imposent une contrainte globale à leur modèle, qui est la conservation du volume intérieur à l'objet. Ils utilisent une méthode directe, sans processus itératif, qui assure une formulation forte. Le vecteur de déplacement est projeté sur le domaine de vérification de la contrainte. La contrainte doit avoir une expression continue, et différentiable au moins une fois.

#### 5.1.2 Déformations modales statistiques

L'intérêt capital du modèle des formes actives vient de ce que la phase d'apprentissage permet de définir le domaine des formes acceptables, et de forcer les déformations à s'y trouver. Si les paramètres de forme calculés à une étape sortent du domaine admissible, les auteurs s'y ramènent par un changement d'échelle. Cette méthode est la mieux adaptée si l'on dispose de statistiques sur la forme recherchée.

### 5.1.3 Viscosité d'un fluide

Dans les approches par déformation fluide, c'est le paramètre de viscosité qui interdit toute déformation trop brutale, et l'incompressibilité du fluide qui garantit la cohérence du modèle déformé.

### 5.1.4 Regularisation

[Delingette 1994] propose une étude des forces de régularisation dans laquelle il étudie les différentes mesures de l'égalité (*smoothness*) d'une surface, en distinguant d'une part les fonctionnelles basées sur la physique qui ont d'intéressantes propriétés d'invariance, mais qui sont difficiles à implémenter, et d'autre part les mesures quadratiques qui ne garantissent pas que la solutions est "acceptable". Il définit des *stabilisateurs différentiels polynomiaux* qui permettent une formulation intrinsèque de la forme du modèle.

La déformation d'un maillage peut également simplement être régularisée en imposant que les déplacements soient de faible amplitude (régularisation d'ordre 0), ou que les déplacements de deux nœuds voisins restent proches (régularisation d'ordre 1). La régularisation est ajoutée lors du calcul de la déformation du modèle [Szeliski et Lavallée 1994 ; Benayoun 1994], ou bien elle est inhérente à la déformation du modèle, comme c'est le cas pour les splines plaques-minces.

### 5.1.5 Energie de tension

[Sclaroff et Pentland 1994] ajoutent à leur modèle à déformations modales une *énergie de tension* qui joue le même rôle que la régularisation. Cette énergie est calculée en fonction de la raideur associée à chacun des modes de déformation.

## I•5.2 Contrôle

Le contrôle est la méthode ou l'algorithme utilisé pour calculer l'état du modèle déformable en fonction des contraintes. Si le modèle suit une loi d'évolution dynamique, le contrôle consiste à intégrer les équations différentielles du mouvement. Si le modèle déformable est associé à une fonctionnelle, il s'agit de son optimisation.

Les méthodes d'intégration des équations différentielles du mouvement les plus utilisées sont celles d'Euler ou de Runge-Kutta [Press et al. 1992]. Nous ne les détaillerons pas, car les modèles que nous avons implémentés ne sont pas définis par des équations différentielles. Nous allons exposer différentes méthodes d'optimisation.

Ce chapitre est une présentation des méthodes d'optimisation les plus utilisées actuellement. Un document de référence qui a une approche très pratique de la question est [Press et al. 1992].

## I•5.3 Optimisation d'une fonctionnelle

Notons  $f(\mathbf{p})$  la fonctionnelle à minimiser. Elle dépend du paramètre d'état du modèle. Parmi les méthodes d'optimisation, certaines utilisent les dérivées de  $f$  (c'est-à-dire le gradient de  $f$  par rapport aux paramètres du modèle), d'autres non.

Le choix de l'une ou l'autre de ces méthodes dépendra donc de la possibilité de calculer ces dérivées, et du coût que cela représente.

### 5.3.1 Descente de gradient

Il s'agit d'une méthode qui permet toujours de trouver un minimum local de la fonction, mais sa convergence est très lente. Elle consiste à se déplacer, à partir du point courant  $\mathbf{p}^{(k)}$  dans la direction de plus forte pente :

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \lambda \nabla f(\mathbf{p}^{(k)})$$

Il est toujours possible de trouver un réel positif  $\lambda$  tel que  $f(\mathbf{p}^{(k+1)}) < f(\mathbf{p}^{(k)})$ , mais ce réel peut être extrêmement petit, ce qui correspond à un déplacement très faible.

### 5.3.2 Méthode de Newton

Si l'on peut calculer les dérivées secondes de  $f$ , la méthode de Newton s'avère beaucoup plus rapide que la descente de gradient. Il s'agit d'approcher  $f$  par son développement de Taylor d'ordre 2, et d'inverser la matrice Hessienne. Cette méthode nécessite que la matrice Hessienne  $\nabla^2 f(\mathbf{p}^{(k)})$  soit définie positive.

### 5.3.3 Algorithme de Levenberg-Marquardt

Cet algorithme combine la sécurité de la descente de gradient et l'efficacité de la méthode de Newton. L'algorithme de Levenberg-Marquardt construit une approximation de la matrice Hessienne  $\mathbf{A}$  et du vecteur gradient  $\mathbf{b}$  qui représente l'erreur.

La matrice Hessienne est rendue définie positive par une perturbation des éléments de sa diagonale.

La résolution de l'équation :

$$(\mathbf{A} + \lambda \text{diag}(\mathbf{A})) \Delta \mathbf{p}^{(k)} = \mathbf{b}, \quad (\text{I}\bullet 5.1)$$

permet de calculer un incrément  $\Delta \mathbf{p}$  qui va rapprocher le vecteur paramètre du minimum local.

Si  $\lambda$  est grand, la diagonale est dominante, et on retrouve la méthode de descente de gradient. Si au contraire  $\lambda$  est petit, on retrouve la méthode de Newton.

Nous avons utilisé cet algorithme pour sa robustesse dans notre modèle volumique déformable (voir IV•5).

### 5.3.4 Gradient conjugué

La descente de gradient oblige à avancer à tout petits pas quand il s'agit de descendre le long d'une longue vallée étroite, même si cette vallée a une forme parfaitement quadratique. La méthode des gradients conjugués remédie à cela en choisissant une direction qui est conjuguée à celle du gradient précédent, et de préférence à toutes les directions précédemment choisies, aussi loin que possible.

---

### 5.3.5 Recuit simulé

Cette méthode stochastique n'utilise pas les dérivées de  $f$ . Elle permet de trouver un extremum global si celui-ci est entouré d'extréma plus petits. Au cœur de cette méthode se trouve une analogie avec la thermodynamique, avec la manière dont les liquides gèlent et cristallisent. En refroidissant lentement un liquide, on peut le faire geler et devenir un cristal pur, alors que si le refroidissement est rapide, le liquide n'atteint pas ce minimum d'énergie représenté par cette forme cristallisée. L'optimisation nécessite de définir une "température" pour le système.

Le recuit simulé est une méthode puissante, puisqu'elle donne des chances de trouver un extremum global, mais pour ce faire, il faut prendre son temps pour "refroidir" le système.

### 5.3.6 Algorithmes génétiques

Les algorithmes génétiques sont une autre méthode stochastique, qui présente de manière générale les mêmes avantages et inconvénients que le recuit simulé. Il s'agit ici de recombinaison des paramètres d'état entre eux, à la manière dont les gènes se mélangent.

Les travaux de [Jacq et Roux 1995] portent sur l'utilisation d'un modèle déformable à base d'un algorithme génétique. [Rouet et al. 1997] en ont expérimenté une variante dont le but est de maintenir la diversité au sein de l'espace de recherche, et ils en ont démontré la robustesse.

### 5.3.7 Minima locaux et initialisation

Les méthodes d'optimisation locales ne garantissent pas de trouver le minimum global de  $f$ , il faut s'assurer par une bonne initialisation que l'on est dans le domaine de convergence de l'algorithme. Il est souvent possible en analyse d'images médicales d'avoir une initialisation grossière, mais convenable, par une intervention simple de l'utilisateur, ou par la connaissance de la géométrie des appareils d'acquisition.

Nous avons proposé une modélisation des modèles déformables en cinq composantes, et nous avons classé les modèles de la littérature selon ces composantes.

Les caractéristiques de liaison sont constituées de caractéristiques de bas niveau et de fonctions de liaison. Nous avons montré la diversité des caractéristiques de bas niveau en insistant sur la possibilité de les combiner. Nous avons étudié différentes solutions exactes ou approchées pour calculer une distance généralisée, qui est une fonction de liaison possible. Nous avons montré que les choix dans cette composante dépendent de la nature et de la qualité des images à traiter, ainsi que de l'application recherchée.

Nous avons proposé une classification des représentations géométriques qui distingue les modèles surfaciques, implicites et volumiques, en identifiant les différences fondamentales entre ces trois types de représentation.

Nous avons constitué une classification originale des déformations en déformations directes et indirectes, ces dernières se décomposant en déformations directes explicites et paramétriques. Le choix du mode de déformation dépendra de la représentation géométrique choisie, et de l'application à traiter.

Nous avons présenté différents dispositifs de déformation interactive, qui peuvent permettre d'intervenir de manière ergonomique dans l'évolution du modèle.

Nous avons également brièvement présenté différents algorithmes de minimisation pour le contrôle de cette évolution et nous avons mis en lumière le concept de déformabilité.

Après cette partie théorique, nous allons passer à la pratique, et à la description d'un premier modèle déformable, qui est un modèle surfacique.





## Deuxième partie

### Un modèle surfacique déformable



Nous considérons dans cette partie un modèle surfacique appelé  $\delta$ -snake. Il s'agit d'une surface active discrète dont le déplacement est guidé par un champ potentiel. Nous avons utilisé le  $\delta$ -snake pour reconstruire la surface d'un objet à partir d'un ensemble de points disposés sur cette surface. Nous appelons cet ensemble de points l'ensemble des *données*  $\mathcal{D}$ . Nous allons présenter le champ potentiel que nous avons mis en œuvre pour guider la déformation du modèle, et l'utilisation interactive que nous avons développée. Nous exposerons enfin des résultats de reconstruction sur le torse, le bras et la jambe d'un mannequin jouet dont les points des données viennent d'un capteur de distance, et sur une vertèbre, segmentée dans une image scanner.

### II●1.1 Le modèle des $\delta$ -snakes

Le  $\delta$ -snake est une surface maillée, composée uniquement de triangles, dont les sommets se déplacent par itérations dans  $\mathcal{R}^3$ . Les lois de déplacement choisies permettent de déplacer la surface vers une iso-potentielle *iso* d'un champ scalaire  $\phi$  de  $\mathcal{R}^3$  dans  $\mathcal{R}$  [Bainville 1992 ; Lachaud et Bainville 1994].

#### 1.1.1 Structure

Un  $\delta$ -snake est représenté par un ensemble fini  $S_t$  de points de  $\mathcal{R}^3$ , les sommets. Le paramètre  $t$  correspond au temps, représenté par le nombre d'itérations. A chaque sommet est associée la liste ordonnée et cyclique de ses sommets voisins. L'ordre des sommets dans la liste détermine une orientation locale de la surface, qui permet de définir globalement un intérieur à la surface. La topologie de l'intérieur ainsi que le nombre de ses composantes connexes peuvent être quelconques, car ces propriétés sont globales, et la structure est définie localement.

#### 1.1.2 Invariant géométrique

Après chaque itération, la triangulation doit vérifier la contrainte :

- **C1** : pour tout couple (AB) de sommets voisins,  $\delta < AB < 2.5\delta$

La valeur 2.5 a été déterminée de manière empirique pour assurer la stabilité du modèle lors de son évolution.

D'autres contraintes d'ordre topologique ont été ajoutées dans l'extension du modèle [Lachaud et Bainville 1994], nous n'en parlerons pas ici.

### 1.1.3 Préservation de l'invariant

Après le déplacement des sommets, un parcours de la surface permet de détecter les couples de sommets voisins qui ne respectent pas la contrainte (C1). Deux opérations permettent d'y remédier. Si deux voisins sont trop proches, ils sont fusionnés en un seul (figure II●1.1). Si deux voisins sont trop éloignés, un nouveau sommet est créé en leur milieu (figure II●1.2).

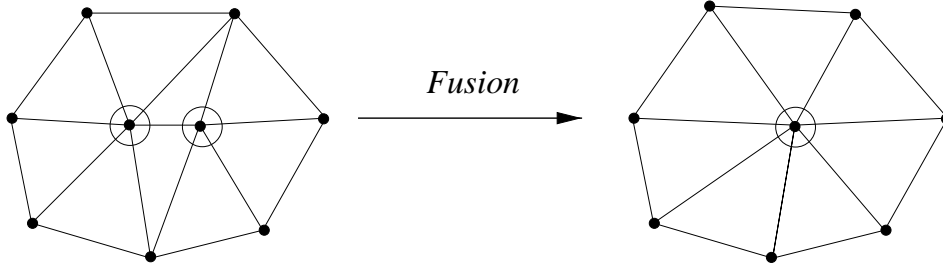


FIG. II●1.1: Fusion de deux voisins (cerclés) trop proches

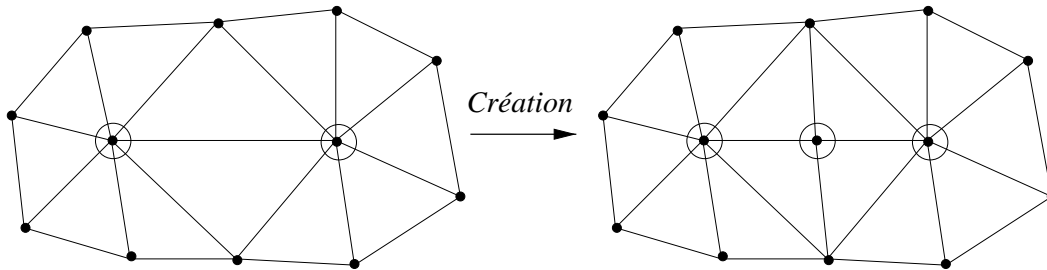


FIG. II●1.2: Création d'un sommet entre deux voisins (cerclés) trop éloignés

## II●1.2 Déformation

A chaque itération, les sommets sont déplacés simultanément. Pour chaque sommet  $\mathbf{x}$ , un vecteur  $\Delta \mathbf{x}$  est d'abord calculé

$$\Delta \mathbf{x} = \Delta \mathbf{x}_i + \Delta \mathbf{x}_e \quad (\text{II}●1.1)$$

$$\Delta \mathbf{x}_i = t(\mathbf{G} - \mathbf{x}) \quad (\text{II}●1.2)$$

$$\Delta \mathbf{x}_e = \text{sens} * (\phi(\mathbf{x}) - \text{iso}) * \delta * \mathbf{N} \quad (\text{II}●1.3)$$

La contribution interne  $\Delta \mathbf{x}_i$  au déplacement prend en compte la tension de la surface. La contribution externe  $\Delta \mathbf{x}_e$  a pour but de rapprocher la surface d'une iso-surface  $\phi(M) = \text{iso}$  du potentiel  $\phi$ .  $\mathbf{G}$  est l'isobarycentre des voisins de  $\mathbf{x}$ .  $\mathbf{N}$  est le vecteur unitaire normal à la surface en  $\mathbf{x}$ . *sens* vaut 1 ou  $-1$  suivant le cas où le snake se déplace depuis l'intérieur ou l'extérieur de la surface recherchée.

Puis chaque sommet est déplacé d'une quantité qui dépend du maximum de la norme des  $\mathbf{x}$ .

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \frac{\delta * \Delta \mathbf{x}}{6 * \max_{\mathbf{x} \in S_t} (\|\Delta \mathbf{x}\|)} \quad (\text{II}●1.4)$$

---

Le lecteur aura noté que le paramètre  $\delta$  qui régit les longueurs des arêtes intervient dans les équations de déformation de la surface comme un facteur d'échelle.

## II•1.3 Champ potentiel

Nous allons décrire dans cette section la méthode que nous avons choisie pour créer un potentiel scalaire  $\phi$  à partir d'un nuage de points. Nous utilisons la composition d'une fonction  $\psi$  par la distance  $dist_{\mathcal{D}}$  au nuage de points.

$$\phi(x, y, z) = \psi \circ dist_{\mathcal{D}}(x, y, z) \quad (\text{II}\bullet 1.5)$$

Nous présentons d'abord la carte de distance octree-spline qui nous sert pour stocker la distance de manière efficace, et discutons de la possibilité d'obtenir une distance signée. Puis nous décrivons la fonction  $\psi$  utilisée et nous discutons ce choix.

### 1.3.1 Carte de distance octree-spline

Il s'agit d'une structure de données qui permet de stocker de manière efficace la distance  $dist_{\mathcal{D}}$  au nuage de points. Elle a été proposée par Stéphane Lavallée et Richard Szeliski [Lavallée et al. 1991, 1996]. L'utilisation d'une octree, qui est une structure adaptative, a l'avantage de fournir plus d'informations près des points de données que loin de ceux-ci. La construction de la carte de distance associée à l'octree se fait en plusieurs étapes.

Il s'agit d'abord de construire l'octree associé à l'ensemble des points de données  $\mathcal{D}$ , en subdivisant récursivement le cube englobant les données en 8 sous-cubes ou nœuds, jusqu'à ce que le nœud considéré ne contienne plus de points de  $\mathcal{D}$  ou que la profondeur maximale de l'octree soit atteinte.

L'octree est ensuite raffiné, en effectuant des subdivisions supplémentaires pour que deux nœuds voisins aient un rapport de taille inférieur à un certain seuil.

La distance à  $\mathcal{D}$  des 8 sommets de chacun des nœuds terminaux est alors calculée et stockée. La distance d'un sommet  $\mathbf{c}$  à  $\mathcal{D}$  est la plus petite des distances de  $\mathbf{c}$  aux points de données.

Pour calculer la distance  $d(\mathbf{q}, \mathcal{D})$  d'un point  $\mathbf{q}$  quelconque à l'ensemble  $\mathcal{D}$ , le plus petit nœud de l'octree englobant  $\mathbf{q}$  est déterminé par une recherche binaire classique dans l'octree, puis  $d(\mathbf{q}, \mathcal{D})$  est calculée par interpolation trilineaire des distances stockées aux 8 sommets de ce nœud.

La figure II•1.3 montre à deux niveaux différents de profondeur l'octree-spline de distance obtenu pour les points du torse.

### 1.3.2 Distance signée

Si les données sont suffisamment denses, et ne comportent pas de lacunes, il est possible de déterminer l'intérieur et l'extérieur de l'objet qu'elles représentent. La carte de distance octree-spline est ainsi signée, avec la convention que la distance sera positive à l'extérieur et négative à l'intérieur. Cette opération consiste d'abord à donner un signe négatif à tous les sommets, puis à propager un signe positif depuis l'extérieur, en s'arrêtant aux nœuds de profondeur maximale contenant des points.

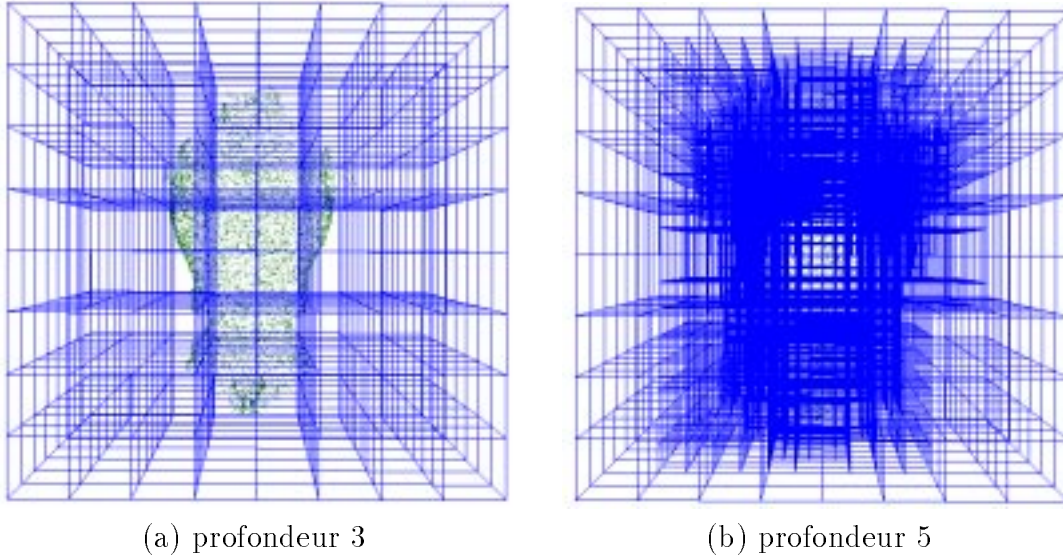


FIG. II.1.3: Octree-spline de distance du torse

Dans les exemples que nous présentons, seule la carte de distance de la vertèbre est signée.

### 1.3.3 Fonction $\psi$

Dans notre problématique de reconstruction d'une surface à partir de points nous désirons créer un champ potentiel à partir uniquement des points. Ce champ potentiel doit définir l'iso-surface qui sera approchée par le  $\delta$ -snake.

En utilisant la carte de distance, nous obtenons une représentation de cette iso-surface comme l'ensemble des points ayant pour valeur zéro. Mais si l'on cherche à atteindre ces points par une méthode d'approximation, la solution sera instable dans le cas où la carte de distance n'est pas signée, car la fonction ne change pas de signe au passage du zéro.

Pour obtenir une convergence stable, nous avons fixé la valeur *iso* à atteindre non pas à 0, mais à une valeur  $y_1$ , que nous avons choisie égale à 0.5 dans nos exemples. De cette manière, le  $\delta$ -snake va arrêter son parcours en deça du lieu réel de la surface recherchée, mais à une distance fixée égale à  $x_1$ . Il suffit ensuite de déplacer chaque sommet de cette distance suivant la normale à la surface pour reconstruire la surface désirée de l'objet.

Une autre façon de procéder, que nous indiquons comme une extension, serait de prendre en compte le signe de la dérivée du potentiel suivant la normale au  $\delta$ -snake pour déterminer si le sommet a ou non dépassé le lieu qu'il doit atteindre.

Pour nos expérimentations, nous avons construit et utilisé une fonction  $\psi$ , pour définir la valeur à atteindre (équation II.1.6 et figure II.1.4).

$$\psi(x) = \begin{cases} x^2 \left( x \frac{dx_1 - 2(y_1 - y_0)}{x_1^3} + \frac{3(y_1 - y_0) - dx_1}{x_1^2} \right) + y_0 & \text{si } x < x_1 \\ \frac{-y_1^2}{d(x - (x_1 + \frac{y_1}{d}))} & \text{sinon} \end{cases} \quad (\text{II.1.6})$$

Nous avons ensuite mis au point une autre fonction qui remplace la fonction  $\psi$  précédente, et qui a l'avantage de ne pas déplacer spatialement l'iso-surface à atteindre. Nous

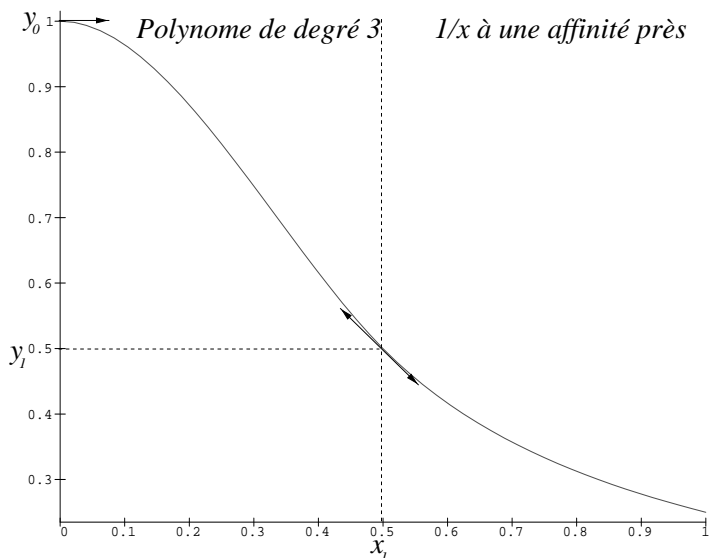


FIG. II●1.4: Fonction  $\psi$

utilisons le signe de la variation de la fonction distance au cours du déplacement des sommets du  $\delta$ -snake.

### 1.3.4 Représentation du champ potentiel

La figure II●1.5 montre deux coupes dans le champ potentiel du torse.

### 1.3.5 Visualisation de l'iso-surface

Une autre forme de représentation de ce champ potentiel est d'en calculer une iso-surface à l'aide de l'algorithme des *marching cubes*. Nous avons utilisé un module AVS approprié, et montrons les résultats obtenus pour le torse (figure II●1.6) et la vertèbre (figure II●1.7). Ce procédé nécessite le calcul du champ en tous les voxels d'un volume discret, que nous avons choisi de taille  $128 \times 128 \times 128$ .

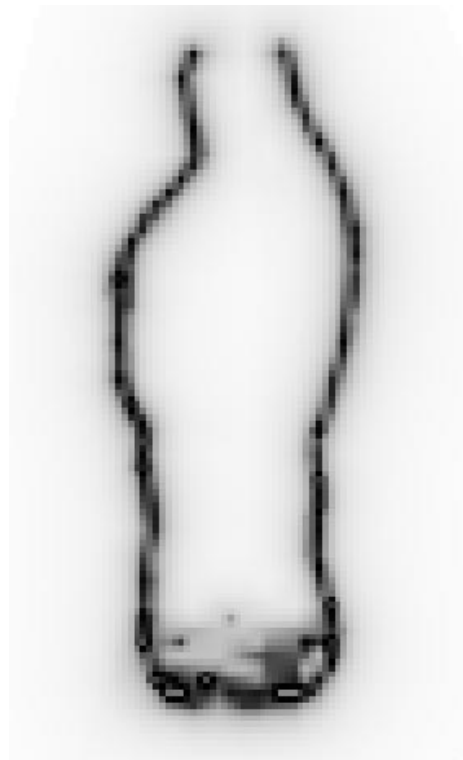
## II●1.4 Apport de l'interactivité

Nous allons voir dans le chapitre suivant pourquoi et comment nous avons mis en place deux procédés d'action interactive sur la surface  $\delta$ -snake.





(a) coupe frontale



(b) coupe sagittale

FIG. II●1.5: Deux coupes dans le champ potentiel du torse.



(a) vue de face



(b) vue de profil



(c) vue de dos

FIG. II●1.6: Iso-surface du torse dans un volume de  $128 \times 128 \times 128$  éléments

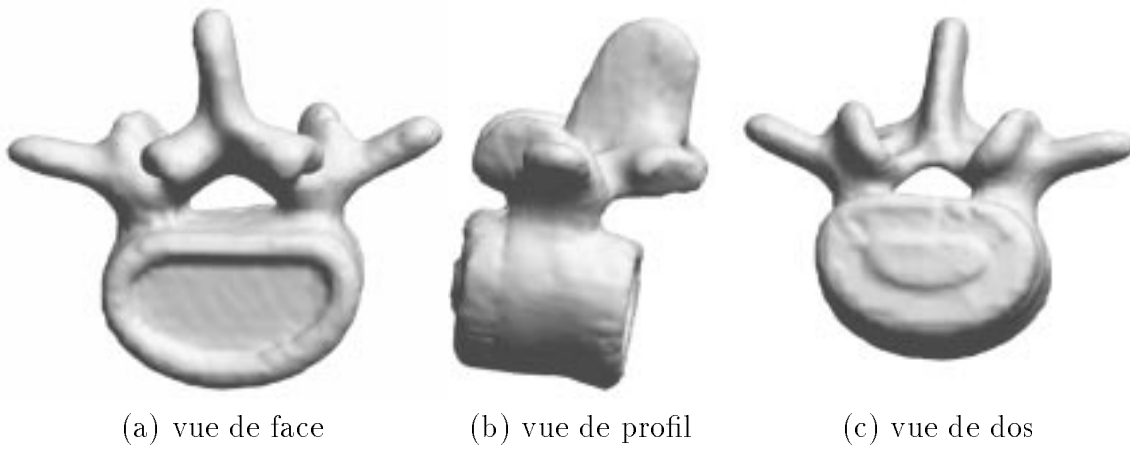


FIG. II●1.7: Iso-surface de la vertèbre dans un volume de  $128 \times 128 \times 128$  éléments

Si les données sont incomplètes, comme celles du torse (voir figure II●1.6), la surface snake passera au travers. Un moyen possible pour y remédier est d'imposer à la surface une rigidité importante, mais cela empêche alors la reconstruction satisfaisante des détails de l'objet. Une autre voie est celle que nous avons explorée dans ce chapitre, celle de l'interactivité.

Il y a deux façons d'interagir avec le modèle, soit de placer et de déplacer dans la scène des objets qui modifient la géométrie ou l'évolution du modèle, soit d'agir directement sur sa surface. Nous avons exploré ces deux possibilités.

## II●2.1 Modélisation de l'environnement

Notre première idée a été de disposer dans la scène des objets qui interagissent avec la surface  $\delta$ -snake. Ces objets sont à disposer au milieu des données comme des rustines sur une chambre à air. Nous avons opté pour un paradigme d'interaction très simple : tout sommet qui pénètre dans un de ces objets est replacé en dehors.

Nous avons choisi des objets élémentaires : des sphères. Un sommet du  $\delta$ -snake qui pénètre dans une sphère intraversable est replacé à la surface de la sphère, en suivant le rayon correspondant à la direction de pénétration (voir figure II●2.1).

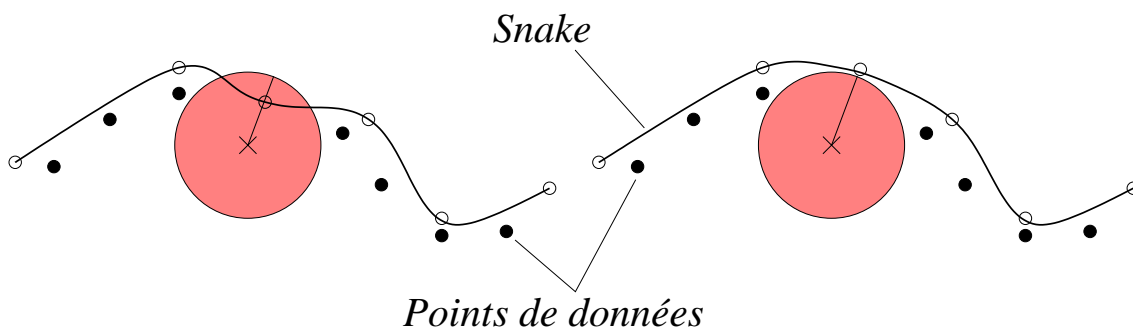


FIG. II●2.1: Illustration de l'action d'un objet intraversable

La figure II●2.2 montre les sphères que nous avons placées parmi les points de données du torse afin de combler les lacunes des données, ainsi que la surface correspondante.

La figure II●2.3 montre ce que devient la surface si l'on supprime deux des sphères au niveau du cou.

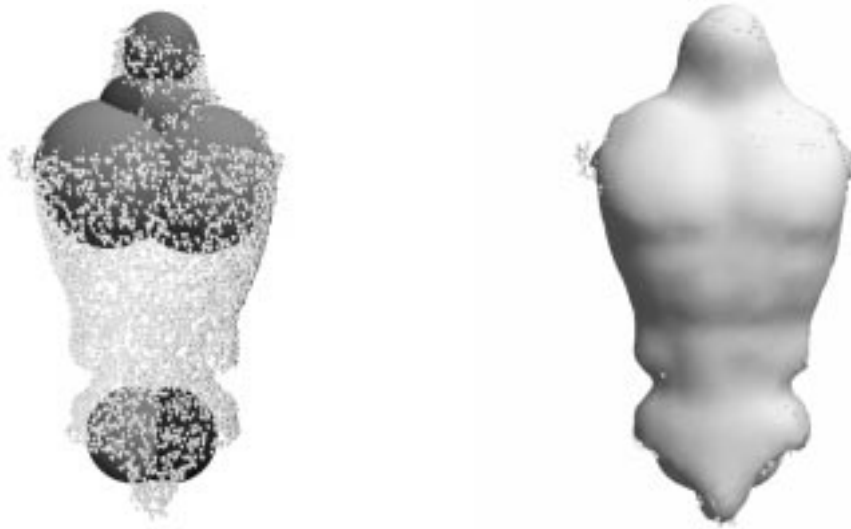


FIG. II●2.2: Ajout de sphères pour combler les lacunes des données.

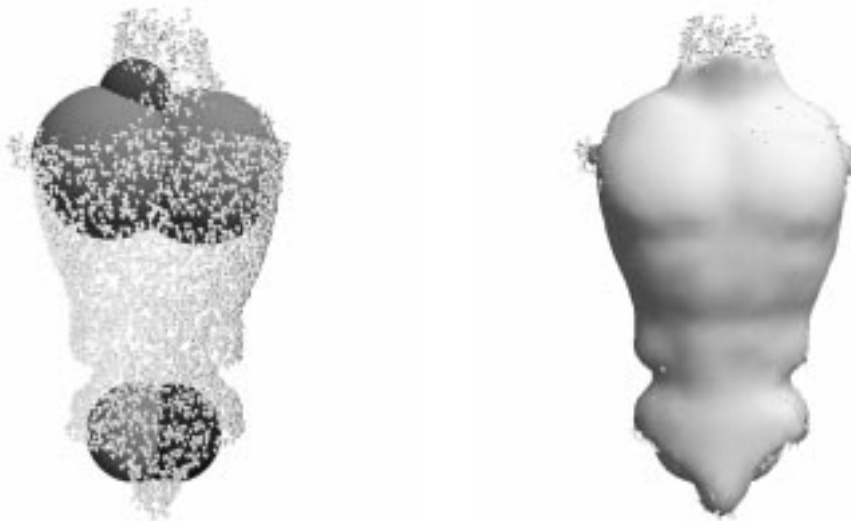


FIG. II●2.3: Suppression de deux sphères au niveau du cou.

## II●2.2 Déformation interactive

Le deuxième mode d'interaction de l'utilisateur avec la surface  $\delta$ -snake est la sculpture interactive de cette surface. Nous avons implémenté un outil de manipulation qui permet de sélectionner un sommet du  $\delta$ -snake, et de le déplacer interactivement, ainsi qu'une portion de la surface au voisinage de ce sommet.

### 2.2.1 Sélection d'un sommet

Lorsqu'un sommet est sélectionné, un manipulateur est affiché à la même position que ce sommet. Nous avons choisi une forme simple : le tétraèdre. La figure II●2.4 montre ce manipulateur, dans l'image (b) à l'extrémité de l'oreille et dans l'image (c) à l'extrémité

du museau.

## 2.2.2 Déplacement du sommet sélectionné

Comme nous utilisons la souris pour définir le mouvement du sommet sélectionné, nous avons choisi d'opérer le déplacement dans un plan parallèle à l'écran. Cela permet une bonne précision dans la désignation du lieu de destination du sommet. C'est pour cette raison que pour modeler le museau du renard, nous avons tourné la tête à  $90^\circ$  (figure II•2.4 (c)).

## 2.2.3 Déplacement d'une partie de la surface

Notre interface permet également de choisir l'étendue de la surface qui doit se déplacer avec le sommet sélectionné. Les sommets voisins se déplacent d'une longueur qui dépend de leur distance avec la position initiale du sommet sélectionné. L'oreille du renard (figure II•2.4 (b)) a été sculptée avec une étendue de 1 : les voisins immédiats du sommet sélectionné sont également déplacés. Le museau a été sculpté avec une étendue de 2 (figure II•2.4 (c)) : les voisins des voisins du sommet sélectionné sont aussi déplacés. Les facettes dont les sommets vont être déplacés sont affichées en rouge dans notre modèle (gris sombre sur la figure II•2.4).

Le déplacement des points voisins du sommet sélectionné se font dans la même direction que le déplacement de celui-ci, et l'amplitude de leur déplacement est une fonction décroissante de leur distance à ce sommet.

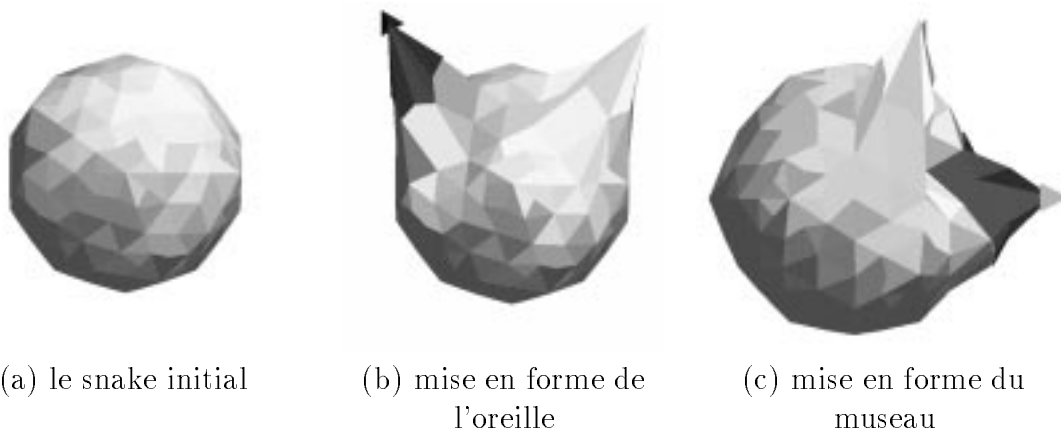


FIG. II•2.4: Modélisation d'une tête de renard

## II•2.3 Comparaison

Dans le cadre de la reconstruction de surface, l'utilisation d'objets intraversables s'est révélée très efficace pour combler les lacunes des données. Nous avons ainsi pu remodeler en quelque sorte les données à reconstruire.

La déformation interactive s'intègre mieux dans des applications de sculpture d'une surface. D'autres outils pourraient cependant lui être ajoutés comme la possibilité de

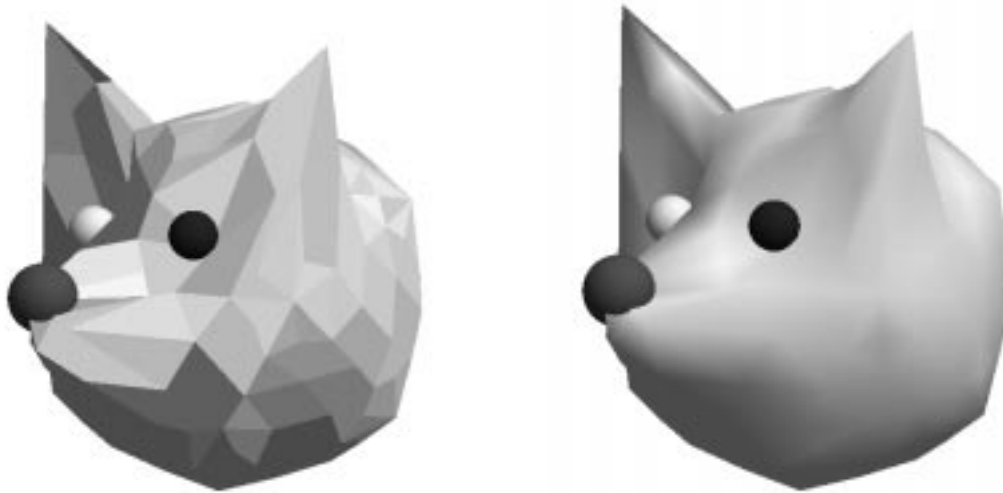


FIG. II●2.5: Vue finale du renard

fixer la position de certains points dans l'espace, qui permettraient alors d'utiliser cette déformation interactive dans le processus de reconstruction.

Nous présentons dans ce chapitre les résultats de reconstruction de la surface d'objets à l'aide de  $\delta$ -snakes, à partir de points disposés sur cette surface.

## II●3.1 Présentation de l'application

Nous avons développé notre application sous la forme de modules AVS. Un premier module (*octree pot*) lit le nuage de points et calcule le champ potentiel dans lequel va évoluer le  $\delta$ -snake. Un second module (*delta\_snake*) gère l'évolution du snake. La visualisation se fait grâce à un module d'AVS (*geometry viewer*).

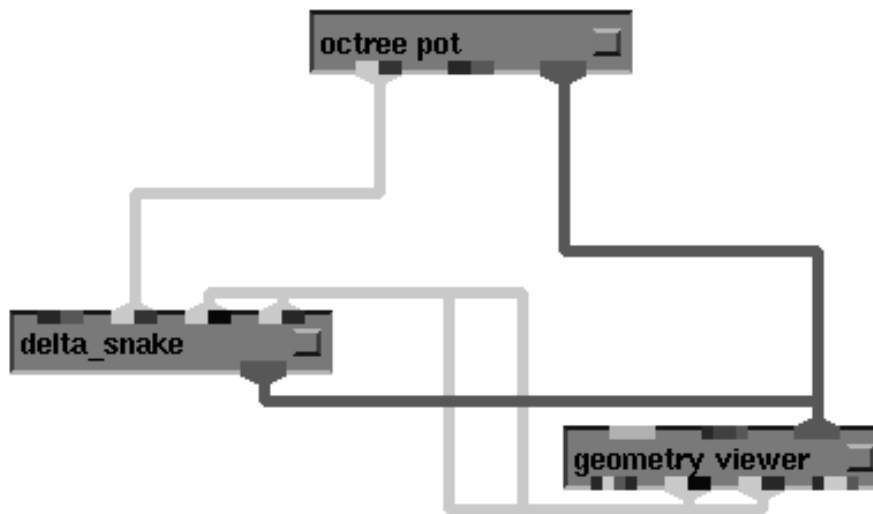


FIG. II●3.1: Le réseau AVS de l'application.

## II●3.2 Resultats

Les données que nous avons utilisées proviennent pour le torse, le bras et la jambe d'une numérisation d'un mannequin à l'aide d'un capteur de distance. Pour chaque membre,

une dizaine de vues ont été acquises tout autour de l'objet. Ces vues étaient constituées de nuages de points se recouvrant partiellement. Elles ont été recalées entre elles pour constituer un nuage assez dense de points de la surface de l'objet [Bittar et al. 1993]. Les données de la vertèbre proviennent de la segmentation manuelle d'images TDM d'un moulage de vertèbre. Les points désignés sur chaque coupe TDM ont été interpolés par une méthode fondée sur la forme (*shape based interpolation* [Raya et Udupa 1990]) afin de reconstituer une répartition homogène des points sur la surface.

Les temps de convergence sur une station DEC alpha 3000 sont compris entre 3 et 5 minutes pour les exemples présentés.

La surface initiale qui a permis de reconstruire la vertèbre (figures II●3.2, II●3.3 et II●3.4) est un tore dont les paramètres initiaux et la position initiale sont déterminés interactivement. L'évolution dans le champ potentiel créé par le nuage de points de la vertèbre est ensuite automatique.

Le bras (figure II●3.6) et la jambe (figure II●3.7) sont reconstruits à partir d'un icosaèdre initial positionné interactivement à l'intérieur des objets.

Le torse (figure II●3.5) est reconstruit à partir d'un cube englobant le nuage de points, et se réduisant, alors que le mouvement est un gonflement dans les autres exemples.

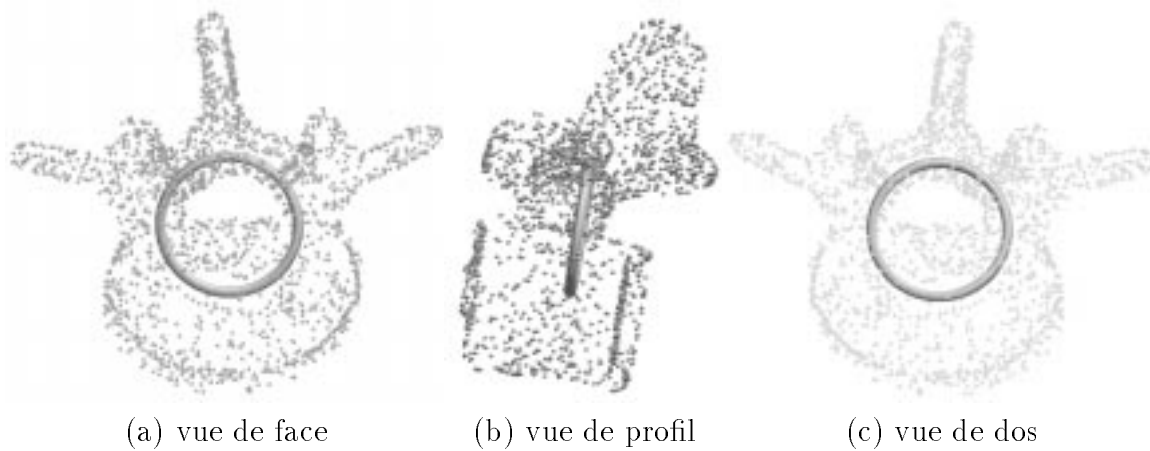


FIG. II●3.2: Reconstruction d'une vertèbre : Etape initiale

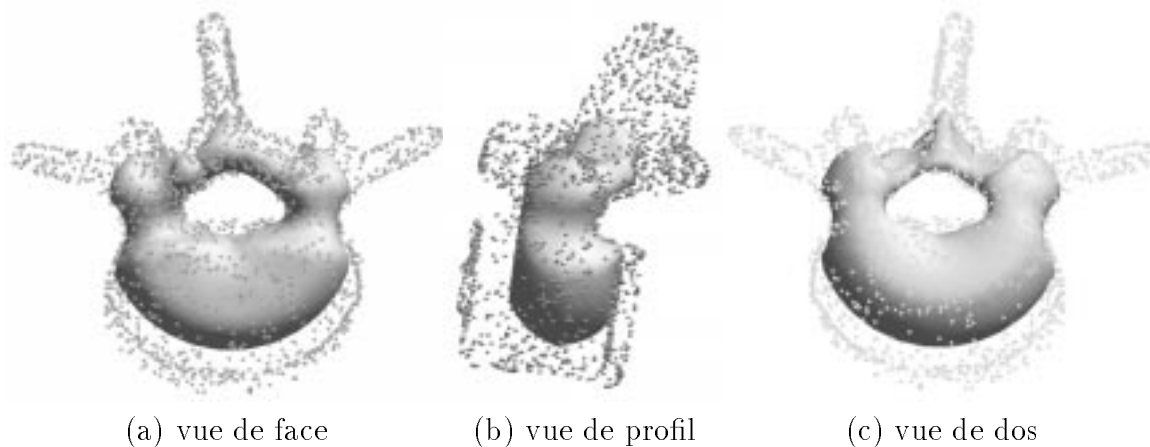


FIG. II●3.3: Reconstruction d'une vertèbre : Etape intermédiaire



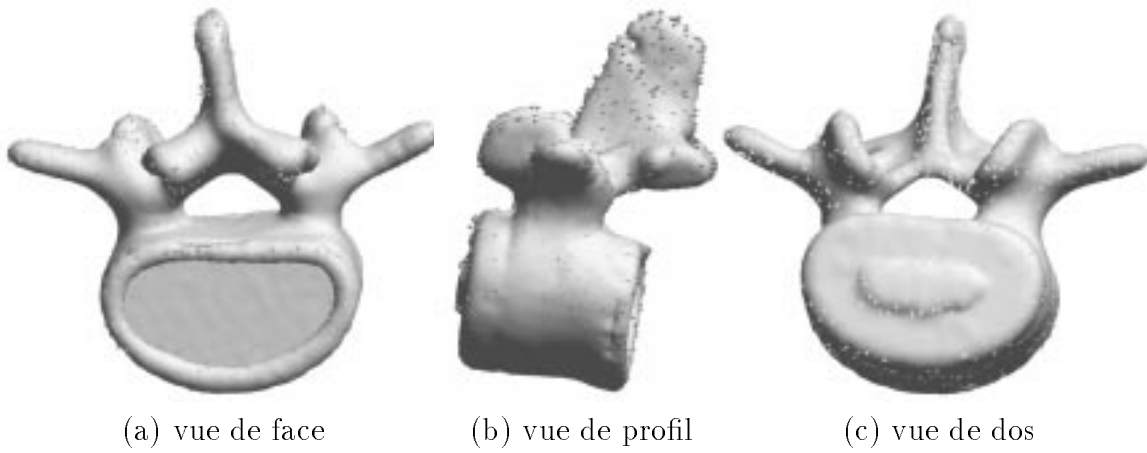


FIG. II●3.4: Reconstruction d'une vertèbre : Etape finale

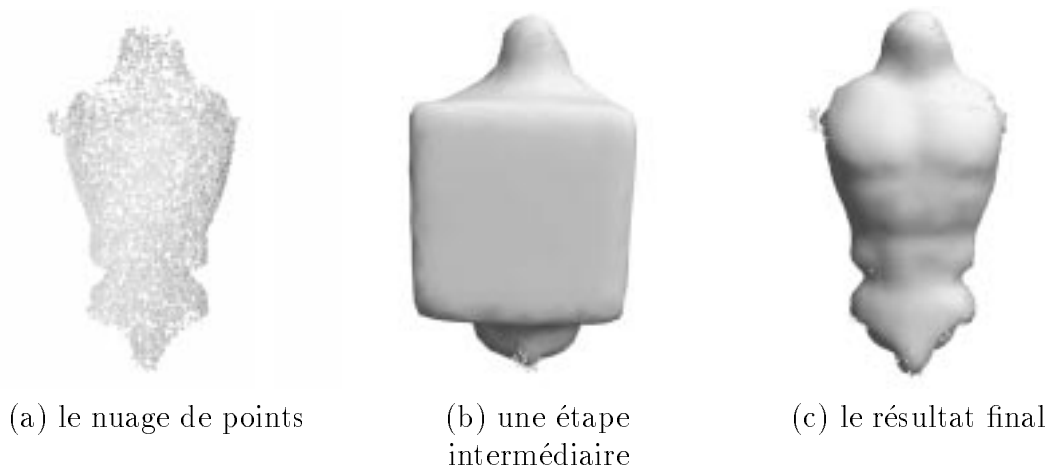


FIG. II●3.5: Reconstruction d'un torse



(a) le nuage de points



(b) la surface snake  
vue de dessus



(c) la surface snake  
vue de dessous

FIG. II●3.6: Reconstruction d'un bras



(a) le nuage de points



(b) la surface snake et les  
points des données



(c) la surface snake  
seule

FIG. II●3.7: Reconstruction d'une jambe

Le modèle des  $\delta$ -snakes fournit une représentation surfacique régulière d'un objet, qui peut à son tour constituer la base d'un modèle déformable. Ainsi Emmanuel Promayon [Promayon et al. 1997 ; Promayon 1997] a utilisé le modèle  $\delta$ -snake du torse pour constituer un modèle à mémoire de forme des parois abdominales qui permet de simuler la respiration.

Nous avons montré comment ce modèle peut être utilisé pour reconstruire la surface d'un objet à partir de points disposés sur cette surface. Pour y parvenir, nous avons d'abord calculé une carte de distance à cette surface, en utilisant les octree splines. Cette représentation permet de reconstruire la surface à l'aide d'algorithmes classiques comme les *marching cubes*. Mais si les données sont incomplètes, de tels algorithmes ne permettront pas de retrouver la forme initiale de l'objet. En revanche nous montrons qu'en disposant interactivement des objets spécifiques au voisinage de la surface, la continuité de celle-ci est facilement reconstruite.

Le  $\delta$ -snake en tant que surface déformable apporte ses caractéristiques topologiques à la reconstruction. La surface résultante possède la topologie désirée, et est régulièrement maillée. Nous avons donné des exemples de surface homotopique à une sphère et également à un tore. La surface déformable peut également être manipulée interactivement, à travers une interface que nous avons réalisée.

La structure géométrique et les lois d'évolution sont très simples. De plus sa déformation est directe au sens énoncé dans la première partie, et locale. Tous ces éléments font du modèle des  $\delta$ -snakes un modèle déformable dont l'évolution est très rapide. En ajoutant les propriétés de déformation interactives, on obtient un ensemble permettant rapidement de reconstruire des objets à partir de points de leur surface.

Ce modèle ne possède cependant pas de forme propre, il n'est pas conçu pour se déformer à partir d'une forme de référence et dans des limites imposées par une déformabilité.

Alors que le modèle des  $\delta$ -snakes peut reconstruire des objets de topologie complexe en partant d'une surface de même topologie, nous allons voir dans la partie suivante un modèle déformable implicite, qui permet de reconstruire une surface de topologie a priori inconnue.

Troisième partie

Un modèle implicite déformable



Nous décrivons dans cette partie un modèle déformable qui est composé d'une surface implicite générée par des primitives. Ce modèle est adapté à des objets fermés de topologie complexe et inconnue a priori.

Dans les travaux que nous avons menés avec ce modèle, nous n'avons pas eu à mettre en œuvre de pré-traitement visant à extraire des points ayant des caractéristiques particulières, car les données étaient déjà présentes sous la forme d'un ensemble de points répartis sur la surface de l'objet. Ce modèle peut bien sûr s'appliquer à d'autres images issues de modalités classiques en imagerie médicale, pourvu que par une pré-segmentation adéquate on puisse se ramener à un tel ensemble de points.

Le pré-traitement que nous avons développé est le calcul de l'axe médian des données. Nous en exposerons la méthode dans le chapitre III•4 page 89, mais précisons d'abord le cadre de notre approche.

### III•1.1 Reconstruction de la surface d'un objet

D'autres méthodes que les modèles déformables peuvent être employées pour reconstruire la surface d'un objet à partir d'un ensemble de points répartis sur sa surface. Nous en présentons quelques unes dans ce paragraphe, avant de décrire notre travail.

Hoppe a introduit dans [Hoppe et al. 1992] une technique de *traversée de graphe* pour approximer le plan tangent à la surface en chaque point des données. Pour chaque point de l'espace, il calcule une distance signée au plan tangent le plus proche, qui sert d'approximation linéaire locale de la surface à reconstruire. Il définit cette surface à reconstruire comme l'ensemble des zéros de la fonction de distance, et il calcule une représentation polygonale par une technique de partitionnement spatial. Dans [Hoppe et al. 1994], l'auteur complète sa méthode par une optimisation de la surface qui est découpée en des approximations lisses par morceaux. Il obtient ainsi une représentation plus précise et une meilleure reconstruction des parties anguleuses. La méthode de reconstruction de Boissonnat [Boissonnat 1984 ; Boissonnat et Geiger 1992] vient de la triangulation de Delaunay de l'ensemble des points des données. Edelsbrunner et Mücke [Edelsbrunner et Mücke 1992] introduisent un paramètre  $\alpha$  qui règle le degré de détail de la forme qu'ils extraient de la triangulation de Delaunay. Attali [Attali et al. 1994] calcule le graphe de Voronoï des points, afin de construire le squelette de l'objet, et de reconstruire sa forme. La limite principale de ces méthodes est qu'elles ne sont pas adaptées aux ensembles de points non structurés, ou aux nuages de points bruités.

## III•1.2 Reconstruction de forme à l'aide de primitives implicites

Une forme reconstruite à l'aide de méta-balles est composée d'un ensemble de primitives, qui ont chacune leur position dans l'espace et des valeurs particulières de leurs paramètres. Le nombre de ces primitives n'est pas connu a priori, et la première heuristique de reconstruction, celle que Muraki a proposée (voir la section 3.2.3 page 35), comprend l'initialisation à l'aide d'une unique primitive, qui est ensuite subdivisée dans un processus qui génère de plus en plus de primitives. Le modèle croît de la même manière que le modèle déformable géométriquement de Miller (voir en 3.1.1 page 31).

### 1.2.1 Une première approche semi-automatique

Nous avons mis en œuvre une méthode semi-automatique de reconstruction avec des surfaces implicites en collaboration avec Nicolas Tsingos et Marie-Paule Gascuel [Tsingos et al. 1995] de l'équipe iMAGIS, dans laquelle nous conservons l'heuristique de la croissance de modèle. Nous utilisons un critère de sélection de la primitive à subdiviser fondé sur l'énergie locale des primitives, qui améliore l'efficacité de l'algorithme.

L'utilisateur positionne les primitives initiales, ce qui permet de traiter des objets de topologie et de géométrie complexes. De plus l'utilisateur place des "fenêtres de reconstruction", ce sont des parallélépipèdes qui se recouvrent légèrement, et permettent de définir des zones de l'objet dans lesquelles la reconstruction se fait quasi indépendamment. Nous détaillons les idées originales de cette approche dans le chapitre III•2, et nous parlerons plus particulièrement de son aspect interactif dans le chapitre III•3.

Notre expérience de cette méthode nous a montré que l'utilisateur passait beaucoup de temps à disposer les primitives initiales et les fenêtres de reconstruction. De plus, il devait souvent recommencer lorsque les conditions initiales qu'il avait choisies ne s'avéraient pas adaptées.

C'est pourquoi nous avons cherché à créer une initialisation automatique. C'est l'axe médian que nous avons choisi d'utiliser pour cela.

### 1.2.2 Combiner un axe médian et des surfaces implicites

Les iso-surfaces générées par des primitives sont très pratiques pour la reconstruction, car elles fournissent un moyen simple de définir une énergie, qui donne une notion de la distance aux données. De plus, elles forment une surface souple, et représentent l'objet d'une manière très compacte, puisqu'il suffit de stocker les positions des primitives, et les paramètres des champs potentiels associés.

Or, l'initialisation des primitives n'est pas satisfaisante, et en plus, le choix de partager des primitives existantes en deux n'est pas efficace, car les primitives filles sont toutes deux placées à l'ancienne position de leur mère, ce qui amène le processus d'optimisation à les déplacer d'une distance non négligeable.

Ainsi, nous présentons une méthode qui utilise l'axe médian des données pour calculer l'ensemble des primitives initiales, et progressivement réduire cet ensemble. Les travaux concernant cette méthode ont été publiés dans [Bittar et al. 1995]. C'est le deuxième volet de notre collaboration avec Nicolas Tsingos et Marie-Paule Gascuel. L'heuristique

---

utilisée sélectionne itérativement les primitives qui servent à la reconstruction de l'objet dans l'ensemble des sphères de l'axe médian. Nous procédons en deux étapes :

- Nous calculons d'abord un axe médian des points des données, et nous accordons sa résolution pour obtenir un nombre de sphères approprié. Cette étape prend en compte la topologie de l'objet, et fournit un ensemble de sphères qui sont candidates à devenir des primitives implicites. Le centre d'une sphère de l'axe médian devient alors la position de la primitive, et le rayon de la sphère donne une information sur les paramètres de la fonction potentielle.
- Afin de reconstruire une surface lisse et d'utiliser peu de primitives, nous utilisons un processus itératif, qui sélectionne progressivement les primitives parmi les sphères de l'axe médian, et optimise leurs paramètres pour ajuster la surface implicite aux données. L'heuristique qui permet de choisir rapidement parmi les sphères de l'axe médian celle qui sera ajoutée à l'ensemble des primitives utilise un critère qui est fondé sur le caractère local des fonctions potentielles.

### III•1.3 Plan de la partie III

Nous avons déjà présenté la bibliographie des surfaces implicites. Nous discuterons dans la suite de cette partie, de la construction du modèle en fonction des données. Nous avons expérimenté deux méthodes. La première (chapitre III•2) est une approche semi-automatique dans laquelle les primitives sont positionnées interactivement. La deuxième est une approche automatique, qui utilise l'axe médian discret (cette notion est décrite au chapitre III•4) de l'ensemble de points des données. Nous montrerons comment l'axe médian peut être employé pour initialiser les positions et les paramètres des primitives qui génèrent l'iso-surface.

Le chapitre III•5 présente l'algorithme itératif qui ajoute progressivement des primitives afin d'obtenir une reconstruction précise, tout en limitant le nombre de ces primitives. Nous présenterons dans le chapitre III•3 les résultats obtenus par l'approche semi-automatique, et dans le chapitre III•6 les résultats de l'approche automatique.



La raison pour laquelle nous avons étudié les modèles implicites à base de primitives était la recherche d'une manière de modéliser des objets en leur associant des caractéristiques physiques. Les primitives à l'intérieur des objets répondent bien à ces critères et sont utilisées en animation [Cani-Gascuel et Desbrun 1997]. Un moyen de modéliser un objet est de le créer de toutes pièces à l'aide d'un logiciel spécialisé, l'alternative étant de reconstruire la forme d'un objet existant.

Nous allons montrer comment notre première approche intègre ces deux moyens. Commençons par détailler les propriétés des primitives que nous avons utilisées, en particulier leurs fonctions de champ.

### III•2.1 Définition des fonctions de champ locales

Des fonctions de champ locales, qui s'annulent ainsi que leurs dérivées à une certaine distance  $R$  de l'origine ont été introduites pour optimiser le calcul des champs pour la construction d'objets complexes [Wyvill et al. 1986]. Un autre avantage de telles fonctions est qu'elles permettent un contrôle local de la surface implicite, ce qui est particulièrement important pour le processus de reconstruction, puisqu'ainsi, l'optimisation des primitives dans une partie de l'objet ne dégrade pas la reconstruction des données dans des zones éloignées. Comme nous le montrons dans le paragraphe suivant, l'utilisation de fonctions locales simplifie la définition d'un critère local de sélection pour définir une surface implicite à partir de l'axe médian.

Les fonctions de champ choisies sont amenées à être utilisées dans une minimisation coûteuse. Nous avons donc choisi un modèle qui est contrôlé par un nombre réduit de paramètres (afin de limiter la dimension de l'espace de recherche), et qui utilise des fonctions rapides à calculer. Ainsi, nous présentons des fonctions de champ  $f_i$  qui sont composées d'un segment de droite et d'un morceau de quadrique. Nous n'avons que deux paramètres. (voir Figure III•2.1 page 81) :

- Le rayon  $e_i$  de la sphère créée par une primitive  $P_i$  seule ( $e_i$  est tel que  $f_i(e_i) = iso$ ) ;
- La raideur  $k_i$  en  $e_i$  (dérivée en  $e_i$  de  $f_i$ ), qui définit les propriétés de la surface de se mélanger avec d'autres.  $k_i$  peut aussi servir pour régler le lissage de la surface.

Le paramètre  $R_i$  est fixé, une fois  $e_i$  et  $k_i$  donnés.

Contrairement à Muraki, qui utilise des fonctions de champ positives et négatives, nous n'utilisons que des champs positifs. Cela semble plus approprié pour des applications qui peuvent comprendre une simulation physique de la déformation des objets reconstruits, comme avec la méthode présentée en [Gascuel 1993].

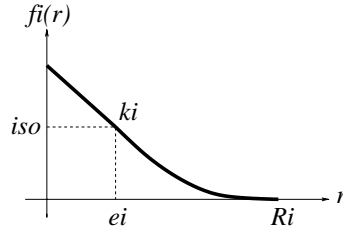


FIG. III•2.1: Fonction de champ locale avec deux paramètres.

L'expression de la fonction est :

$$f_i(M) = \begin{cases} -k_i r + k_i e_i + 1 & \text{si } r \in [0, e_i] \\ \frac{k_i e_i (e_i - R_i) + 3e_i - R_i}{(e_i - R_i)^3} (r - R_i)^2 & \text{si } r \in [e_i, R_i] \\ 0 & \text{sinon} \end{cases}$$

Où  $r = d(M, P_i)$  et  $R_i = (e_i - \frac{2}{k_i})$ , c'est le rayon d'influence.

Pour chaque primitive, nous n'avons donc que 5 paramètres à optimiser :  $e_i, k_i, x_i, y_i, z_i$ .

## III•2.2 Bases de la reconstruction semi-automatique

Nous avons choisi de garder les principes posés par Muraki [Muraki 1991], tout en apportant des modifications. Expliquons-en les raisons :

Premièrement, nous avons opté pour une initialisation interactive des primitives servant de base au processus. L'utilisateur s'assurant qu'il a bien placé ces primitives à l'intérieur de l'objet à reconstruire, la nécessité de tenir compte des normales dans la reconstruction n'est plus de mise. En effet les normales permettaient à Muraki de s'assurer que le volume reconstruit était bien le volume interne. Nous définissons donc l'énergie à minimiser comme suit.

$$E = \frac{1}{n_{\text{données}}} \left( \sum_{j=1}^{n_{\text{données}}} (f(D_j) - iso)^2 \right) + \frac{1}{n_{\text{prim}}} \left( \alpha_1 \sum_{i=1}^{n_{\text{prim}}} e^{-\beta_1 e_i} + \alpha_2 \sum_{i=1}^{n_{\text{prim}}} e^{-\beta_2 k_i} \right) \quad (\text{III•2.1})$$

$n_{\text{prim}}$  est le nombre de primitives implicites courant. Les deux derniers termes empêchent les primitives de dégénérer vers des paramètres de rayon ou de raideur négatifs.

L'apport capital de notre approche est lié à l'utilisation de fonctions de champ locales, qui permettent de s'affranchir du lourd processus de sélection de la primitive suivante à subdiviser qui nécessitait dans la méthode de Muraki de tester systématiquement toutes les primitives existantes.

## III•2.3 Un algorithme de subdivision efficace

Comme nous avons une fonction de champ locale, nous pouvons calculer la qualité de la reconstruction localement autour de chaque primitive, dans ce que nous appelons sa

*zone d'influence*. Il s'agit de la portion de l'espace autour de la primitive pour laquelle sa fonction de champ est non nulle. La meilleure primitive candidate à être subdivisée sera la primitive dans la zone d'influence de laquelle la reconstruction sera la moins bonne. Nous avons choisi pour caractériser la qualité de la reconstruction dans la zone d'influence d'une primitive  $I_i$  le critère suivant :

$$C(I_i) = \left( \sum_{j=1}^{m_i} (f(D_{j,i}) - iso)^2 \right) \quad (\text{III}\bullet 2.2)$$

La somme est faite sur les  $m_i$  points des données  $D_{j,i}$  qui sont à l'intérieur de la zone d'influence de  $I_i$ . A chaque itération, la primitive dont la valeur de  $C_i$  est la plus grande est remplacée par deux primitives, dont les 10 paramètres sont optimisés seuls dans un premier temps, en gardant les paramètres des autres primitives constants. Nous lançons ensuite quelques itérations d'optimisation de tous les paramètres de toutes les primitives, pour réduire l'énergie totale et le nombre total de primitives qui sera utilisé.

### III•2.4 Fenêtres de reconstruction

Pour éviter de prendre en compte tous les points de données à chaque étape de la minimisation, nous avons défini le concept de *fenêtres de reconstruction*, dans lesquelles la reconstruction se déroule indépendamment. Dans chacune de ces fenêtres, l'énergie sera minimisée uniquement à partir des points de donnée de la fenêtre courante. De plus l'étape globale d'optimisation après une subdivision ne s'effectuera que sur les primitives de cette fenêtre. Ce procédé de reconstruction locale est justifié par notre utilisation de fonctions de champ locales : lorsqu'une primitive se déplace, seuls les points de sa zone d'influence sont affectés, il n'est donc pas utile de recalculer l'énergie de tout l'objet. La figure III•2.2 montre un exemple de définition de trois fenêtres de reconstruction.

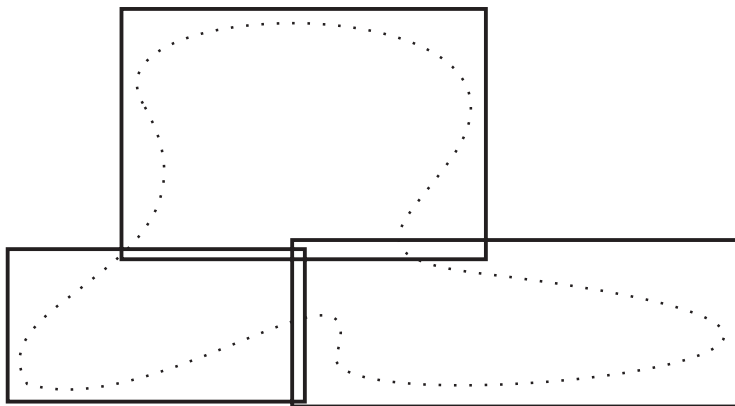


FIG. III•2.2: Définition de plusieurs fenêtres pour une reconstruction de forme locale

L'algorithme que nous utilisons pour reconstruire les données en présence de plusieurs fenêtres de reconstruction est défini comme suit. A chaque étape de la reconstruction :

1. Sélectionner la fenêtre pour laquelle

$$W_k = \frac{1}{w_k} \left( \sum_{j=1}^{w_k} (f(D_j) - iso)^2 \right) \quad (\text{III}\bullet 2.3)$$

est le plus grand ( $w_k$  est le nombre de points des données dans cette fenêtre).

2. Utiliser le critère III•2.2 pour sélectionner la primitive à subdiviser dans cette fenêtre.
3. Subdiviser cette primitive et optimiser ses deux “enfants”, en gardant les paramètres de toutes les autres primitives constants. L'énergie à minimiser est calculée à partir des  $w_k$  points et des  $n_i$  primitives de la fenêtre courante.

$$E_k = \frac{1}{m_k} \left( \sum_{j=1}^{m_k} (f(D_j) - iso)^2 \right) + \frac{1}{n_k} \left( \alpha_1 \sum_{i=1}^{n_k} e^{-\beta_1 e_i} + \alpha_2 \sum_{i=1}^{n_k} e^{-\beta_2 k_i} \right) \quad (\text{III}\bullet 2.4)$$

4. Calculer quelques itérations de minimisation globale dans la fenêtre pour réduire  $E_k$  (et  $W_k$ ).
5. Si la valeur globale définie par  $W_{global} = \sum W_i$  est inférieure à un seuil fixé, le processus est terminé. Sinon, retourner à l'étape 1.

En pratique, nous établissons la liste des primitives et des points inclus dans chaque fenêtre de reconstruction, afin d'accélérer les calculs.

Nous décrirons dans le chapitre III•3 l'interface grâce à laquelle l'utilisateur peut définir les fenêtres de reconstruction et modifier les positions des primitives et leurs paramètres de champ. Nous allons maintenant décrire notre deuxième approche pour l'initialisation des primitives initiales, qui fait intervenir le calcul de l'axe médian discret des données.

# Chapitre III●3

## Interactivité et premiers résultats

### III●3.1 Approche semi-automatique

Nicolas Tsingos a implémenté cette approche semi-automatique en utilisant le système *Fabule*, une plate-forme pour la modélisation interactive et l'animation développée à *iMAGIS* [Gascuel 1994]. Cette plate-forme comprend une bibliothèque orientée-objets, implémentée en C++, fondée sur *Open-Inventor*, et une interface qui utilise le langage interprété *Tcl* et *OSF-Motif* pour une spécification aisée de démonstrateurs et d'interfaces.

#### 3.1.1 Positionnement interactif des primitives initiales

L'utilisateur spécifie interactivement les primitives implicites et les fonctions de champ associées qui serviront à initialiser le processus de reconstruction à l'aide des widgets 3D d'*Inventor*.

#### 3.1.2 Fenêtres de reconstruction

L'utilisateur crée et positionne également les fenêtres de reconstruction en fonction des données.

#### 3.1.3 Interface

L'interface graphique pour modéliser les primitives, visualiser les surfaces implicites générées et éditer les fonctions de champ est présentée dans les figures III●3.1 et III●3.2.

Les surfaces implicites sont visualisées en temps réel grâce à une technique d'échantillonnage fondée sur la migration de graines, qui s'adapte avec un rafraîchissement interactif aux évolutions de la surface implicite. La totalité du système de visualisation et de modélisation interactive de surfaces implicites est décrit dans [Tsingos et Gascuel 1994].

### III●3.2 Resultats

#### 3.2.1 Validation de l'algorithme sur des objets définis implicitement

L'algorithme de reconstruction a d'abord été validé sur des objets initialement créés par des primitives implicites. L'objet représenté dans la figure III●3.3 a été reconstruit à

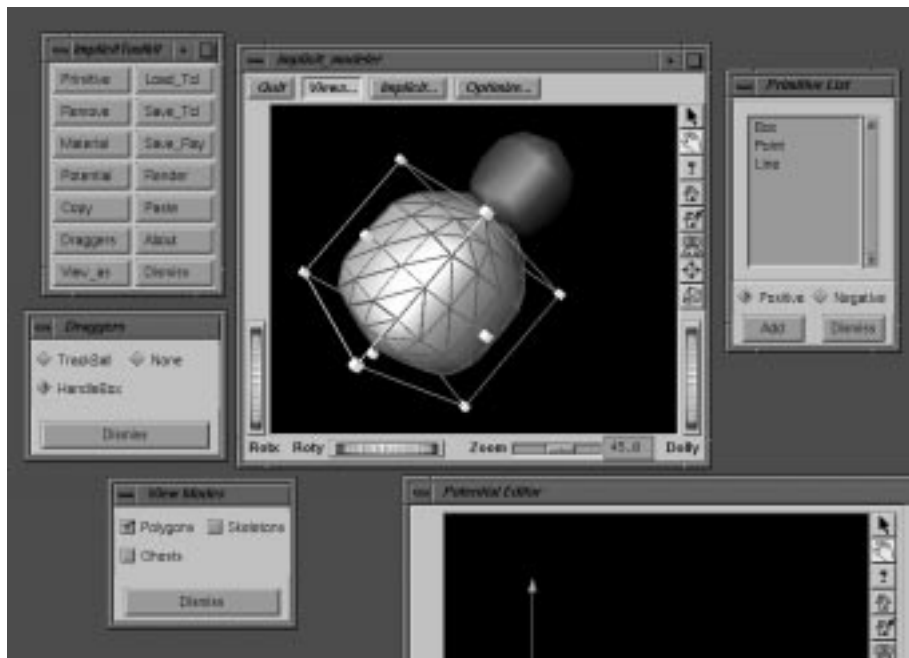


FIG. III●3.1: Interface pour éditer les primitives

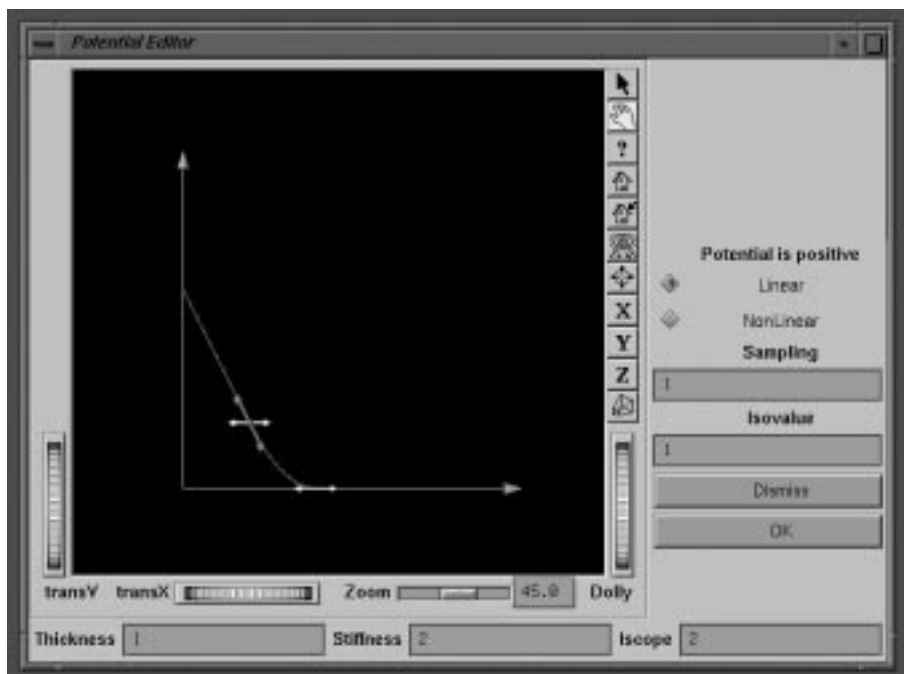
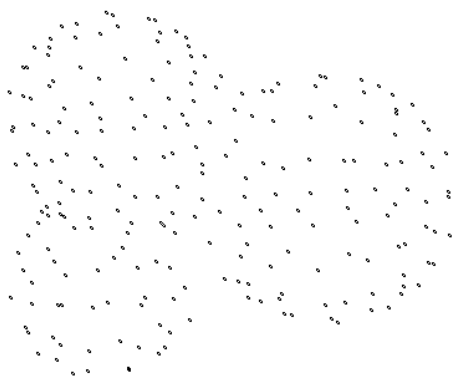
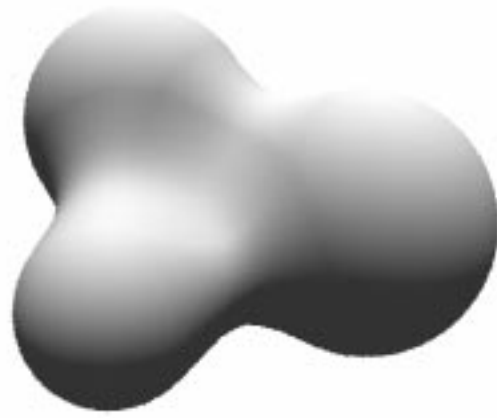


FIG. III●3.2: Interface pour éditer les fonctions potentielles

partir d'une primitive unique, pour valider le processus de subdivision automatique. Il y avait une unique fenêtre de reconstruction, qui incluait toute la scène. Les résultats montrent que les primitives originelles ainsi que leurs fonctions de champ ont été reconstruites correctement, sans créer de primitives en trop.



(a) points des données



(b) objet reconstruit

FIG. III●3.3: Reconstruction d'une iso-surface initialement créée avec trois primitives ponctuelles

### 3.2.2 Utilisation de fenêtres de reconstruction

L'exemple suivant illustre l'utilisation de fenêtres de reconstruction sur un objet qui possède un trou et des branchements, qui a également été créé à l'aide de primitives implicites.

Une premier calcul en utilisant une unique fenêtre de reconstruction a nécessité plus de deux heures de calcul sur une station Indigo2, R4400 à 150Mhz. La figure III●3.4 montre la surface obtenue à l'issue de ce calcul, elle est générée par 16 primitives.

Un deuxième calcul utilisant les fenêtres de reconstruction montrées sur la figure III●3.5 (a), a permis d'obtenir en une demi-heure seulement les 19 primitives qui génèrent la surface III●3.5 (b).

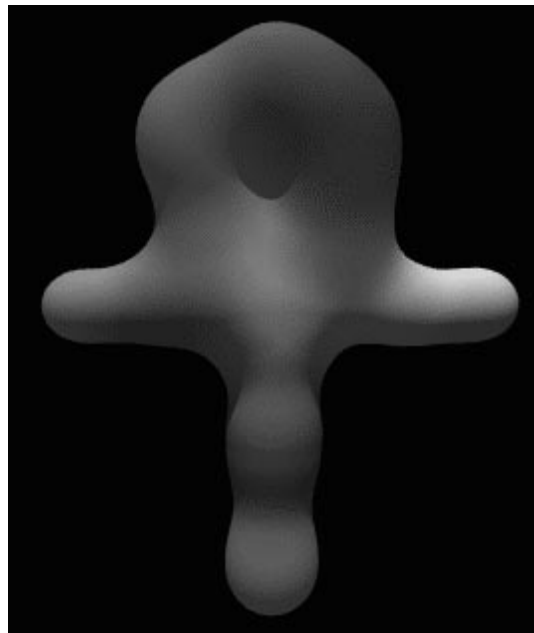
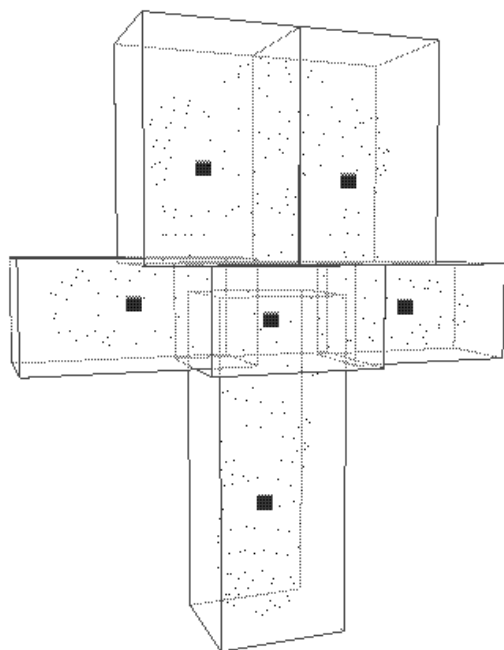
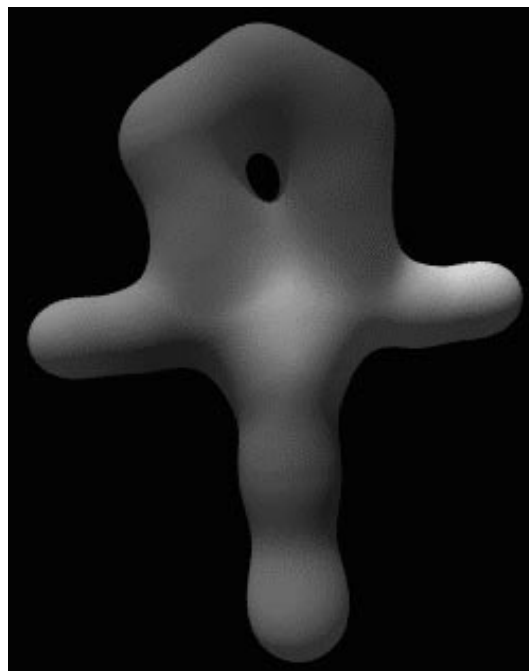


FIG. III●3.4: Reconstruction d'un objet avec une unique fenêtre de reconstruction



(a) Six fenêtres ont été interactivement positionnées



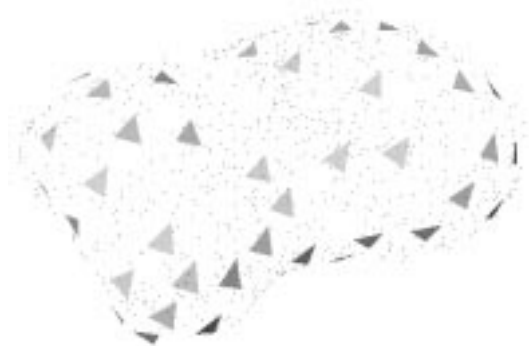
(b) Reconstruction avec ces fenêtres

FIG. III●3.5: Utilisation de fenêtres locales de reconstruction

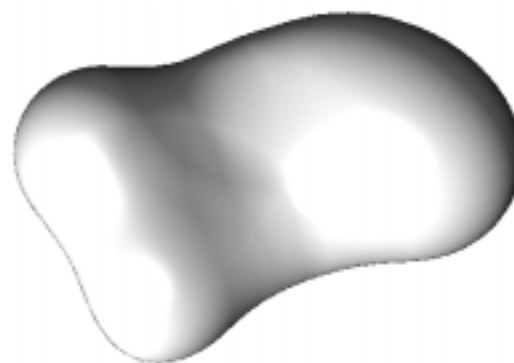
### 3.2.3 Reconstruction d'organes médicaux

#### 3.2.3.1 Un rein

La figure III●3.6 montre les 768 points des données et la surface reconstruite par 12 primitives. En (a), la surface implicite est représentée en utilisant des écailles, qui sont des triangles disposés sur la surface.



(a) Points des données et écailles de surface



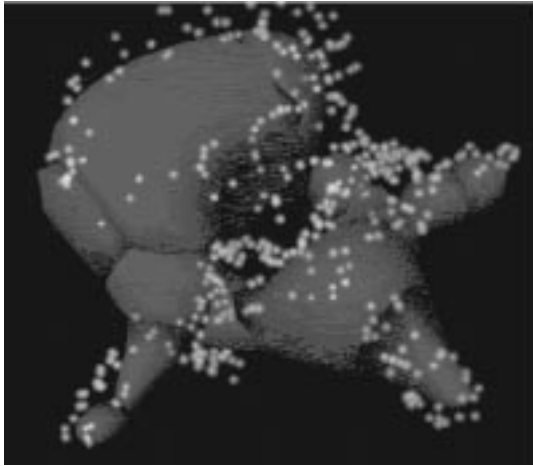
(b) Surface finale

FIG. III●3.6: Reconstruction de la surface d'un rein

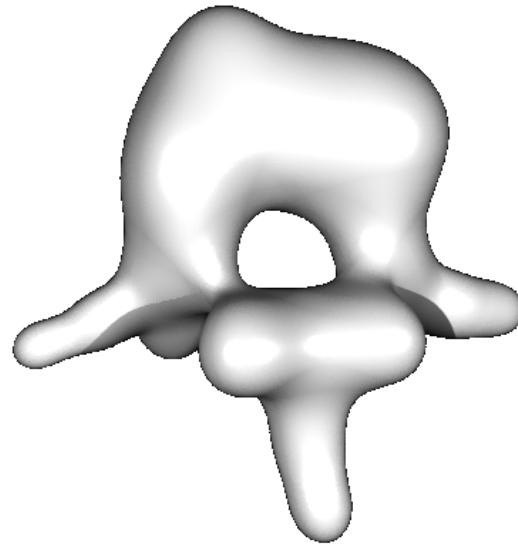


### 3.2.3.2 Une vertèbre

La figure III●3.7 montre en (a) une étape intermédiaire de la reconstruction, et en (b) le résultat final, qui représente la reconstruction de 2431 points des données avec 18 primitives en 45 minutes. Le taux de compression est d'environ 1 : 100 (environ 29000 octets pour les données, et seulement 360 pour la représentation implicite).



(a) Etape intermédiaire



(b) Reconstruction avec 18 primitives

FIG. III●3.7: Reconstruction d'une vertèbre

# Chapitre III●4

## Construction de l'axe médian

### III●4.1 Présentation

L'axe médian [Pfaltz et Rosenfeld 1967] est une représentation d'un objet à l'aide de sphères, qui se trouvent réparties sur le squelette de l'objet. Ces sphères semblent donc appropriées pour devenir des primitives d'une iso-surface de l'objet.

#### 4.1.1 Définitions

Considérons un ensemble  $S$  de sphères incluses dans l'objet.

**Sphère maximale :** une sphère maximale est une sphère de  $S$  qui n'est incluse dans aucune autre sphère de  $S$ .

**Axe médian :** l'axe médian est le lieu des centres des sphères maximales. On lui associe également les rayons des sphères maximales.

#### 4.1.2 Description de notre méthode

Notre approche est fondée sur la géométrie discrète : elle nécessite une approximation de l'objet comme un ensemble de voxels. Pour obtenir cette représentation, nous partons de la boîte englobante des points des données, que nous divisons en une grille 3d régulière. Nous devons ensuite identifier les voxels de cette grille qui appartiennent à l'intérieur de l'objet. Nous étiquetterons ceux-ci "voxels intérieurs".

1. **Partitionnement de l'espace :** Les voxels qui contiennent des points des données sont étiquetés "voxels frontière", car ils sont traversés par la surface de l'objet.
2. **Étiquetage de l'intérieur :** Il n'est pas simple d'identifier l'intérieur d'un objet de topologie quelconque, qui peut avoir plusieurs composantes connexes. Pour le faire, nous étiquetons d'abord l'extérieur. Nous partons d'un coin de la grille 3D, et nous propageons l'information "voxel extérieur" à tous les voisins non encore étiquetés. Les voxels intérieurs sont ceux qui n'ont pas été étiquetés à la fin du balayage. Les voxels frontière ne sont pas entièrement inclus dans l'objet : ils ne sont donc pas considérés comme des voxels intérieurs.

Cette méthode marche dans la plupart des cas, mais elle ne peut pas s'appliquer à un objet fermé contenant un trou fermé. Dans un pareil cas, c'est-à-dire lorsque l'extérieur a plusieurs composantes connexes, il suffit de faire un étiquetage de la partie complémentaire aux points de surface et de supprimer les composantes qui

touchent le bord. Nous verrons en III●4.4 la limitation de cette méthode, qui est que l'objet discret composé des voxels frontière doit définir une surface fermée.

3. **Calcul de la carte de distance et extraction de l'axe médian :** Une fois que les voxels frontière et intérieur sont définis, nous calculons une carte en utilisant la distance du chanfrein. Pour chaque voxel à l'intérieur de l'objet, la carte donne la distance à la frontière de l'objet. Nous utilisons ensuite un critère local pour extraire l'axe médian.

## III●4.2 Carte de distance

La *transformée de distance* est une opération qui prend en entrée une image dans laquelle les voxels sont étiquetés. Les voxels intérieurs à l'objet ont la valeur 1, les autres la valeur 0. La transformée de distance remplace la valeur de chacun des voxels de l'intérieur par la distance du voxel à l'extérieur de l'objet, c'est à dire la distance au voxel extérieur à l'objet le plus proche. La nouvelle image ainsi générée s'appelle une *carte de distance*.

### 4.2.1 Distance du chanfrein

Nous utilisons une distance discrète appelée distance du chanfrein  $d_{3,4,5}$  [Borgefors 1986 ; Thiel 1994] pour approximer la distance euclidienne  $3D$ .

Le principe de cette distance est de donner un coût pondéré de 3, 4, ou 5 pour se déplacer d'un voxel  $v_0$  à chacun de ses 26 voisins (voir figure III●4.1 page 90). Les figures III●4.1 et III●4.3 constituent des exemples de la restriction de  $d_{3,4,5}$  en  $2D$ .

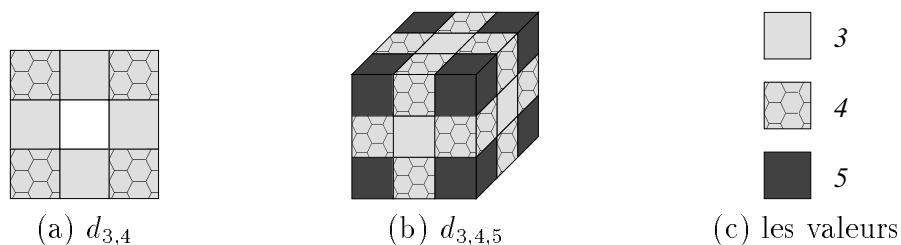


FIG. III●4.1: poids des distances du chanfrein  $d_{3,4,5}$  et  $d_{3,4}$ .

### 4.2.2 Construction de la carte de distance

La distance du chanfrein permet un calcul rapide de la carte de distance. En effet, l'utilisation de l'algorithme de Rosenfeld [Rosenfeld et J.L. 1966] évite de faire pour chaque voxel intérieur une coûteuse recherche exhaustive parmi les voxels pour trouver celui qui réalise le minimum de la distance. L'information de distance est propagée à tous les voxels en deux opérations séquentielles de convolution. Les filtres de convolution sont deux demi-cubes de  $3 \times 3 \times 3$ , centrés sur le voxel courant  $v_0$ . Ils sont symétriques par rapport à  $v_0$ . Le premier filtre  $F_A$  est utilisé pour le parcours aller de l'image. Il contient les voxels  $v_i$  qui se trouvent avant  $v_0$  dans la direction de parcours. Leurs coordonnées sont  $(x_i, y_i, z_i) \in \{-1, 0, 1\}^3$ , et leurs poids  $w_i$  (3, 4 ou 5). Le deuxième filtre  $F_R$  sert au parcours retour. Il contient les voxels  $v_i(-x_i, -y_i, -z_i)$  de poids  $w_i$ . Aucun des deux filtres ne contient 0 (Figure III●4.2).

Initialement, chaque voxel de l'objet a une valeur de 1 et les autres voxels sont à 0. Les deux passes du filtre remplacent la valeur  $V$  d'un voxel par une valeur qui dépend des valeurs des voxels voisins.

**Passé Aller** : de haut en bas, d'arrière en avant, et de gauche à droite

$$V[x, y, z] = \min_{v_i \in F_A} \{V[x + x_i, y + y_i, z + z_i] + w_i\}$$

**Passé Retour** : de bas en haut, d'avant en arrière, et de droite à gauche

$$V[x, y, z] = \min_{v_i \in F_R} \{V[x, y, z], V[x - x_i, y - y_i, z - z_i] + w_i\}$$

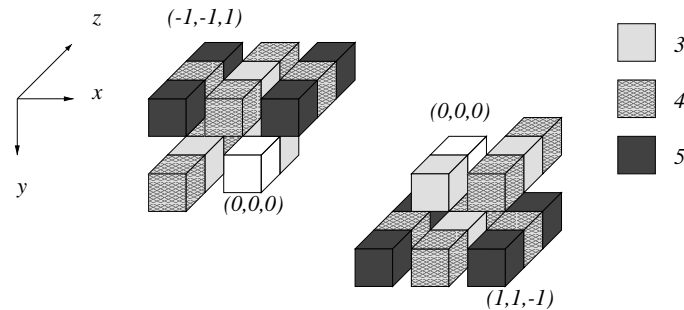


FIG. III●4.2: demi-filtres Aller et Retour, et les poids associés.

Une fois la carte de distance calculée, il faut sélectionner les points de l'axe médian.

### III●4.3 Extraction de l'axe médian

Nous utilisons un simple critère local. Chaque voxel qui vérifie le critère de maximum local est inclus dans l'axe médian.

**Maximum local** : Un voxel  $v$  qui ne propage l'information à aucun de ses voisins  $n_i(v)$  du filtre  $3 \times 3 \times 3$  est appelé un maximum local.

$$n_i(v) < v + w_i \quad \forall i$$

Avant d'utiliser le critère du maximum local, les voxels de la carte de distance ayant une valeur de 3 sont modifiés et prennent la valeur 1. Le lecteur intéressé par plus de détails se reportera à [Thiel 1994].

### III●4.4 Ajuster la résolution de l'axe médian en fonction des données

Dans ce paragraphe nous étudions l'influence du paramètre de résolution de l'axe médian, que nous définissons comme le nombre de voxels sur une arête de la grille 3D.

Un axe médian de faible résolution ne représente pas les détails de l'objet (voir Figure III●4.4 page 92 (a)), mais la résolution ne peut pas être augmentée arbitrairement. Si

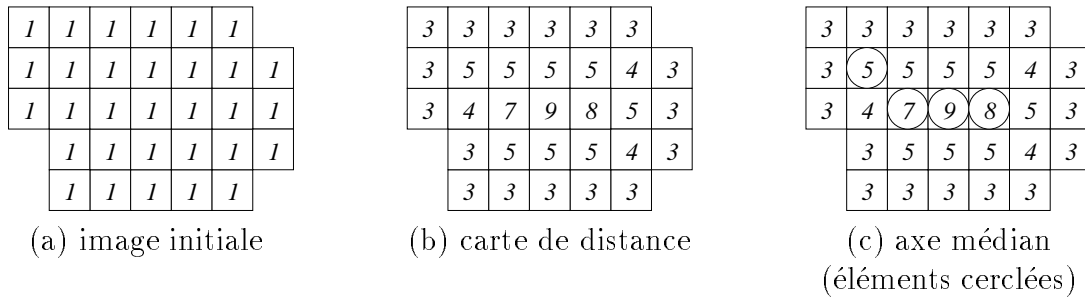


FIG. III●4.3: Exemple des étapes du calcul de l'axe médian en  $2D$  avec  $d_{3,4}$ .

nous représentons le volume intérieur de l'objet en fonction de la résolution, nous voyons que cette fonction atteint un maximum, pour décroître ensuite (voir Figure III●4.5).

Après ce maximum, la grille est trop fine, et les voxels "frontière" ne définissent plus une surface fermée. Dans ce cas, la séparation intérieur/extérieur ne peut plus être correctement établie. (voir Figure III●4.4 (c)).

En pratique, la valeur du paramètre de résolution qui est choisie (voir Figure III●4.4 (b)) doit représenter un compromis entre la qualité de la résolution finale et le nombre de sphères de l'axe médian. En effet, ce nombre augmente avec la résolution et intervient dans le temps de traitement. Nous choisissons la valeur au milieu de la zone où le volume augmente.

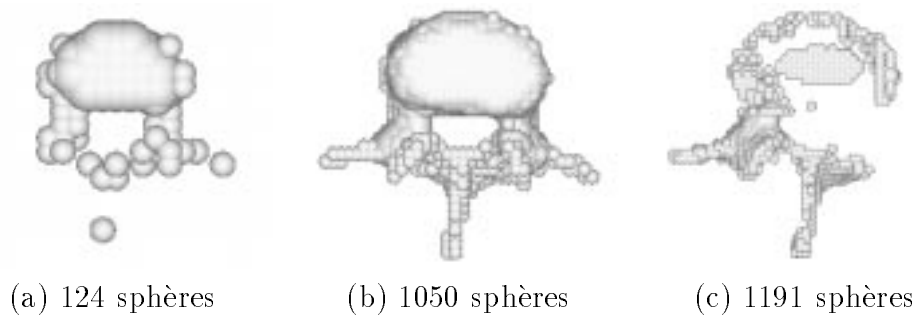


FIG. III●4.4: axe médian de la vertèbre, aux résolutions (a) 24, (b) 42, (c) 58

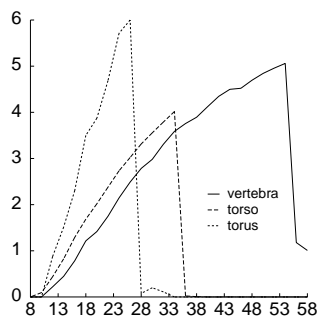


FIG. III●4.5: Estimation du paramètre de résolution.

# Chapitre III•5

## 2ème approche : reconstruction automatique

Ce modèle est continu, il donne une représentation analytique implicite de la surface engendrée, et nécessite donc un traitement particulier pour la visualisation (marching cube, discrétisation, migration de graines ...). Nous supposons par la suite que nous disposons d'un outil adapté pour la visualisation des surfaces implicites. Nous avons utilisé l'algorithme des marching cubes, dans son implémentation existante dans AVS, dans le module isosurface.

### III•5.1 D'un axe médian à une surface implicite

Comme une primitive ponctuelle isolée génère une sphère implicite, une solution simple serait d'utiliser directement les points de l'axe médian pour générer des sphères implicites qui se juxtaposeraient pour reconstruire la surface de l'objet. Les paramètres des fonctions de champ des primitives seraient calculés d'après le rayon des sphères de l'axe médian, en utilisant des pentes élevées, pour limiter la fusion des sphères voisines.

Mais cette approche a des inconvénients majeurs :

- La surface obtenue ne serait pas lisse, du fait de l'utilisation de pentes élevées pour les fonctions de champ.
- La propriété fondamentale de l'axe médian est qu'aucune sphère n'est incluse dans une autre. Mais l'axe médian n'est pas minimal, il peut exister des sphères incluses dans l'union de leurs voisines. Le potentiel de plusieurs sphères s'ajoutant, la surface implicite est éloignée des points des données.

De fait, notre idée est de mieux exploiter les propriétés que les surfaces implicites ont de se mélanger pour générer l'iso-surface de l'objet à partir d'un sous-ensemble des points de l'axe médian. Cela mène à une représentation plus précise et plus compacte de la forme reconstruite.

Le paragraphe suivant explique comment nous utilisons l'axe médian pour sélectionner et ajouter de nouvelles primitives de manière itérative, en affinant progressivement la représentation de la forme. Cette méthode évite le procédé coûteux de diviser les primitives en deux primitives filles qui était utilisé dans les approches précédentes.

## III•5.2 Création et affinage de la surface implicite

L'idée principale est de sélectionner progressivement des éléments de l'axe médian, de les transformer en primitives implicites, avec leurs fonctions de champ associées, et d'ajuster leurs paramètres pour améliorer la qualité de la reconstruction.

Pour cela, il est important de définir une heuristique de sélection des primitives qui soit robuste et efficace. L'utilisation de fonctions de champ locales permet de définir une telle heuristique. Ce paragraphe décrit le choix des fonctions de champ, et détaille l'heuristique de sélection et l'algorithme général de reconstruction.

### 5.2.1 Un critère local efficace de sélection

Avant de commencer le processus de reconstruction, nous associons une primitive implicite ponctuelle  $I_i$  à chaque sphère  $S_i$  de l'axe médian. Les coordonnées  $x_i, y_i, z_i$  de la primitive sont celles du centre de la sphère, le paramètre  $e_i$  est le rayon de la sphère. Nous donnons au paramètre  $k_i$  la valeur inverse de la taille des voxels utilisée dans la construction de l'axe médian. Cette valeur donne l'ordre de grandeur de la zone d'influence de la primitive.

Ensuite, à chaque étape du processus de reconstruction, nous ajoutons certaines primitives de l'axe médian à l'ensemble de primitives qui définissent l'objet implicite, afin d'améliorer son adéquation aux données. Les meilleures candidates pour être ajoutées à l'objet implicite sont les primitives dont la zone d'influence correspond à la partie de la surface où la reconstruction est la moins bonne, comme le montre la figure III•5.1.

Nous utilisons le critère de l'équation III•2.2 :

$$C(I_i) = \left( \sum_{j=1}^{m_i} (f(D_{j,i}) - iso)^2 \right), \quad (\text{III•5.1})$$

que nous avons présenté dans la section III•2.3 page 81.

En pratique, nous utilisons ce critère pour sélectionner plusieurs primitives à chaque itération de l'algorithme, en prenant soin qu'elles soient bien distribuées à l'intérieur de l'objet. Nous optimisons ensuite les paramètres de ces primitives avant d'en sélectionner de nouvelles. Le paragraphe suivant indique les détails du procédé.

### 5.2.2 L'algorithme de reconstruction

L'algorithme que nous avons composé et mis en œuvre dans le cas des surfaces implicites implémente un schéma de traitement de l'information différent du schéma général présenté par la figure I•1.5 page 20. Dans l'approche que nous présentons ici, le modèle est créé lors du traitement des données, et n'existe pas au préalable. La figure III•5.2 détaille le nouveau schéma.

Les primitives  $I_i$  sont initialement toutes dans la liste  $LAM$  des éléments de l'axe médian. Le processus de reconstruction enlève peu à peu des primitives de cette liste pour les ajouter à la liste  $LI$  des primitives de l'objet implicite. A chaque étape les primitives sélectionnées sont transférées dans une liste temporaire  $LI_{\text{etape}}$ , puis les primitives de cette liste sont ajoutées à  $LI$ , et la liste temporaire est vidée.

De la même manière, les points des données sont au cours de chaque étape de reconstruction progressivement transférés de la liste  $LD$  qui les contient initialement tous vers

une liste des points des données sélectionnés  $LDS$ . Les points de  $LDS$  sont ceux qui se trouvent dans la zone d'influence d'une primitive qui a été ajoutée à la liste  $LI$  au cours de l'étape courante.

A chaque étape de reconstruction :

1. La liste  $LI_{\text{etape}}$  est initialement vide.
2. La liste  $LD$  contient tous les points des données, la liste  $LDS$  est vide.
3. Tant que la liste  $LD$  n'est pas vide :
  - Calculer le critère local  $C_i$  de toutes les primitives de  $LAM$ , en ne prenant en compte que les points des données de  $LD$
  - Transférer la primitive de plus haut  $C_i$  de  $LAM$  vers  $LI_{\text{etape}}$ , et transférer les points des données de  $LD$  qui sont dans sa zone d'influence vers  $LDS$ .
4. Optimiser les paramètres  $e_i$  et  $k_i$  des primitives de la liste  $LI_{\text{etape}}$ , en gardant leurs coordonnées fixées. (En effet, l'estimation de la position de ces primitives, qui étaient des points de l'axe médian, est bonne en première approximation)
5. Transférer les primitives de  $LI_{\text{etape}}$  vers  $LI$ . Optimiser tous les paramètres des primitives de  $LI$ .
6. Les points des données sont remis dans la liste  $LD$ , la liste  $LDS$  est vidée.

Comme dans les approches précédentes, l'énergie qui est minimisée lors des étapes d'optimisation est définie par la somme des valeurs du champ aux  $n_{\text{données}}$  points des données.

$$E = \frac{1}{n_{\text{données}}} \left( \sum_P (f(P) - iso)^2 \right) \quad (\text{III}\bullet 5.2)$$

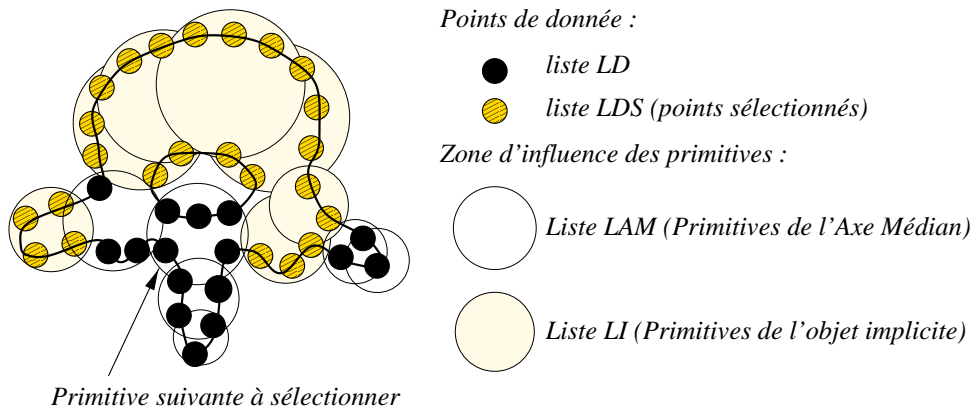


FIG. III•5.1: Une étape de l'algorithme



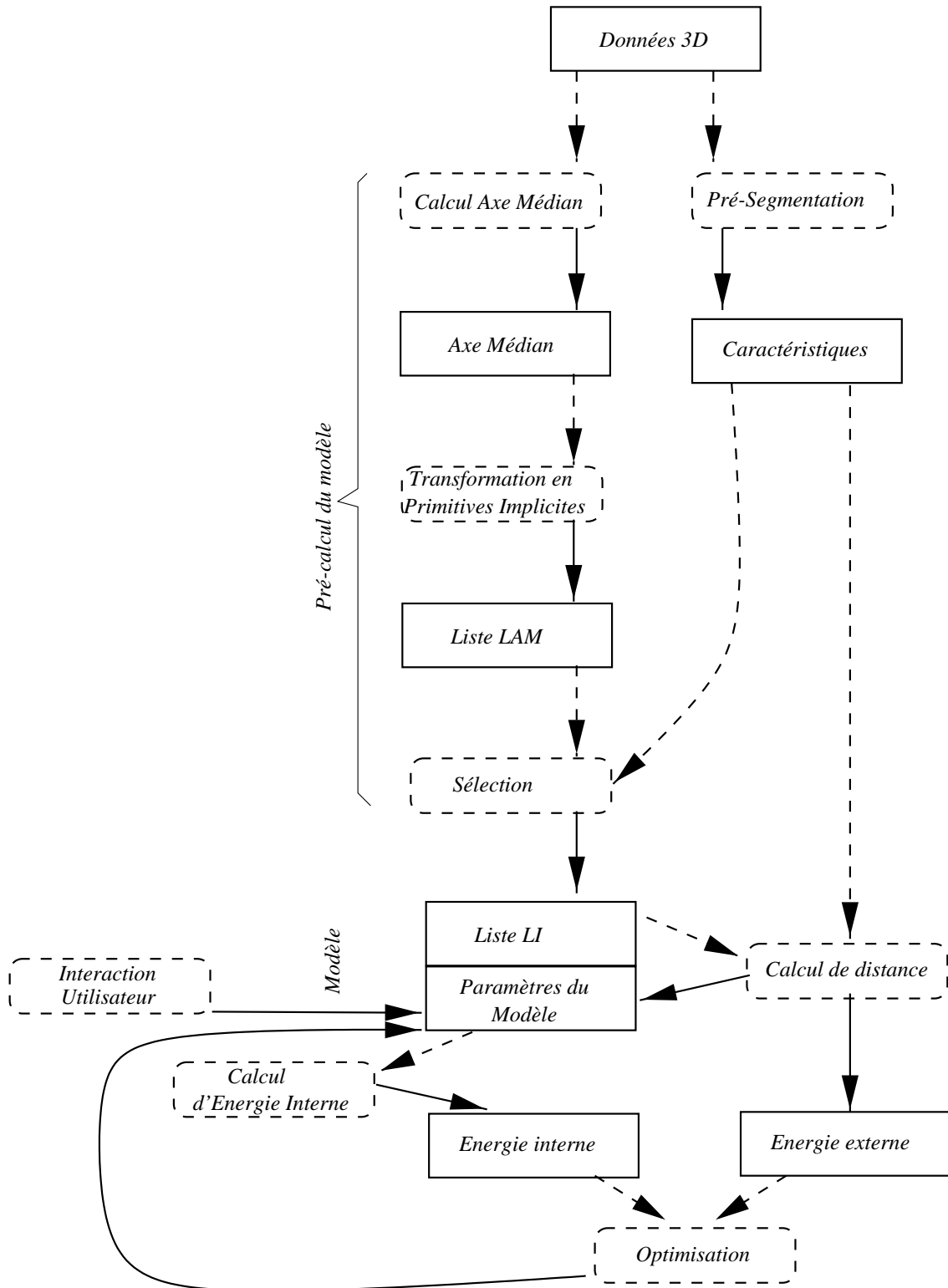


FIG. III●5.2: Traitement de l'information pour nos surfaces implicites

# Chapitre III•6

## Résultats de l'approche automatique

Nous présentons dans ce chapitre les résultats obtenus grâce à notre méthode de reconstruction automatique.

### III•6.1 Résultats expérimentaux

Nous avons implémenté le processus de reconstruction à l'aide du logiciel de visualisation AVS, sur une station DEC AXP3000. Les résultats numériques sont présentés dans le tableau III•6.1 page 99.

Les exemples que nous avons choisis sont d'une part un tore et un "Y" synthétique, et d'autre part le torse et la vertèbre que nous avons présentés dans la partie II . Pour ces derniers objets, afin d'avoir des points de données représentant une surface fermée, nous avons utilisé les sommets des  $\delta$ -snakes.

#### 6.1.1 Exemples d'axes médians

Les figures III•6.1 et III•6.2 montrent les résultats des calculs d'axes médians sur trois objets très différents. Les rayons sont représentés par des niveaux de gris : les plus sombres sont les plus grands.

Pour ces exemples, un calcul d'axe médian prend une dizaine de secondes.

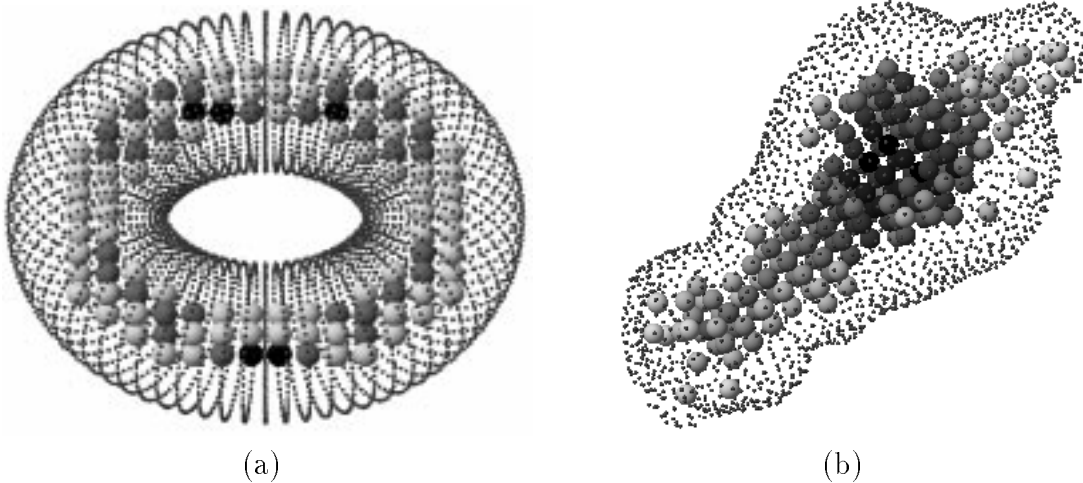


FIG. III•6.1: Les points des données et les sphères de l'axe médian du tore et du torse



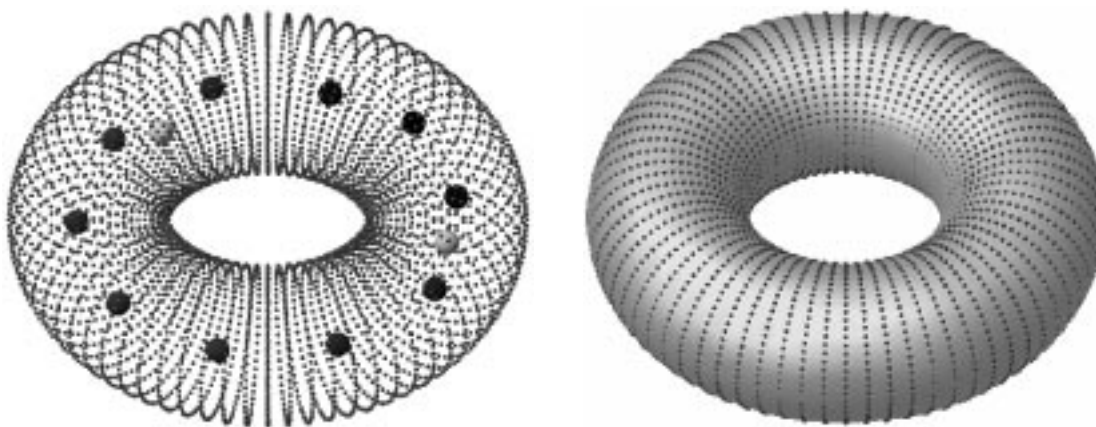
FIG. III●6.2: Les points des données et les sphères de l'axe médian de la vertèbre

### 6.1.2 Exemples de reconstruction

Cette section montre les résultats de la reconstruction avec notre algorithme de sélection progressive, sur le tore (Figures III●6.3, le torse (Figure III●6.4), Le "Y", et la vertèbre (Figures III●6.6 et III●6.7).

L'exemple du "Y" synthétique (Figure III●6.5) montre que la reconstruction permet de traiter les cas où les points sont répartis sur des contours plans, et où la distance entre les plans est grande par rapport à la distance entre points d'un même contour, ce qui est un cas où d'autres méthodes comme celle de Hoppe [Hoppe et al. 1994] échouent.

L'ensemble des points des données de la vertèbre qui comprenait 19837 points a été réduit à 2037 points pour l'étape de reconstruction.



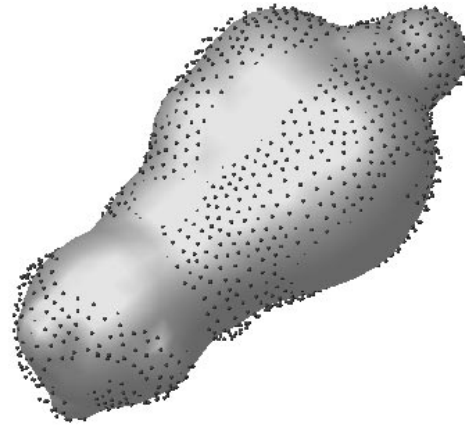
(a) Points des données et Primitives

(b) Points des données et surface reconstruite.

FIG. III●6.3: Le tore

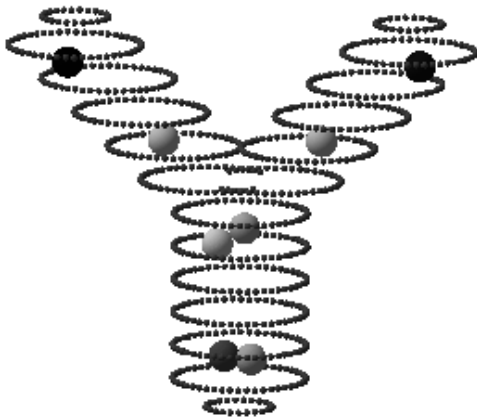


(a) Points des données et Primitives



(b) Points des données et surface reconstruite.

FIG. III●6.4: Le torse



(a) Points des données et Primitives



(b) Points des données et surface reconstruite.

FIG. III●6.5: Le "Y"

TAB. III●6.1: Statistiques de reconstruction

objet	nombre de points des données	résolution de l'axe médian	nombre de sphères de l'axe médian	nombre de primitives	Energie finale	temps de calcul (sec.)	nombre de passes
tore	4176	22	172	12	$5.46e^{-4}$	163	1
torse	2027	26	181	42	$3.78e^{-3}$	843	3
Y	871	14	16	10	$2.74e^{-3}$	194	1
vertebre	19837 2037	42	1050	46	$3.14e^{-2}$	3135	1

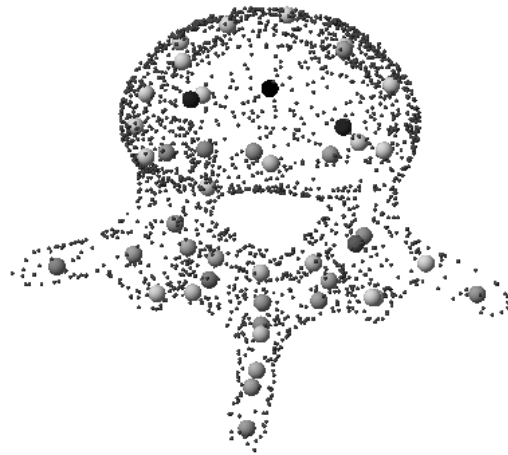
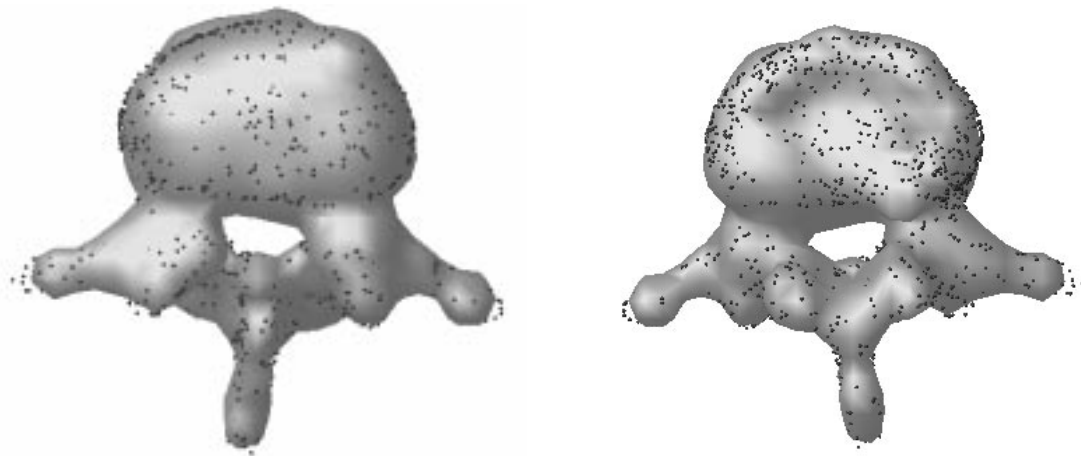


FIG. III●6.6: Reconstruction de la vertèbre à l'aide de 46 primitives ponctuelles.  
Vue de face des points des données et des primitives



(a) vue de face

(b) vue de dos

FIG. III●6.7: La surface reconstruite de la vertèbre

# Chapitre III•7

## Conclusion de la partie III

### III•7.1 Conclusion

Nous avons présenté une méthode automatique de reconstruction d'une surface qui peut être appliquée à un ensemble de points des données non structuré, et qui permet de reconstruire des objets de géométrie et de topologie complexes. Cette méthode est fondée sur des iso-surfaces générées par des primitives, et permet une représentation compacte de formes complexes.

Notre approche est originale en le sens que nous calculons grâce à la distance du chanfrein l'axe médian d'un ensemble de points à la surface d'un objet, afin d'exploiter l'information donnée par cet axe médian discret pour faciliter et accélérer la reconstruction.

Plutôt que de partager des primitives implicites en deux comme dans [Muraki 1991] avant chaque étape d'optimisation, nous avons élaboré et implémenté un algorithme rapide et robuste qui sélectionne progressivement de nouvelles primitives dans l'axe médian discret, en utilisant un critère de sélection local pour déterminer la zone la plus mal reconstruite.

En alternance avec le processus de sélection, nous optimisons les paramètres des primitives implicites sélectionnées pour faire passer la surface par les points des données.

Nous obtenons ainsi une représentation compacte de formes lisses, car le nombre de primitives sélectionnées est de beaucoup inférieur au nombre de points de l'axe médian discret.

### III•7.2 Perspectives

Le principal inconvénient de cette méthode porte sur la qualité de la reconstruction. Nous avons vu que cette approche est très bien adaptée pour la reconstruction des formes lisses. Elle est en revanche insuffisante si l'on recherche un niveau de détail important. Dans ce cas nous proposons deux pistes de travail qui nous semblent prometteuses.

#### 7.2.1 Evolution de la représentation géométrique

La première porte sur la représentation géométrique du modèle. Elle est en fait double, et composée de deux points. Nous avons utilisé des primitives ponctuelles. Ce choix peut évoluer, le remplacement des points par des primitives plus complexes comme des segments de courbes permet d'envisager une reconstruction d'une qualité équivalente avec

moins de primitives, et une meilleure restitution des détails pour un nombre de primitives équivalent. Eric Ferley a travaillé dans ce sens au sein de l'équipe iMAGIS [Ferley et al. 1997]. La deuxième partie de cette piste de travail sur l'évolution de la représentation géométrique concerne les fonctions de champ associées aux primitives. Nous avons utilisé une fonction locale déterminée par deux paramètres (cf III●2.1) qui dépend de la distance à la primitive implicite, or Carole Blanc et Christophe Schlick ont montré l'intérêt de fonctions de champ non isotropes [Blanc et Schlick 1995]. En associant un repère à la primitive, il est possible, en coordonnées sphériques par exemple, de définir une nouvelle "distance" à la primitive qui sera modulée en fonction de  $\theta$  et  $\phi$ . Les auteurs présentent des exemples de modelage d'objets en utilisant ces objets. Dans notre optique de reconstruction automatique, un certain travail reste à effectuer pour intégrer des fonctions non isotropes.

### 7.2.2 Adjonction de déformations locales

La deuxième piste consiste à appliquer à la surface implicite des déformations locales. Nous avons décrit dans la section I●4.3 comment plusieurs auteurs combinent des surfaces implicites simples comme les superquadriques avec des déformations locales. Il est également possible d'ajouter des déformations locales à nos surfaces implicites à base de primitives [Opalach et Cani-Gascuel 1997]. Cette piste permettra d'obtenir plus de détails dans la représentation de la surface.

### 7.2.3 Vers un modèle déformable à mémoire de forme

Nous avons montré quels types d'évolutions permettront des représentations plus détaillées des objets. Interrogeons-nous maintenant sur les évolutions possibles de notre modèle implicite en tant que modèle déformable. Notre modèle implicite permet de reconstruire la surface d'un objet, en construisant d'abord un squelette discret. Tel que nous l'avons élaboré il ne contient pas de connaissance a priori sur la forme de l'objet à reconstruire. Nous inscrivons cette tâche dans les perspectives ouvertes par notre travail : l'étude de l'utilisation d'un modèle implicite d'un objet pour traiter une image qui représente un objet du même type, que ce soit pour une application de reconstruction, de reconnaissance ou de segmentation.

Nous avons mené ce type d'étude sur un autre modèle, un modèle volumique déformable, qui est mieux adapté à la segmentation des images que les modèles surfaciques. En effet nous avons des moyens d'extraire des caractéristiques de bas niveau dans les images médicales qui guident la déformation du modèle. Mais ces caractéristiques ne se trouvent pas seulement sur une unique surface dans les images, elles se trouvent réparties dans le volume de l'image. Le choix d'un modèle volumique permet d'utiliser ces caractéristiques, c'est ce que nous allons montrer dans la partie suivante.

## Quatrième partie

### Un modèle volumique déformable





*“La confrontation des données et des modèles est une dimension essentielle de l’activité scientifique.”*

Sciences Humaines, commentaire d’une phrase de Carl Sagan.

Le modèle déformable volumique que nous présentons dans cette dernière partie a été développé à l’origine par Richard Szeliski et Stéphane Lavallée [Szeliski et Lavallée 1994 ; Szeliski et Lavallée 1996]. Il présente l’avantage d’utiliser à la fois une représentation géométrique volumique : un ensemble de particules, et une déformation elle aussi volumique : l’octree-spline. Nous l’avons testé dans sa version originale, et nous avons constaté que son caractère volumique lui donne bien la possibilité de s’appuyer sur des caractéristiques réparties dans l’espace et non pas seulement des caractéristiques disposées sur une surface comme les modèles surfaciques ou implicites. Mais nous avons également mis en évidence la limitation que constituait l’obligation de ne pouvoir utiliser que la position 3D des caractéristiques pour calculer la distance entre le modèle et les données. Nous avons donc mis au point une méthode de calcul de distance adaptée au problème, en utilisant des arbres k-d.

## IV•1.1 Formulation de la méthode

La modélisation que nous avons présentée dans la partie I se retrouve dans le cadre de notre modèle volumique. Il y a cependant une différence : la déformation ne s’applique pas au modèle, mais aux données.

### 1.1.1 Déformation des données

Appelons  $\text{Ref}_{\text{modèle}}$  le système de coordonnées du modèle, et  $\text{Ref}_{\text{données}}$  celui des données. Il s’agit d’estimer la déformation indirecte  $\mathbf{T}$  déterminée par le vecteur de paramètres  $\mathbf{p}$  qui transforme tout point du repère  $\text{Ref}_{\text{données}}$  dans le repère  $\text{Ref}_{\text{modèle}}$ .

La déformation habituellement recherchée est la déformation inverse, qui permet d’obtenir le modèle déformé dans le repère des données.

Or le modèle  $\mathcal{M}$  a une structure de données associée qui possède une méthode de calcul de distance. Cette méthode  $\text{dist}_{\mathcal{M}}(\mathbf{m})$  indique la distance au modèle pour tout point  $\mathbf{m}$ . Cette structure de données, qui était une carte de distance octree-spline dans la méthode originale, est un arbre k-d dans notre version. Elle permet de calculer rapidement

la distance au modèle, et nécessite un pré-calcul pour sa construction. Si nous avons conservé l'option de déformer le modèle, il aurait fallu adapter cette structure de données au fur et à mesure de la déformation, ce qui représente des calculs importants.

C'est pourquoi nous avons été conduits à ce choix de déformer les données. Pour nous ramener à une déformation du modèle, nous inversons la déformation obtenue en fin de processus (voir section IV•4.4).

### 1.1.2 Minimisation d'une énergie

A partir des paramètres d'état de notre modèle volumique déformable, nous définissons une fonctionnelle énergétique. Détaillons son expression.

Les données  $\mathcal{D}$  sont représentées par un ensemble de points de dimension  $k$ ,  $\mathcal{D} = \{\mathbf{d}_i, i = 1 \dots n_{\text{données}}\}$ . En effet nous utilisons un **modèle enrichi**, à base d'une association de caractéristiques de bas niveau que nous décrivons au chapitre IV•2.

Notons  $\mathbf{r}_i = \mathbf{T}_p(\mathbf{d}_i)$  les transformés des points des données. Les notations que nous avons introduites sont illustrées par la figure IV•1.1.

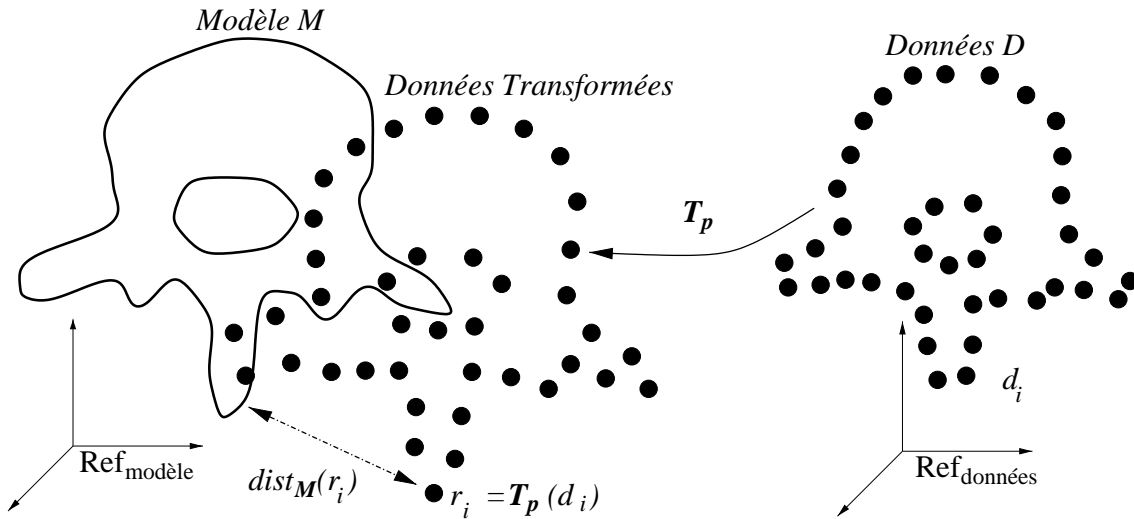


FIG. IV•1.1: Paradigme du modèle volumique déformable

La fonctionnelle s'écrit :

$$\mathcal{C}(\mathbf{p}) = \sum_{i=1}^{n_{\text{données}}} \frac{1}{\sigma_i^2} [\text{dist}_{\mathcal{M}}(\mathbf{r}_i)]^2 + \mathcal{R}(\mathbf{p}), \quad (\text{IV}\bullet 1.1)$$

Où  $\text{dist}_{\mathcal{M}}(\mathbf{r}_i) = \text{dist}_{\mathcal{M}}(\mathbf{T}_p(\mathbf{d}_i))$  est la distance du point  $\mathbf{r}_i$  à  $\mathcal{M}$  et  $\sigma_i^2$  est la variance associée au point  $i$ .

Le terme  $\mathcal{R}(\mathbf{p})$  correspond à une fonction de coût de régularisation ou stabilisation, qui sera décrite à la section IV•4.2

La fonctionnelle est minimisée en fonction des paramètres du modèle, afin de trouver une transformation géométrique  $\mathbf{T}$  telle que tous les points se trouvent sur le modèle  $\mathcal{M}$ .

## IV•1.2 Notion de points aberrants

Il n'est pas possible de trouver une transformation régulière qui amène tous les points des données sur le modèle dans le cas où certains points des données ne correspondent à aucune partie du modèle. Ces points sont appelés **points aberrants** ou **outliers** (figure IV•1.2).

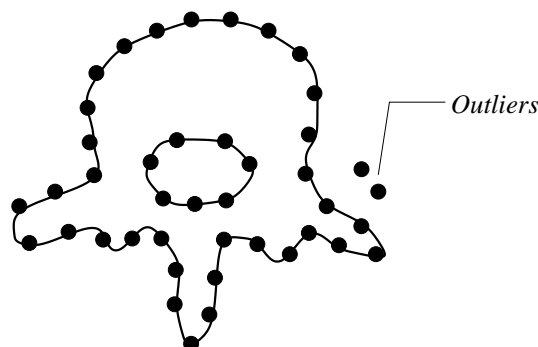


FIG. IV•1.2: Illustration des points aberrants

## IV•1.3 Segmentation par inférence

L'objectif de cette partie est de mettre au point une méthode de segmentation à partir d'un modèle déformable volumique.

La déformation du modèle est calculée à partir d'éléments géométriques qui ont une certaine répartition dans l'espace, mais elle est définie dans tout l'espace. La régularisation de la déformation au cours du processus de minimisation de la fonctionnelle du modèle assure sa régularité spatiale. La déformation va pouvoir être appliquée à d'autres structures que celles qui ont guidé la déformation, en particulier à des structures de référence, c'est ce que nous appelons **inférence**.

L'inférence à deux types d'applications : la localisation de structures anatomiques non présentes dans l'image des données, et la segmentation de cette image des données. C'est cette dernière piste que nous avons choisi d'explorer.

Considérons un exemple qui correspond à notre expérimentation (section IV•6.2). Nous avons comme données un volume TDM 3D du rachis lombaire. Le modèle contient d'une part un volume 3D qui correspond à un tel examen (qui peut avoir été obtenu comme une image moyenne sur une population donnée), et d'autre part la définition exacte dans ce volume 3D de la surface d'une vertèbre. Les deux images 3D des données et du modèle serviront à définir la déformation, et la surface présente dans le modèle définira par inférence la surface de la vertèbre correspondante dans les données (figure IV•1.3).

## IV•1.4 Organisation de cette partie

Dans le chapitre IV•2 nous allons d'abord présenter les caractéristiques que nous avons retenues, et la manière de les extraire que nous avons choisie. Nous détaillerons la

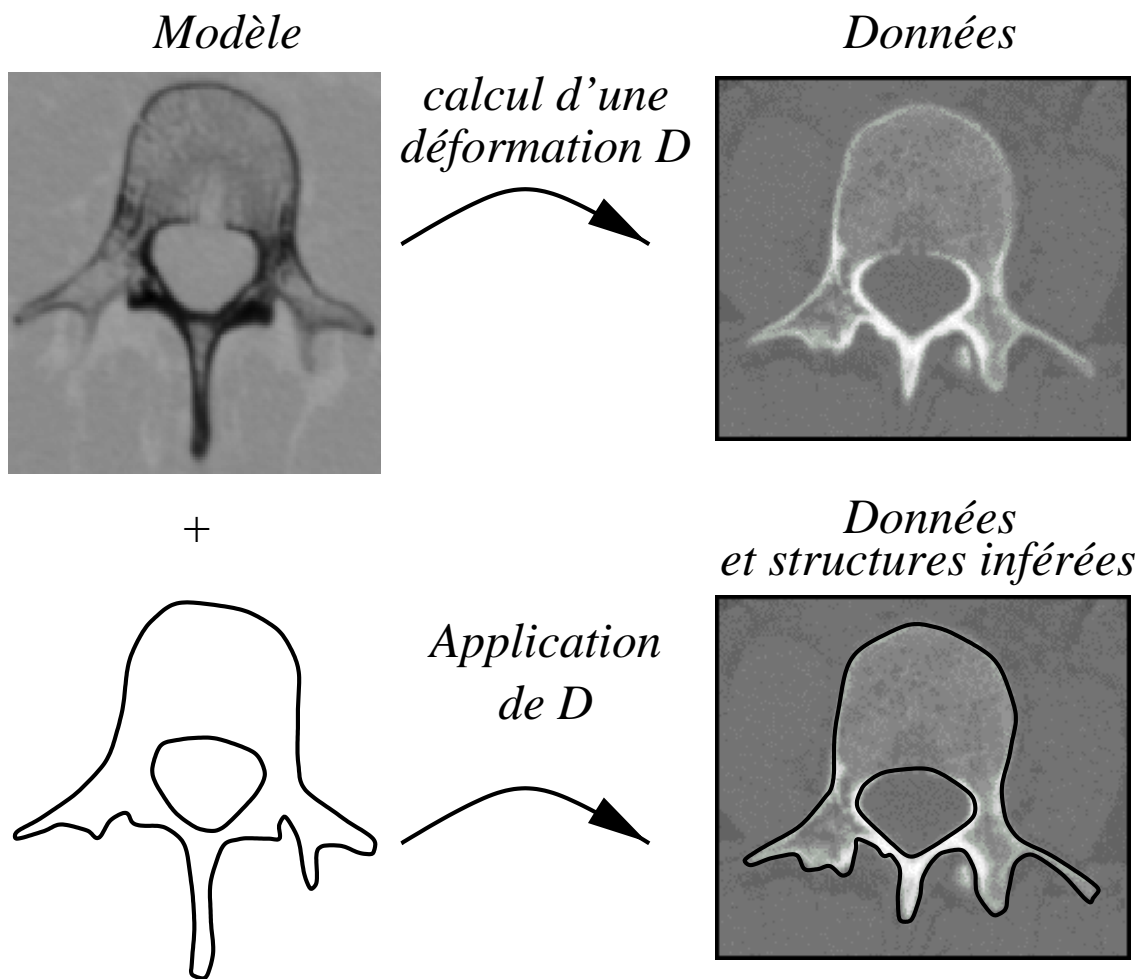


FIG. IV●1.3: La déformation  $D$  qui est calculée à partir de la partie *image TDM* du modèle est appliquée à la partie *surface de la vertèbre* du modèle, pour inférer la surface de la vertèbre des données.

déformation de notre modèle, qui est fondée sur un octree-spline dans le chapitre IV●4. Nous expliquerons dans le chapitre IV●3 pourquoi nous avons choisi les arbres k-d pour avoir un calcul rapide de la distance en dimension  $k$  à un ensemble de points. La stratégie de minimisation de cette distance en fonction des paramètres de déformation du modèle sera dévoilée dans le chapitre IV●5. Nous présenterons enfin des applications en IV●6 avant de conclure cette partie avec le chapitre IV●7.

# Chapitre IV•2

## Caractéristiques de bas niveau

### IV•2.1 Introduction

Dans notre étude bibliographique, nous avons présenté au chapitre I•2 une revue des différentes caractéristiques qui pouvaient être extraites des images afin de guider la déformation du modèle. Nous allons dans ce chapitre présenter les caractéristiques que nous avons retenues pour notre modèle déformable volumique. Il s'agit de contours enrichis de la direction du gradient. Nous justifierons ce choix dans le paragraphe IV•2.2, puis nous détaillerons les moyens de calculer ces caractéristiques.

### IV•2.2 Choix du vecteur gradient

Nous devons déterminer quelles caractéristiques conviennent à notre objectif de segmentation par modèle déformable volumique. Le choix de la courbure est intéressant, mais il faut des images de très bonne qualité pour que les dérivées secondes soient fiables. De même pour les lignes de crête. L'information sur la région d'appartenance du voxel, ou sur les régions qu'il sépare s'il s'agit d'un point de contour est pertinente, mais elle nécessite un traitement supplémentaire pour segmenter l'image en régions. Le profil de niveau de gris met en œuvre des comparaisons complexes, et une distance de dimension plus importante, qui sera longue à calculer.

Nous avons choisi une caractéristique simple, le vecteur gradient des niveaux de gris de l'image. Son calcul se fait en même temps que l'extraction des points de contour, et elle ne mène pas à une distance généralisée de trop grande dimension.

### IV•2.3 Calcul du gradient 3D

Soit une image  $I$  représentée par la fonction  $I(x, y, z)$  (voir figure IV•2.1).

Nous utilisons des filtres séparables pour calculer les dérivées partielles lissées [Monga et Benayoun 1995]. Pour cela, deux fonctions unidimensionnelles sont utilisées comme des filtres de convolution.

$$f_0(x) = c_0(1 + \alpha|x|)e^{-\alpha|x|} \quad (\text{IV}\bullet\text{2.1})$$

$$f_1(x) = c_1x\alpha^2e^{-\alpha|x|} \quad (\text{IV}\bullet\text{2.2})$$

$f_0$  est un filtre de lissage, et  $f_1$  donne la dérivée première lissée. Les coefficients de normalisation  $c_0$  et  $c_1$  sont donnés par les équations suivantes.



FIG. IV●2.1: un exemple d'image, il s'agit d'une coupe dans un volume TDM de vertèbre

$$c_0 = \frac{(1 - e^{-\alpha})^2}{1 + 2e^{-\alpha}\alpha - e^{-2\alpha}} \quad (\text{IV}●2.3)$$

$$c_1 = \frac{-(1 - e^{-\alpha})^3}{2\alpha^2 e^{-\alpha}(1 + e^{\alpha})} \quad (\text{IV}●2.4)$$

En notant les dérivées partielles avec des indices, on obtient les dérivées partielles lissées par les équations suivantes.

$$I_x = (f_1(x)f_0(y)f_0(z)) * I \quad (\text{IV}●2.5)$$

$$I_y = (f_0(x)f_1(y)f_0(z)) * I \quad (\text{IV}●2.6)$$

$$I_z = (f_0(x)f_0(y)f_1(z)) * I \quad (\text{IV}●2.7)$$

La dérivée partielle lissée selon  $x$  est donc le résultat d'une dérivation suivant la direction  $x$  suivie de deux opérations de lissage, suivant la direction  $y$  puis suivant la direction  $z$ . Les filtres  $f_0$  et  $f_1$  sont implémentés à l'aide de filtres récurrents, qui utilisent des masques de convolution [Deriche 1990 ; Monga et al. 1991].

## IV●2.4 Extraction des extrema

L'article [Monga et al. 1991] propose une méthode pour extraire les maxima locaux de la norme du gradient suivant la direction du gradient. Il s'agit de comparer la valeur de la norme du gradient en chaque point  $M$  avec celles en deux points  $N$  et  $Q$ . Ces points sont situés dans la direction du gradient sur les faces des cubes voisins du voxel  $M(i, j, k)$ .  $N$  est dans le sens du gradient,  $Q$  dans le sens inverse. Les valeurs du gradient en ces deux points sont obtenues par interpolation linéaire des valeurs en  $N_i$  et  $Q_i$ , qui sont des sommets de facettes des cubes voisins de  $M$ . La figure IV●2.2 illustre cela.

Le point  $M$  est déclaré correspondre à un maximum local de la norme du gradient suivant la direction du gradient si :

$$\|\mathbf{Grad}(M)\| \geq \|\mathbf{Grad}(N)\|$$

et  $\|\mathbf{Grad}(M)\| \geq \|\mathbf{Grad}(Q)\|$

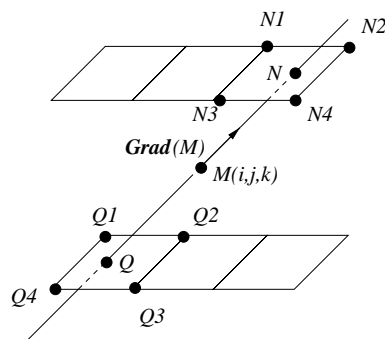


FIG. IV•2.2: Extraction des extrema locaux

## IV•2.5 Seuillage par hystérésis

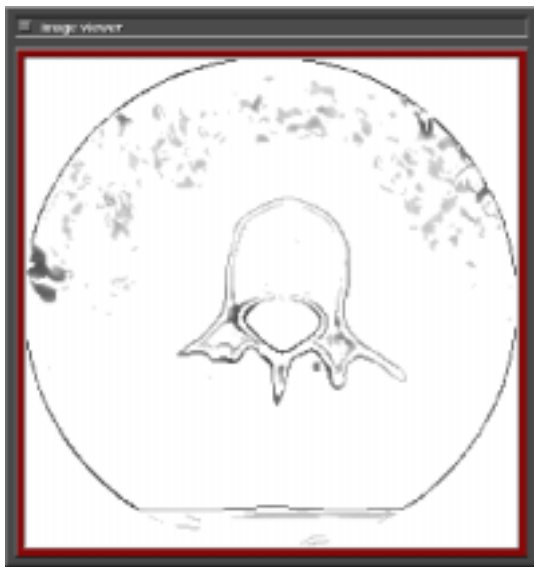
Parmi les extrema locaux que nous avons calculés, nous éliminons les points dont la norme du gradient est faible. Il s'agit d'abord des points  $M$  dont la norme du gradient  $\|\mathbf{G}\|$  est inférieure à un seuil bas  $S_b$ . Les points restants sont organisés en composantes connexes 3D. Tous les points des composantes connexes qui ne contiennent aucun point dont la norme du gradient est supérieure à un seuil haut  $S_h$  sont également éliminés. Nous conservons donc les points dont la norme du gradient est supérieure à  $S_b$  et qui peuvent être reliés à un point dont la norme du gradient est supérieure à  $S_h$  par un chemin connexe de points dont la norme du gradient est supérieure à  $S_b$ .

L'opération totale de seuillage s'opère de manière très simple, en utilisant un filtrage analogue à celui qui permet de calculer la distance du chanfrein (voir III•4.2 page 90). Un balayage en arrière, et un autre en avant par des demi-filtres suffisent pour cette opération. En un voxel de valeur  $V_i$ , le filtre va renvoyer 0 si  $V_i$  est inférieure à  $S_b$ , et le maximum des valeurs des voxels du filtre si ce maximum est supérieur à  $S_h$ , la valeur  $V_i$  sinon.

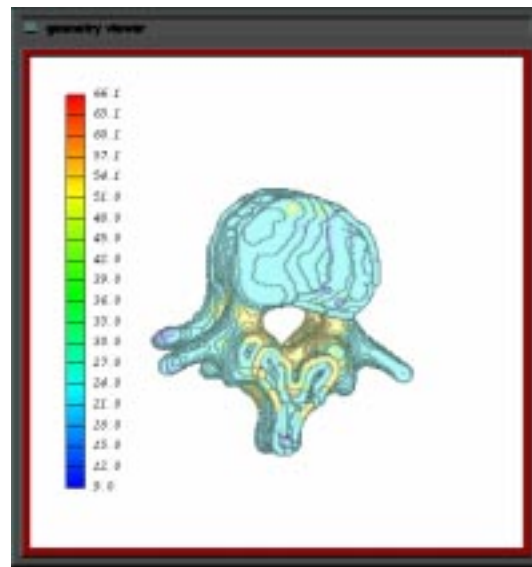
## IV•2.6 Influence du paramètre $\alpha$

Le paramètre  $\alpha$  correspond à l'inverse de la taille du filtre de lissage utilisé dans l'opérateur de calcul du gradient. Plus la valeur de  $\alpha$  est petite ( 0.25 par exemple), plus le lissage est important. Seuls les contours les plus significatifs seront détectés, mais du fait du lissage, leur localisation sera peu précise. Pour une valeur de  $\alpha$  importante (2.0 par exemple), le lissage de l'image est faible, et les contours détectés seront bien localisés. En contrepartie, même de petits contours liés au bruit de l'image apparaîtront alors. Comme valeur de compromis, nous avons choisi d'utiliser une valeur de  $\alpha$  égale à 0.85.





(a) Coupe du volume TDM après calcul des gradients 3D et seuillage par hystérésis



(b) Visualisation 3D des points de fort gradient. Les images sont nettoyées en enlevant les points à l'extérieur d'un certain cylindre.

FIG. IV●2.3: Résultat de l'extraction des caractéristiques de bas niveau. Dans l'image (b), chaque point est représenté par une petite sphère.

Après avoir extrait les caractéristiques de bas niveau dans les images, il est nécessaire d'avoir un moyen pour calculer l'énergie externe, c'est-à-dire d'avoir une mesure qui relie le modèle et les données pour une déformation donnée, afin de guider cette déformation. Dans notre cas, il s'agit de la mesure de la distance des points des données au modèle, comme nous l'avons vu au chapitre IV●1.

Nous allons présenter dans ce chapitre un moyen efficace pour calculer cette distance : les arbres k-d.

## IV●3.1 Utilisation optimisée des arbres k-d

Un arbre k-d est une généralisation de l'arbre binaire, dont la construction est en  $O(n \log n)$  opérations. La taille d'un arbre k-d est en  $O(n)$ . La complexité moyenne de la recherche du point le plus proche dans l'arbre k-d pour un point est en  $O(\log n)$ .

### 3.1.1 Construction d'un arbre k-d

Détaillons la construction d'un arbre k-d pour un ensemble de  $n$  points  $x^i, i = 1 \dots n$  de dimension  $k$ .

#### 3.1.1.1 Les arbres k-d classiques

Les points sont triés suivant chacune de leurs coordonnées, et la racine est la médiane des points, suivant le premier axe. Les fils de la racine seront les médianes respectives des deux sous-ensembles de points selon le deuxième axe. Et ainsi de suite chaque nœud non terminal divise l'espace en deux demi-espaces par un hyperplan qui est orthogonal à un des  $k$  axes de coordonnée.

Un nœud de l'arbre est donc représenté par l'indice d'un point et une dimension de l'espace. L'arbre est construit en considérant les  $k$  dimensions de manière circulaire. La figure IV●3.1 montre un exemple d'arbre 2-d. Dans les nœuds de l'arbre, 0 indique une séparation en  $x$ , et 1 une séparation en  $y$ .

#### 3.1.1.2 Les arbres k-d optimisés de Friedman

Friedman et al. [Friedman et al. 1977] ont proposé une modification des arbres k-d qu'ils ont appelée arbres k-d optimisés, qui permet une recherche avec retour arrière dont

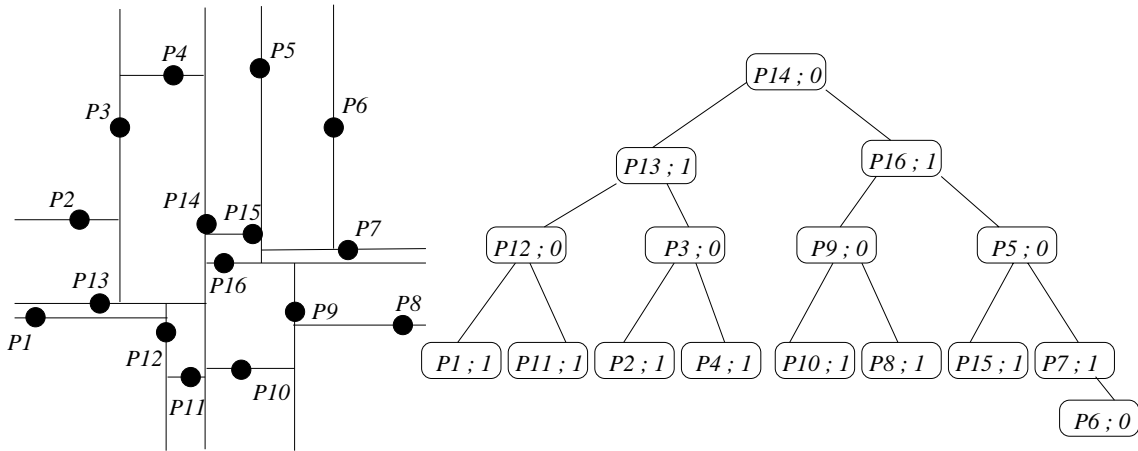


FIG. IV.3.1: Arbre 2-d d'un ensemble de points 2-d

la complexité moyenne est en  $O(\log n)$ , mais la limite de la complexité moyenne a une borne en  $2^k$ . Leur optimisation ne tient pas compte de la distribution des données.

Au lieu de découper l'espace suivant chacune des dimensions successivement, ils choisissent la dimension qui a la plus grande extension. En effet si la distance que l'on considère donne la même importance à toutes les dimensions, la distance moyenne à la partition opposée sera la plus importante si le partitionnement s'est fait sur la clef dont l'étendue des valeurs était la plus importante. Ils augmentent donc la probabilité d'éviter la recherche dans cette partition opposée.

### 3.1.2 Recherche dans un arbre k-d

#### 3.1.2.1 Recherche classique

Les premiers algorithmes de recherche [Preparata et Shamos 1986] dans un arbre k-d utilisaient une boîte de taille fixe pour réduire l'espace de recherche par une méthode de type aiguiller-et-borner (branch-and-bound). Le rayon  $dmin$  de la boîte est un paramètre de la recherche. Si  $dmin$  est trop petit, il y a un risque que le point le plus proche ne soit pas trouvé. Si au contraire il est trop grand, les branches de l'arbre seront peu élaguées, et la recherche sera très longue. On doit alors ajuster  $dmin$  en fonction des données. C'est ce que propose Zhang [Zhang 1992], qui prend en compte la moyenne et l'écart-type des distances  $dist_i$  dans son calcul. Mais cette solution reste grossière.

#### 3.1.2.2 Recherche de Friedman

Friedman a présenté dès 1977 un algorithme de recherche qui consiste à rechercher d'abord un voisin du vecteur  $x$  qui fait office de plus proche voisin courant, et à déterminer le vrai plus proche voisin en cherchant dans les compartiments qui intersectent la boule du plus proche voisin courant.

#### 3.1.2.3 Recherche plus efficace

Nous utilisons une amélioration de l'algorithme classique proposée par Friedman [Friedman et al. 1977] dans laquelle la dimension de la boîte se réduit avec les points trouvés

au cours du parcours. La recherche est accélérée en réduisant, lors du parcours de l'arbre,  $dmin$  à la distance au point le plus proche trouvé parmi les points examinés. Seule la valeur initiale de  $dmin$  influe sur l'étendue de la recherche. En effet si l'on est sûr que la distance recherchée est inférieure à  $dmin$ , on parcourra moins de nœuds que si l'on fixe  $dmin$  à la distance à la racine de l'arbre k-d.

Voici la procédure de recherche du point le plus proche du point  $P$  dans un nœud  $v$  d'un arbre k-d.  $dmin$  est la distance au point le plus proche  $Ppp$  trouvé jusqu'à présent.  $dv$  est la dimension du nœud  $v$  (cf la définition d'un nœud en 3.1.1.1).  $Pv$  est le point par lequel passe le nœud  $v$ .  $P[dv]$  est la coordonnée  $d$  de  $P$ .  $Pv[dv]$  est la coordonnée  $dv$  de  $Pv$ .  $fg$  est le fils gauche de  $v$ .  $fd$  est le fils droit de  $v$ .

On remarque qu'un premier test qui est une simple comparaison unidimensionnelle permet d'éviter le coûteux calcul de la distance k-d si les composantes de  $Pv$  et  $P$  sur l'axe  $dv$  sont éloignées de plus de  $dmin$ .

Si ensuite la distance k-d entre  $P$  et  $Pv$  est inférieure à  $dmin$ , c'est que  $Pv$  est le plus proche des points que l'on a trouvé pour le moment.

```

procedure cherche_n\oe ud(v, P, Ppp, dmin)
  si (|Pv[dv]-P[dv]| <= dmin) {
    si (distance_kd(P,Pv) < dmin){
      Ppp =Pv
      dmin=distance_kd(P,Pv)
    }
  }
  si ((P[dv]-dmin<Pv[dv]) et que fg est non vide)
    cherche_n\oe ud(fg,p,Ppp,dmin)
  si ((Pv[dv]-dmin<P[dv]) et que fd est non vide)
    cherche_n\oe ud(fd,p,Ppp,dmin)
}

```

Comme le parcours de l'arbre se fait en profondeur d'abord, on commence en fait par l'exploration du fils dans le demi-espace duquel se trouve le point  $P$ . En effet une mise à jour de la valeur de  $dmin$  lors de l'exploration de cette première branche peut éviter l'exploration de l'autre demi-espace.

Les figures suivantes illustrent la recherche du point le plus proche du point  $M$  parmi les points  $Pi$ . La figure IV●3.2 montre comment à partir d'une distance  $dmin0$  initiale la distance  $dmin$  évolue en  $dmin1$  et  $dmin2$ . La figure IV●3.3 montre que la distance  $dmin$  vaut  $dmin3$ , et que le point le plus proche de  $M$  dans l'arbre 2-d est  $P9$ . Cette figure montre également que comme la différence entre l'ordonnée de  $P16$  et celle de  $M$  est inférieure à  $dmin3$ , le sous-arbre de racine  $P5$  va être exploré. Le parcours total de l'arbre est illustré par la figure IV●3.4. Les \* correspondent aux calculs de distance.

### 3.1.3 Recherche itérée dans un arbre k-d

Notre algorithme de déformation étant itératif, nous devons calculer à chaque itération le point le plus proche dans le modèle de chacun des points des données. Nous utilisons la distance au précédent point le plus proche  $Ppp_{Pre}$ .

$$dmin = distance_{kd}(P, Ppp_{Pre})$$

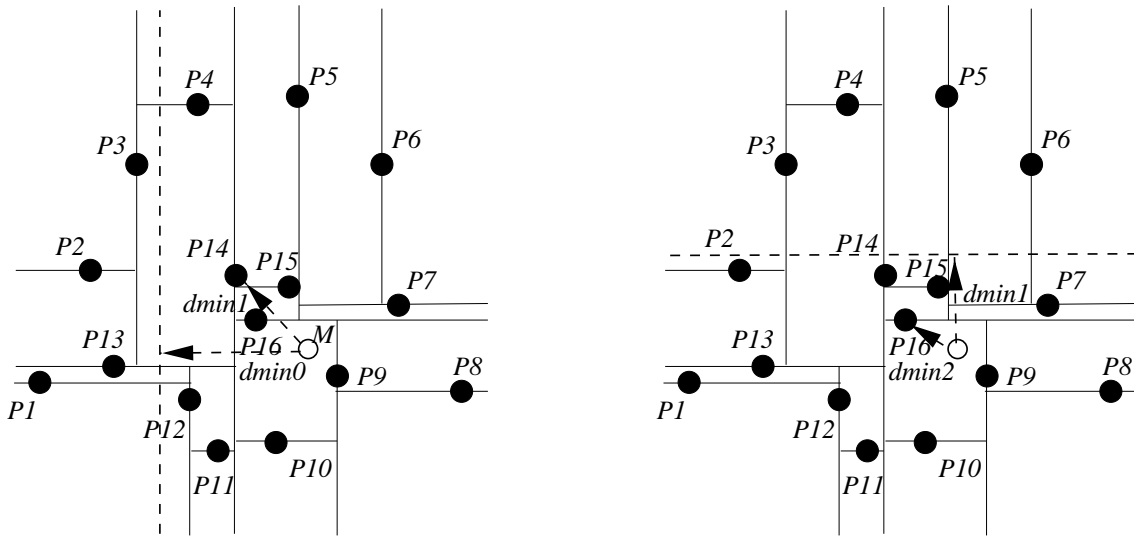


FIG. IV•3.2: Recherche dans un arbre 2-d (début)

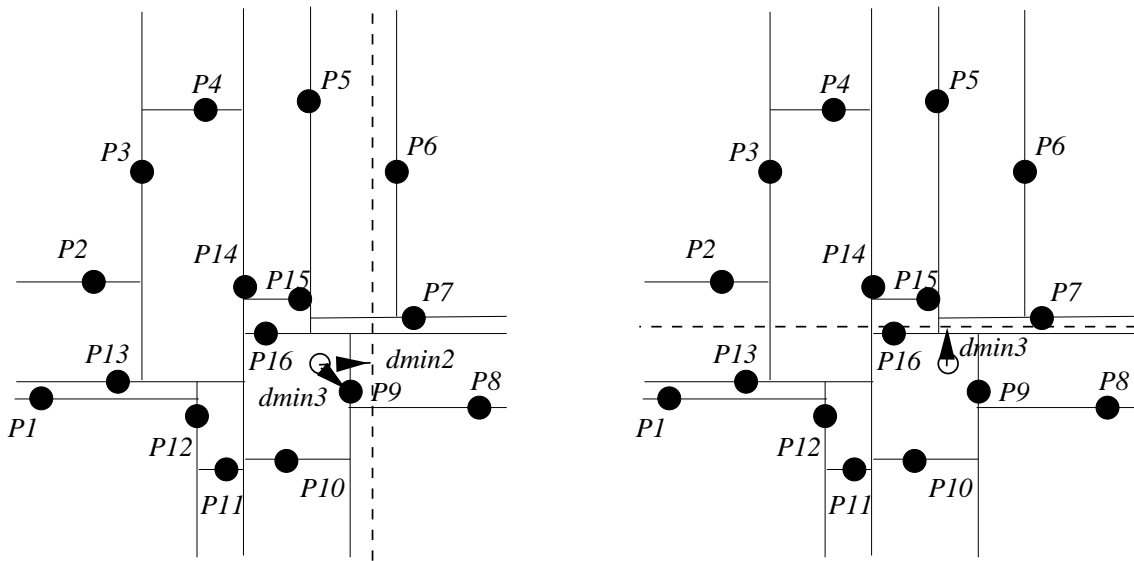


FIG. IV•3.3: Recherche dans un arbre 2-d (suite)

pour borner initialement la recherche dans l'arbre k-d. L'algorithme ainsi initialisé garantit de trouver le point le plus proche.

La première idée est que le point le plus proche n'a peut-être pas changé. Soit  $PP$  le point le plus proche de  $P$ . Si chaque point le plus proche connaît la distance à son point le plus proche à lui  $PPP$ , l'inégalité triangulaire indique que  $PP$  est toujours le point le plus proche de  $P$  si :

$$d(P, PP) \leq \frac{d(PP, PPP)}{2}$$

Nous faisons en plus un deuxième choix, qui permet d'accélérer la recherche mais ne permet plus de garantir l'obtention exacte du point le plus proche.

Si d'une itération à l'autre un point des données  $D$  ne s'est pas trop déplacé, son point le plus proche a une grande probabilité de rester le même. On ne recalcule réellement le

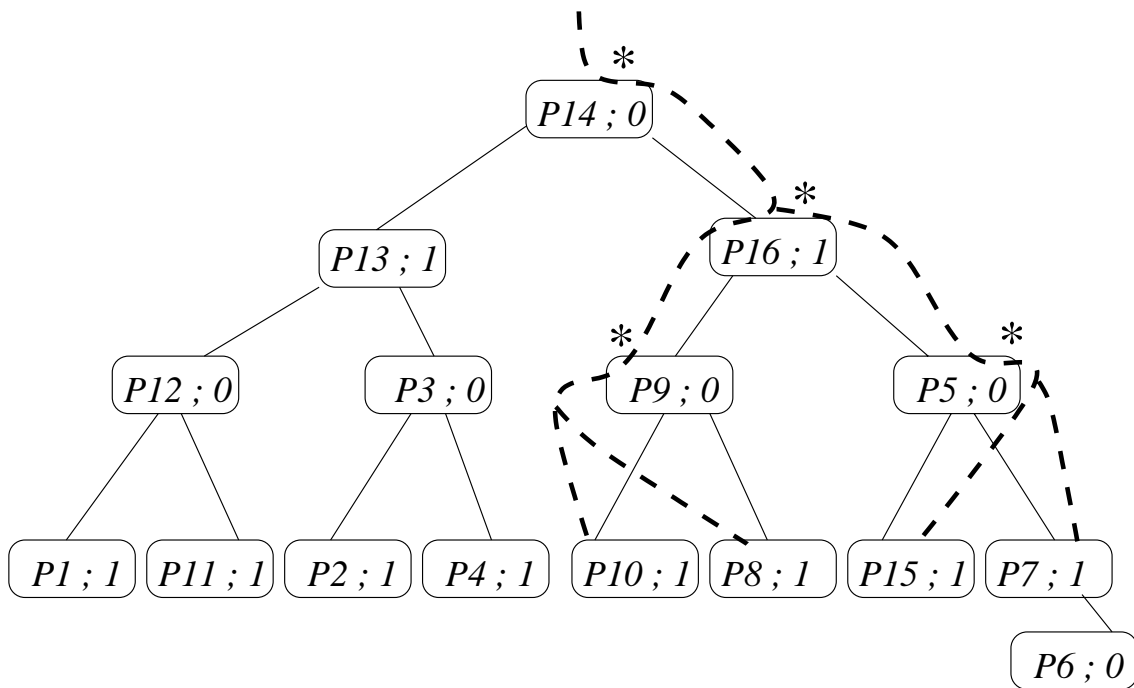


FIG. IV●3.4: Recherche dans un arbre 2-d (fin)

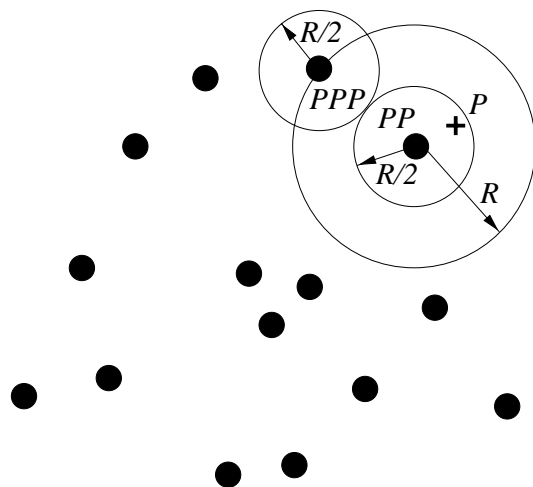


FIG. IV●3.5: PP est le point le plus proche de tout point P tel que  $d(P, PP) \leq \frac{d(PP, PPP)}{2}$

point le plus proche que si la distance entre  $P$ , et sa position précédente à laquelle on a effectivement calculé le point le plus proche dépasse un certain seuil.

De manière exacte, le point le plus proche de  $D$  reste le même si  $D$  reste dans le voisinage de Voronoi de ce point.

## IV•3.2 Nature de la distance

### 3.2.1 Distance généralisée

Soient  $P = (x, y, z, g_x, g_y, g_z)$  et  $PP = (x_{PP}, y_{PP}, z_{PP}, gx_{PP}, gy_{PP}, gz_{PP})$ , l'expression de la distance que nous avons utilisée est :

$$d(P, PP) = ((x - x_{PP})^2 + (y - y_{PP})^2 + (z - z_{PP})^2 + \frac{1}{\beta}((g_x - gx_{PP})^2 + (g_y - gy_{PP})^2 + (g_z - gz_{PP})^2))^{\frac{1}{2}}$$

avec  $\beta$  coefficient de normalisation. Il s'agit d'un cas particulier de la distance généralisée utilisée par Feldmar [Feldmar 1995].

### 3.2.2 Pondération de la distance

L'équation IV•1.1 permet de pondérer les valeurs des distances des points  $\mathbf{r}_i$  au modèle  $\mathcal{M}$ . On peut utiliser la qualité de la corrélation, ou l'incertitude que l'on a sur les points. Cette pondération reste à étudier, nous avons fait le choix de prendre  $\sigma_i = 1$ .

### 3.2.3 Distance fine

Lorsque les points des données sont assez proches du modèle, la discrétisation de celui-ci devient gênante. En effet, référons-nous à la figure IV•3.6 : si le point  $M$  se trouve à la frontière du diagramme de Voronoï entre les points  $P_i$  et  $P_{i+1}$ , le choix d'un de ces deux points peut être un facteur d'instabilité.

Or si nous considérons que le gradient des niveaux de gris est une approximation de la normale à la surface de l'objet sur laquelle se trouvent le point  $P_i$  et ses voisins, nous pouvons approximer cette surface par un morceau de plan passant par  $P_i$  et de vecteur normal  $\mathbf{Grad}(P_i)$ . Le point le plus proche de  $M$  sur le modèle sera, dans le cadre de cette approximation, le projeté orthogonal de  $M$  sur ce plan.

Ainsi nous remplaçons le point le plus proche calculé par ce point projeté.

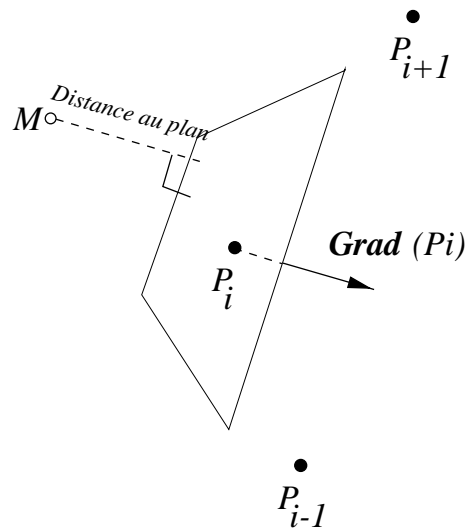


FIG. IV•3.6: Illustration de la distance au plan

### 3.2.4 Gradient de la distance

Une fois le point le plus proche trouvé, les coordonnées de celui-ci sont utilisées pour calculer le gradient spatial de la distance, dont nous avons besoin pour la minimisation.

Notons que le calcul d'une distance généralisée sert à établir de manière robuste pour chaque point des données le point le plus proche dans le modèle. Nous avons montré qu'une distance 3D génère plus d'erreurs de correspondance. Lorsque nous calculons le gradient de la distance, cependant, nous considérons uniquement le gradient spatial à trois composantes. En effet le gradient sert à guider la déformation, et celle-ci se fait bien uniquement dans l'espace à trois dimensions.

## IV•3.3 Transformation des caractéristiques

Les caractéristiques que nous avons choisi d'associer aux points de contours ne sont pas invariantes par rotation/translation, ni a fortiori par une transformation non-rigide. Il faut donc calculer les effets de la déformation sur le vecteur gradient pour tous les points de l'image des données qui interviennent dans le calcul de la distance finale des données au modèle.



# Chapitre IV●4

## Déformation et déformabilité

Nous utilisons la déformation de type octree-spline développée par Szeliski et Lavallée [Szeliski et Lavallée 1994 ; Szeliski et Lavallée 1996]. Dans ce chapitre, nous allons expliquer comment se déforme le modèle volumique que nous avons choisi, et comment la déformation est régularisée.

### IV●4.1 Types de déformations

La déformation  $\mathbf{T}_p(\mathbf{d}_i)$  est obtenue par une combinaison de plusieurs types de déformations, qui évolue au cours du temps. Au début du processus, seule une transformation rigide est appliquée, pour approximer la translation et la rotation entre le modèle et les données. A cette transformation rigide sont ensuite ajoutées des déformations polynomi-ales globales, dont nous avons parlé en I●4.1. Des déformations locales sont enfin ajoutées.

#### 4.1.1 Déformations splines locales

Pour obtenir des déformations encore plus flexibles, nous utilisons une famille de splines à base de produits de tenseurs volumiques. [Sederberg et Parry 1986 ; Cinquin 1987 ; Bajcsy et Kovacic 1989]

$$\mathbf{T}_p^l(\mathbf{d}_i) = \mathbf{d}_i + \sum_{j,k,l} \mathbf{u}_{jkl} B_j(x_i) B_k(y_i) B_l(z_i), \quad (\text{IV}●4.1)$$

Les  $\mathbf{u}_{jkl}$  sont les coefficients de déformation de la spline. Ils comprennent le vecteur de paramètres  $\mathbf{p}$ .  $B_j$ ,  $B_k$ , et  $B_l$  sont les fonctions de base de la B-spline [Sederberg et Parry 1986]. Les vecteurs  $\mathbf{u}_{jkl}$  se trouvent sur les sommets des cubes de l'octree-spline (voir IV●4.3). Chaque fonction de base (une fonction polynomiale par morceaux) a un support d'étendue finie. Elle n'est non-nulle que dans l'intervalle  $x_{j-o} \leq x \leq x_{j+o}$ , où  $o$  est l'ordre de la spline, et où les  $x_j$ ,  $j = 0 \dots M_x$  forment une subdivision de chaque axe de coordonnées dans  $\text{Ref}_{\text{données}}$ . En conséquence, seuls un petit nombre de  $\mathbf{u}_{jkl}$  va contribuer à la valeur de  $\mathbf{r}_i$ , soit de façon équivalente, les matrices  $\mathbf{M}_i$  seront très creuses dans cette représentation. On peut considérer que les  $\mathbf{u}_{jkl}$  sont des estimations locales des déplacements nécessaires pour mettre en correspondance les deux modèles, avec en plus un lissage et une interpolation dus à l'action de la fonction spline.

Pour  $o = 1$ , on obtient une interpolation d'ordre un (trilinéaire) des déplacements sur les sommets des cubes. C'est cette valeur qui a été utilisée dans nos expérimentations.

## IV•4.2 Déformabilité

### 4.2.1 Regularisation

Pour contraindre les paramètres  $\mathbf{u}_{jkl}$  de la spline de déformation, nous utilisons la *régularisation* [Bajcsy et Kovacic 1989 ; Szeliski 1990]. Une forme générale de régularisation est

$$\mathcal{R}(\mathbf{u}) = \frac{1}{2} \sum_{m=0}^p w_m \mathcal{R}_m(\mathbf{u}) \quad (\text{IV}\bullet 4.2)$$

Où  $\mathbf{u}$  est la spline de déformation de dimension  $d$  (le second terme de IV•4.1), les coefficients  $w_m$  sont les poids, et

$$\mathcal{R}_m(\mathbf{u}) = \int \sum_{j_1+\dots+j_d=m} \frac{m!}{j_1! \dots j_d!} \left| \frac{\partial^m \mathbf{u}(\mathbf{x})}{\partial x_1^{j_1} \dots \partial x_d^{j_d}} \right|^2 d\mathbf{x}$$

est le stabilisateur d'ordre  $m$  [Szeliski 1989a]. Le stabilisateur d'ordre 0 est similaire à la norme des  $\mathbf{u}_{jkl}$ . Le stabilisateur d'ordre 1 pénalise les variations linéaires en  $\mathbf{u}(\mathbf{x})$ . En pratique, nous utilisons une combinaison linéaire des stabilisateurs d'ordre 0 et 1.

Pour calculer la fonction de coût discrète définie sur les  $\mathbf{u}_{jkl}$ , qui correspond à (IV•4.2), nous avons le choix entre deux approches. La première est d'utiliser les *éléments finis* [Cohen et Cohen 1993], qui nécessitent l'évaluation analytique de (IV•4.2) en utilisant (IV•4.1). Cette approche a l'avantage de mener à des mesures plus précises, mais requiert en contrepartie de résoudre des équations complexes.

C'est pourquoi nous avons choisi d'utiliser les *différences finies*. Nous approximations les intégrales par la moyenne des carrés des estimées des dérivées discrètes. Par exemple, pour  $\mathcal{R}_0$ , nous utilisons  $h^3 \sum_{jkl} \|\mathbf{u}_{jkl}\|^2$ , et pour  $\mathcal{R}_1$  nous utilisons

$$\begin{aligned} \mathcal{R}_1(\{\mathbf{u}_{jkl}\}) \approx h \sum_{jkl} & |\mathbf{u}_{j+1,k,l} - \mathbf{u}_{jkl}|^2 + |\mathbf{u}_{j,k+1,l} - \mathbf{u}_{jkl}|^2 \\ & + |\mathbf{u}_{j,k,l+1} - \mathbf{u}_{jkl}|^2, \end{aligned}$$

où  $h$  est la taille de chacun des cubes du domaine spline.

### 4.2.2 Régularisation adaptative

Les coefficients de régularisation sont actuellement uniformes dans tout le volume. Des coefficients différents permettent au modèle de se déformer plus ou moins facilement. Dans les zones où ils sont importants le modèle se déforme moins que dans les zones où ils sont faibles. Pour cela il faut connaître la déformabilité locale du modèle. Cette connaissance est, par exemple d'ordre statistique ou anatomique. Sur un modèle déformable de vertèbre, on peut supposer que le plateau vertébral se déforme moins que les apophyses. Cette idée pourra faire l'objet d'une étude ultérieure à ce présent travail.

Une première recherche menée par Jean Romanet [Romanet 1997] a considéré l'emploi de coefficients différents dans les trois dimensions de l'espace pour la correction d'images IRM, en partant du constat que la direction principale de distorsion de ces images correspond à celle du gradient de lecture de l'IRM.

Pour faire évoluer le nombre de degrés de liberté au cours du temps, nous n'avons pas choisi d'utiliser des coefficients de régularisation variables en fonction du temps, mais de mettre en œuvre des déformations hiérarchiques.

## IV●4.3 Déformations hiérarchiques par octree spline

### 4.3.1 Base hiérarchique

Les coefficients associés à la spline sont des vecteurs 3-D qui représentent le déplacement entre les repères modèle et données. L'octree spline est représenté sur une *base hiérarchique* [Yserantant 1986 ; Szeliski 1990].

Dans cette représentation (Figure IV●4.1), les valeurs des déplacements aux nœuds des niveaux fins (résolution élevée) sont ajoutées aux déplacements interpolés à partir des parents de ces nœuds dans les niveaux de résolution plus faible [Szeliski 1989b ; Szeliski et Lavallée 1994].

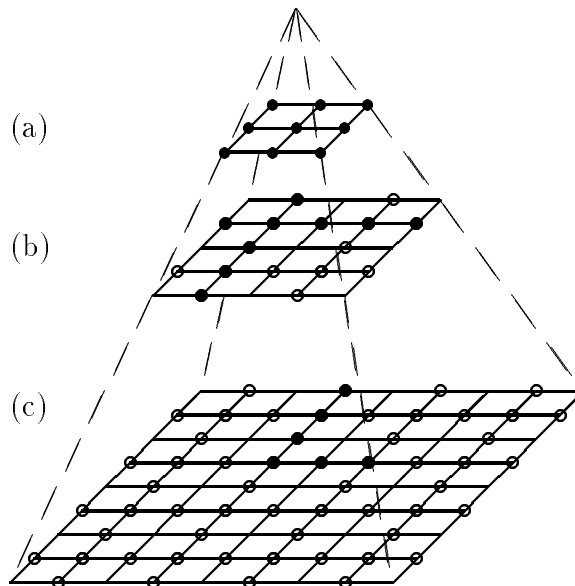


FIG. IV●4.1: Pyramide multi-résolutions. Les niveaux de résolution ((a) faible, (b) moyenne, (c) élevée) sont une représentation schématique de l'octree-spline (représenté en 2D pour plus de simplicité). Les cercles indiquent les nœuds dans la base hiérarchique. Les cercles pleins (●) sont des variables libres, correspondant à des nœuds qui peuvent se déplacer. Les cercles ouverts (○) sont des variables fixées à zéro, correspondant à des nœuds qui ne peuvent pas se déplacer.

La représentation est donc relative, elle permet d'accélérer la convergence de l'étape de minimisation et propage les corrections locales sur le domaine tout entier, ce qui lisse significativement la déformation obtenue.

### 4.3.2 Implémentation de l'octree spline

La base hiérarchique rend également possible l'implémentation de l'octree spline en fixant à zéro certains nœuds de cette base hiérarchique. Dans cette base, un nœud est libre



### 4.4.1 Calcul de la déformation inverse

Nous allons expliquer notre démarche en prenant comme exemple le cas de la correction de distorsions de l'IRM. Les images TDM sont déformées, et mises en correspondance avec les images IRM. Nous désirons appliquer la déformation inverse aux images IRM. Dans cet exemple, les images TDM constituent les données, et les images IRM le modèle.

L'algorithme est très simple : considérons un voxel  $v_{TDM}$  du volume TDM, en lui appliquant la déformation, on obtient un point  $p_{IRM}$  dans le volume IRM. La valeur de ce point est calculée par interpolation dans le volume IRM, et est affectée à  $v_{TDM}$  (Figure IV•4.3).

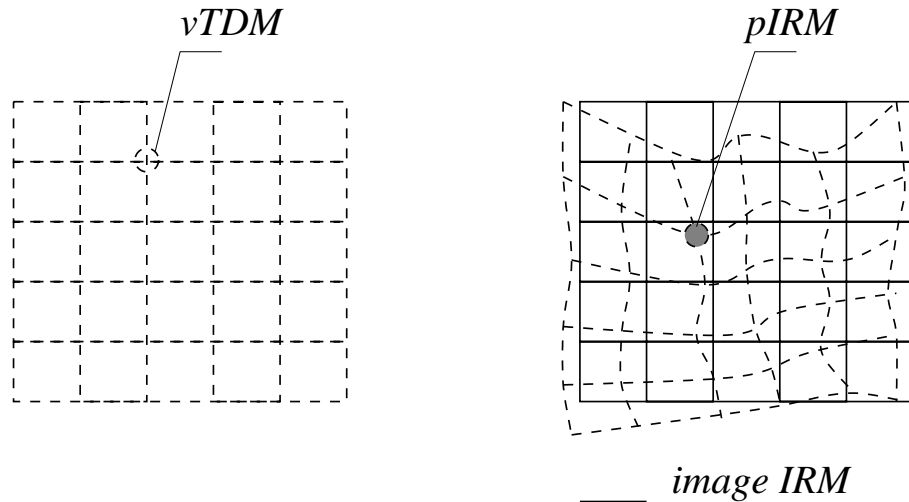


FIG. IV•4.3: Comment assigner une valeur à  $v_{TDM}$

### 4.4.2 Interpolation

Pour interpoler la valeur d'un point à partir de valeurs disposées sur une grille régulière, on utilise 4 points par dimension, qui sont placés sur les segments du maillage, et que nous représenterons comme des cercles blancs. En 2D, nous avons représenté 8 points blancs sur la figure IV•4.4. Les valeurs de ces points blancs sont elles-mêmes obtenues par interpolation unidimensionnelle sur la grille régulière, IRM dans notre exemple. Il faut 4 points qui se trouvent aux nœuds de la grille régulière (représentés en noir sur la figure IV•4.4) pour définir la valeur d'un point blanc.

Le calcul de la valeur en un point  $p_{IRM}$  prend donc en compte  $4^d$  valeurs, où  $d$  est la dimension. En 2D, le calcul nécessite 16 points noirs, et en 3D, 64 valeurs.

Pour la partie interpolation, nous avons adapté un logiciel écrit par Gelu Ionescu, qui implémente une interpolation par une fonction spline cubique de haute résolution  $S_a$ .

$$S_a(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & \text{si } x \in [-1, 1] \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & \text{si } x \in [-2, -1] \cup [1, 2] \end{cases} \quad (\text{IV}\bullet 4.3)$$

Le domaine de variation de  $a$  est  $[-2, 0]$ . Nous avons choisi  $a = .5$  car la fonction  $S_{.5}$  (figure IV•4.5) est celle qui a les meilleures propriétés. Cette fonction se révèle meilleure que la fonction qui calcule simplement le plus proche voisin, et qu'une fonction linéaire, ou une fonction B-Spline cubique : l'interpolation au plus proche voisin décale l'image. Les

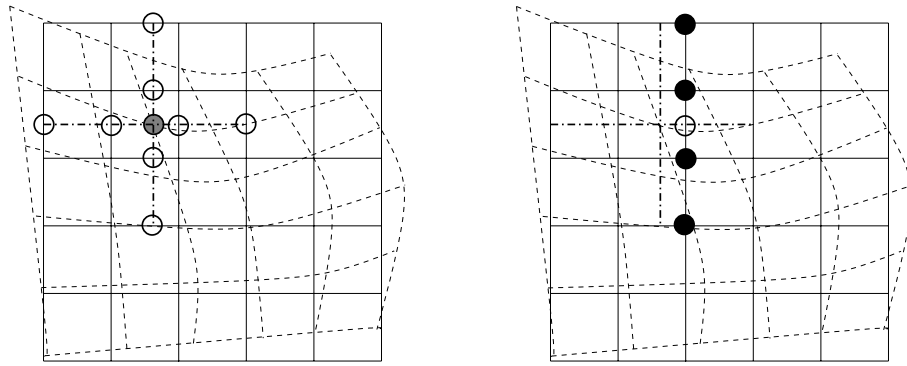


FIG. IV●4.4: Interpolation de la valeur d'un voxel en deux étapes

interpolations linéaire et B-Spline cubique rendent l'image plus floue. Une comparaison des méthodes d'interpolation pour le rééchantillonnage d'images peut être trouvée dans [Parker et al. 1996].

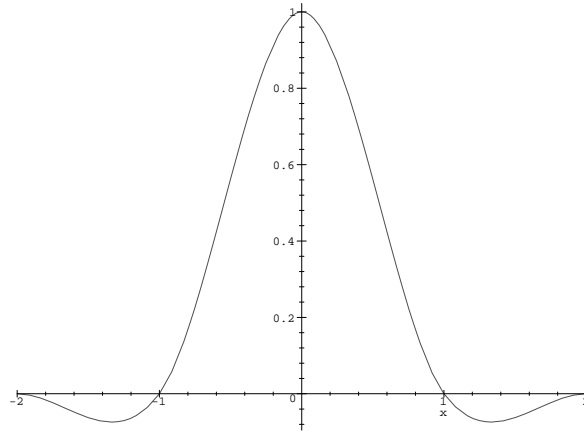


FIG. IV●4.5: Spline cubique haute résolution  $S_{-0.5}$

## IV●4.5 Conclusion

la déformation d'un volume est en général une opération coûteuse. L'utilisation d'un octree spline et sa représentation sur une base hiérarchique permettent cependant une souplesse pour s'adapter à des déformations locales particulières. La régularisation assure que la déformation est minimale. De plus, les bases hiérarchiques permettent une convergence rapide de l'algorithme de minimisation.

### IV•5.1 Minimisation aux moindres carrés

Pour cela, nous utilisons une méthode itérative dont le cœur est l'algorithme de Levenberg-Marquardt, combiné avec une descente de gradients conjugués preconditionnés qui opère sur une représentation de l'octree-spline dans des bases hiérarchiques.

#### 5.1.1 Algorithme de Levenberg-Marquardt

Pour mettre à jour l'estimation courante des paramètres  $\mathbf{p}^{(k)}$ , notre méthode nécessite l'évaluation de la fonction de distance  $dist(\mathbf{r}_i, \mathcal{M})$  et de ses dérivées par rapport aux paramètres. Nous avons présenté dans le chapitre IV•3 une technique efficace de calcul de la fonction  $dist$ , et de son gradient spatial  $\mathbf{g} = \nabla_{\mathbf{r}} dist$ .

L'évaluation des dérivées s'obtient par composition,

$$\frac{\partial dist_i}{\partial \mathbf{p}} = \frac{\partial dist_i}{\partial \mathbf{r}_i} \frac{\partial \mathbf{r}_i}{\partial \mathbf{p}} = \mathbf{g}_i \frac{\partial \mathbf{r}_i}{\partial \mathbf{p}} = \mathbf{g}_i \mathbf{M}_i \quad (\text{IV}\bullet 5.1)$$

où la transformation s'écrit  $\mathbf{r}_i = \mathbf{M}_i \mathbf{p}$ .

Une fois que les valeurs de la distance  $dist_i$  et de ses dérivées  $\partial dist_i / \partial \mathbf{p}$  ont été calculées pour tous les points des données, l'algorithme de Levenberg-Marquardt construit une approximation de la matrice Hessienne,  $\mathbf{A}$  et du vecteur gradient  $\mathbf{b}$  qui représente l'erreur. Dans notre cas, nous avons :

$$\mathbf{A} = \sum_{i=1}^N \frac{1}{\sigma_i^2} \frac{\partial dist_i}{\partial \mathbf{p}} \left( \frac{\partial dist_i}{\partial \mathbf{p}} \right)^T \quad \text{and} \quad \mathbf{b} = - \sum_{i=1}^N \frac{1}{\sigma_i^2} dist_i \frac{\partial dist_i}{\partial \mathbf{p}} \quad (\text{IV}\bullet 5.2)$$

Il calcule ensuite un incrément  $\Delta \mathbf{p}$  qui va rapprocher le vecteur paramètre du minimum local, en résolvant l'équation

$$(\mathbf{A} + \lambda \text{diag}(\mathbf{A})) \Delta \mathbf{p}^{(k)} = \mathbf{b}, \quad (\text{IV}\bullet 5.3)$$

$\lambda$  est un facteur de stabilisation qui varie au cours du temps [Press et al. 1992].

La matrice  $\mathbf{A}$  est une approximation de la matrice Hessienne, car les termes d'ordre deux en  $dist_i$  ne sont pas pris en compte. En effet, d'après [Press et al. 1992] ces termes peuvent être déstabilisants si le modèle converge mal, ou s'il est gêné par la présence d'outliers.

Après avoir effectué l'affectation  $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} + \Delta\mathbf{p}^{(k)}$ , le processus est répété, jusqu'à ce que  $\mathcal{C}(\mathbf{p})$  soit inférieur à un seuil fixé, ou que la différence entre les paramètres  $|\mathbf{p}^{(k)} - \mathbf{p}^{(k-1)}|$  lors de deux itérations successives soit inférieure à un seuil fixé, ou encore que le nombre maximum d'itérations soit atteint.

### 5.1.2 Importance de la représentation de l'octree spline

La résolution du système d'équations IV•5.3 est immédiate lorsque le nombre de paramètres à estimer est raisonnablement faible, comme c'est le cas pour les déformations rigides et globales. Nous pouvons alors utiliser l'algorithme d'élimination de Gauss-Jordan [Press et al. 1992].

Pour des systèmes ayant plus de paramètres, comme c'est le cas pour les déformations locales, il devient impossible de calculer explicitement la Hessienne  $\mathbf{A}$  (car cela prend une place en  $O(M^2)$ , où  $M$  est le nombre de paramètres), ou de résoudre IV•5.3 directement, ce qui nécessite  $O(M^3)$  opérations. Nous utilisons alors une unique itération d'un algorithme de gradient conjugué préconditionné [Press et al. 1992]. Le fait d'utiliser une représentation de l'octree spline sur une base hiérarchique accélère la convergence [Szeliski et Lavallée 1994].

## IV•5.2 Outliers

Une fois que l'algorithme de Levenberg-Marquardt a convergé, nous calculons une estimation robuste du paramètre  $\mathbf{p}$  en rejetant les points pour lesquels  $dist_i^2 \gg \sigma_i^2$  et en calculant quelques itérations de plus [Lavallée et al. 1991 ; Bittar et al. 1993]. Ce procédé réduit l'influence des outliers [Huber 1981].

De plus, à chaque itération, les points les plus distants du modèle sont considérés comme aberrants. Le seuil pour les rejeter doit être fixé en fonction de l'application ou de l'expérimentation. Nous l'estimons en prenant  $3\sigma$  où  $\sigma$  est la déviation standard moyenne a priori du bruit.

## IV•5.3 Minima locaux

Lorsque l'on utilise une technique de minimisation fondée sur les gradient comme Levenberg-Marquardt, il y a une possibilité d'échouer dans un minimum local. Pour limiter cette éventualité, nous appliquons une approche hiérarchique au sens de la complexité de la déformation, dans laquelle nous estimons d'abord la transformation la plus simple possible (une transformation rigide), puis nous estimons successivement des déformations plus complexes (affine, puis trilinéaire ou quadratique). Pour les déformations locales, nous commençons par une spline à une faible résolution (typiquement un simple cube), et nous utilisons ensuite les paramètres optimaux calculés à cette résolution pour initialiser les estimations des paramètres au niveau suivant.



Notre méthodologie de validation de notre modèle déformable comporte deux phases. Il s'agit dans un premier temps de tester notre modèle sur des exemples simples. Nous avons choisi des données synthétiques, et nous rendons compte de ces expérimentations dans la section IV●6.1. Dans un deuxième temps, la tâche est d'évaluer la capacité de notre modèle à segmenter une image, qui est l'application réelle que nous envisageons. Nous avons donc réalisé la segmentation d'une image 3D qui avait déjà été segmentée manuellement par un expert, afin de comparer les deux résultats. Nous rendons compte de cette expérience dans la section IV●6.2.

Une troisième phase sera à mettre en œuvre : valider la segmentation automatique sur un ensemble significatif de jeux de données. Cette phase est inscrite dans les perspectives de ce travail.

Nous avons déjà appliqué notre méthode dans deux domaines originaux, la création d'images échographiques virtuelles (IV●6.3) et la correction d'images IRM (IV●6.4).

### IV●6.1 Validation sur des données synthétiques

Notre première phase de validation opère sur des objets simples. Nous montrons sur des ellipsoïdes de synthèse que l'utilisation d'une distance 6D permet une convergence là où une distance 3D est mise en défaut. Pour être plus précis, le modèle et les données sont composés de points régulièrement échantillonnés sur deux ellipsoïdes de paramètres différents. Les caractéristiques de bas niveau sont les vecteurs normaux à la surface des ellipsoïdes, que nous prenons normalisés. La figure IV●6.1 présente le modèle et les données après l'étape de transformation rigide.

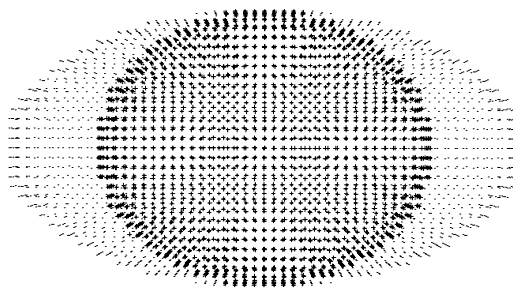


FIG. IV●6.1: Position après transformation rigide

La figure IV●6.2 (a) montre qu'avec une distance 3D, la convergence n'est pas complète, et on voit (figure IV●6.2(b)) qu'avec la distance 6D, les deux nuages se recouvrent bien.

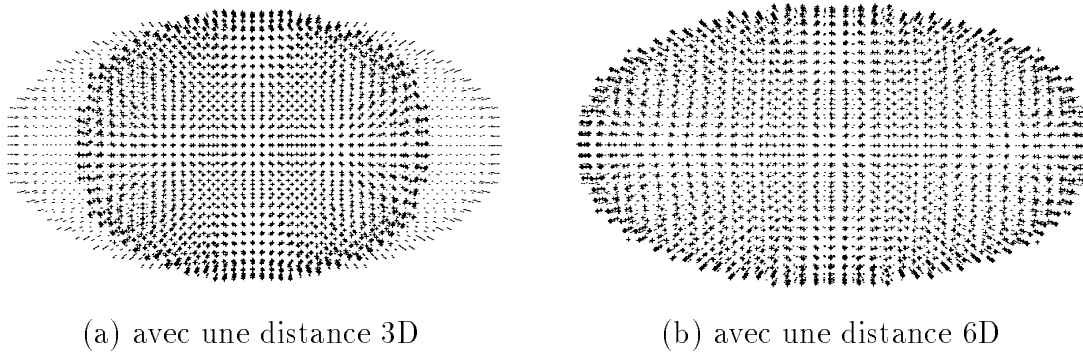


FIG. IV●6.2: Après les déformations locales

## IV●6.2 Segmentation d'une vertèbre dans une image TDM

Nous allons décrire les expérimentations que nous avons effectuées pour segmenter une image TDM de vertèbre à partir d'un modèle de vertèbre issu d'images TDM. Selon le principe d'inférence, nous avons utilisé les mêmes caractéristiques de bas niveau dans le modèle et dans l'image pour guider la déformation, et nous avons ensuite inféré la déformation des points de la surface du modèle.

### 6.2.1 Création du modèle

L'examen TDM que nous avons choisi pour construire notre modèle est composé de 47 images  $512 \times 512$ , que nous avons sous-échantillonnées en images  $256 \times 256$ . Il comprend une vertèbre lombaire complète et des parties de ses deux voisines. L'espacement inter-coupe est de 1 mm, l'épaisseur des coupes de 1 mm, la largeur de champ de 15 cm.

#### 6.2.1.1 Caractéristiques de bas niveau

Nous avons obtenu les éléments géométriques du modèle par la méthode décrite au chapitre IV●2. Les images d'illustration de ce chapitre (figures IV●2.1 et IV●2.3) étaient les images de ce modèle de vertèbres.

Les éléments géométriques calculés sur les vertèbres voisines de la vertèbre complète sont conservés. Le modèle est finalement constitué de 49848 points et de leurs vecteurs gradients. Nous montrons la visualisation de ces gradients figure IV●6.3. En (a), chaque vecteur gradient est représenté par un segment de droite issu du point du modèle correspondant. En (b) nous proposons un mode original de visualisation, la **visualisation par écailles** dans lequel chaque élément est représenté par une facette équilatérale triangulaire de barycentre l'élément et de vecteur normal le vecteur gradient.

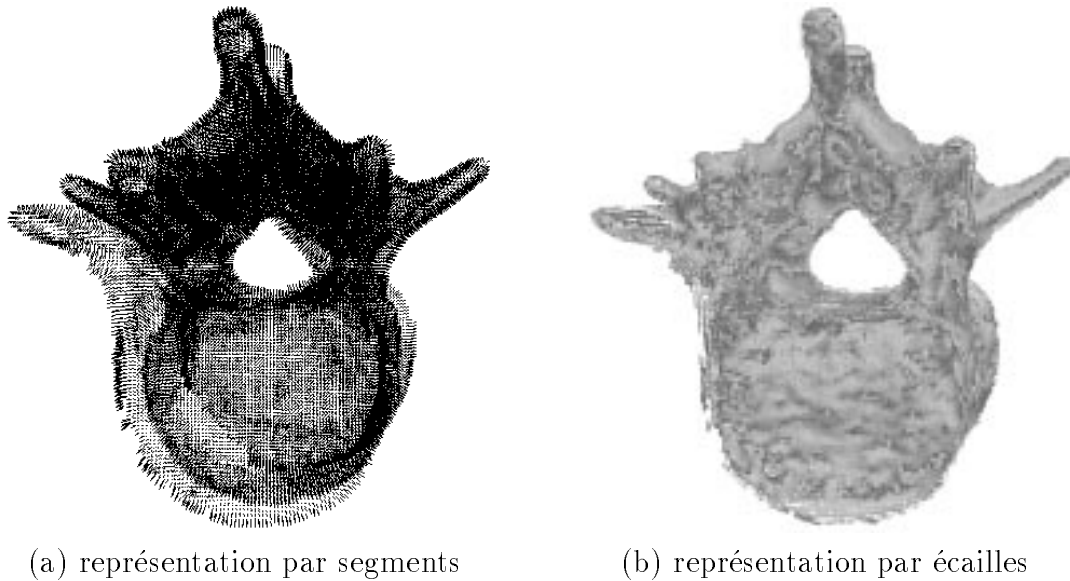


FIG. IV●6.3: Vues 3D des gradients associés aux éléments géométriques du modèle

### 6.2.1.2 Points de la surface

Les points de la surface de la vertèbre complète ont été segmentés à la main sur les images TDM modèle (figure IV●6.4).

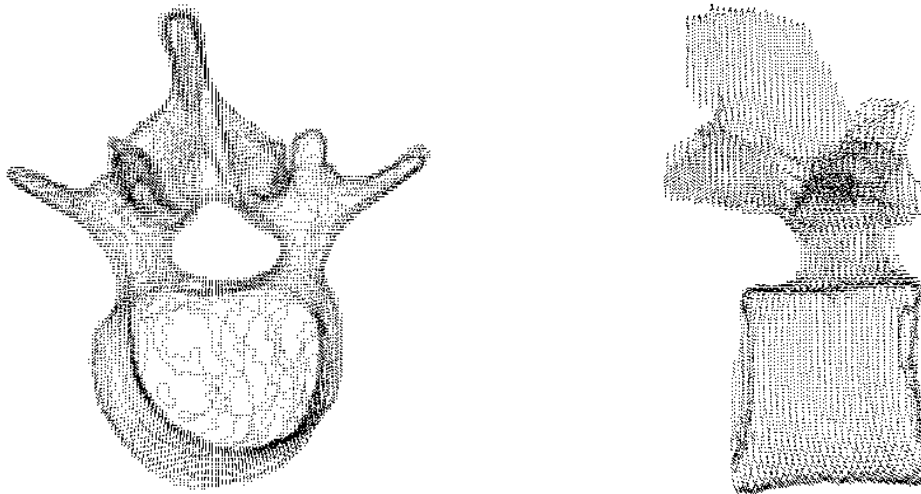


FIG. IV●6.4: Vues 3D des points segmentés à la main sur la vertèbre complète du volume TDM modèle

C'est la surface de cette vertèbre modèle qui sera utilisée par inférence pour segmenter les images des données. Dans cette phase de validation, les points de la surface sont constitués des contours de la vertèbre tracés sur les coupes 2D du volume TDM. Pour une utilisation clinique, il est nécessaire d'obtenir une meilleure représentation de cette surface, afin d'utiliser la surface déformée du modèle comme surface de l'objet contenu dans l'image des données. Cela pourra se faire à l'aide d'une représentation surfacique

comme celles que nous avons présenté dans le chapitre I•3. On pourra choisir par exemple le modèle des  $\delta$ -snakes.

## 6.2.2 Pré-segmentation des données

Les données sont constituées de 43 images TDM  $512 \times 512$ , sous-échantillonnées en images  $256 \times 256$ , d'une vertèbre lombaire L3 complète et de parties de ses voisines. L'espacement inter-coupes est de 1 mm, l'épaisseur des coupes de 1 mm, la largeur de champ de 12 cm. Il ne s'agit bien entendu pas du même patient que pour le volume du modèle, puisque nous désirons avoir des vertèbres différentes pour l'application de notre modèle déformable.

Les figures IV•6.5 et IV•6.6 illustrent l'extraction des 64256 caractéristiques de bas niveau des données.

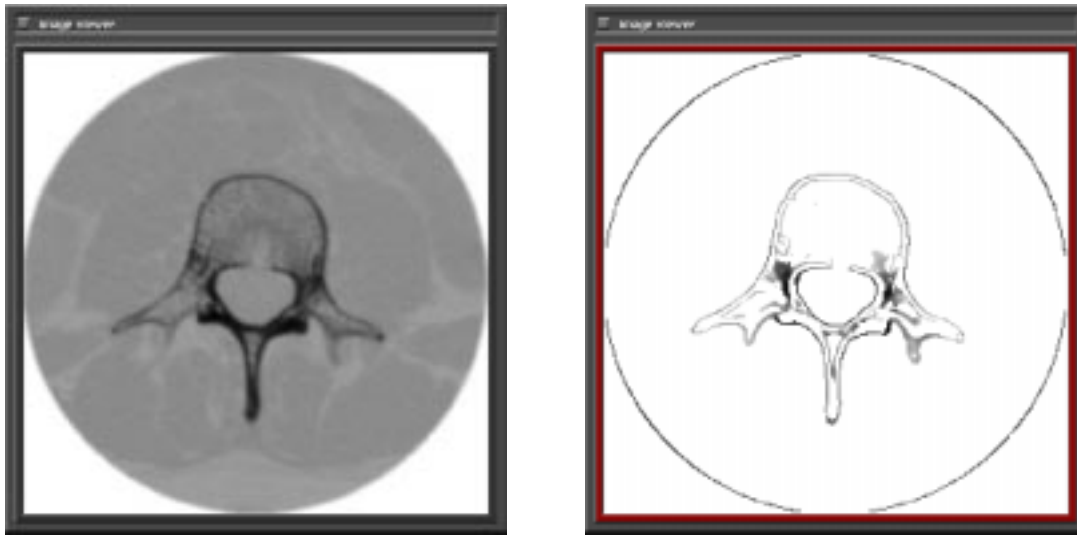


FIG. IV•6.5: Coupe dans le volume TDM des données, avant et après le calcul du gradient

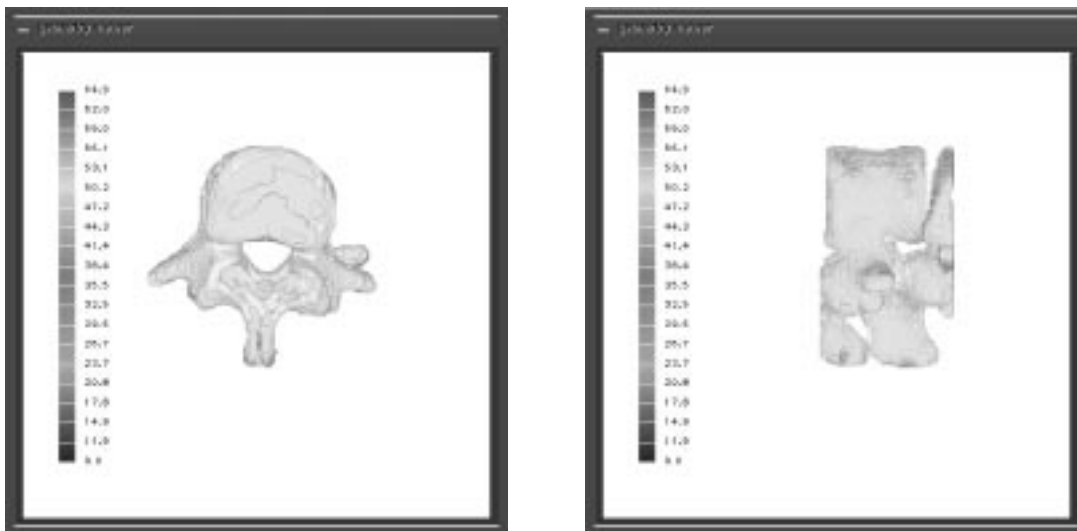


FIG. IV•6.6: Visualisation 3D des points obtenus après seuillage et nettoyage

### 6.2.3 Validation sur un exemple

Nous avons validé notre modèle déformable volumique utilisant une distance généralisée à travers cette application. Nous avons comparé les résultats obtenus avec notre calcul de distance itéré dans un arbre k-d avec d'une part le calcul de la distance au modèle par exploration systématique de tous les points du nuage modèle, et d'autre part avec l'algorithme utilisant une distance 3d précalculée dans une carte de distance octree spline.

Dans nos tests, nous avons utilisé 52400 points du nuage des données, soit environ 80% du nuage initial. Le tableau IV●6.1 présente de premiers éléments de comparaison entre les trois méthodes. Il apparaît que les erreurs moyennes en fin de convergence pour les deux méthodes de calcul de distance généralisée sont du même ordre. La différence importante entre les erreurs moyennes des distances 6d et 3d s'explique par la nature différente de la distance calculée. Ces valeurs ne sont pas issues des mêmes formules, et ne peuvent pas être comparées.

Nous avons comparé par contre les temps de calcul des trois méthodes, et nous nous apercevons que le calcul de distance 6d est 1.6 fois plus lent que le calcul de distance 3d. Le calcul exhaustif, lui, est 83 fois plus lent que le calcul 3d !

TAB. IV●6.1: Comparaisons quantitatives

méthode	temps de calcul (sec.)	temps de calcul (sec.)	erreur moyenne finale (mm)
distance itérée, arbre 6-d	280	160	2.15
distance exhaustive 6d	14549	8300	2.3
octree spline 3d	175	100	1.2

### 6.2.4 Déformation utilisant un calcul de distance 6D

#### 6.2.4.1 Distance itérée 6d dans un arbre k-d

La figure IV●6.7 montre en coupe - selon deux plans orthogonaux - la correspondance entre le modèle (points) et les données (petites croix). Les éléments qui apparaissent sur cette figure sont les voxels de gradient fort, qui ont guidé la déformation. On remarque les contours internes et les contours externes. Ces deux types de contours ne se mélangent pas, sauf peut-être sous l'apophyse gauche dans la coupe transversale. On note que les extrémités des apophyses transverses ne correspondent pas exactement.

Les points de surface apparaissent en coupe sur la figure IV●6.8. On note le bon placement des points du modèle relativement aux points des données, sauf aux extrémités des apophyses transverses.

#### 6.2.4.2 Calculs exhaustifs de la distance 6d

Le résultat de la déformation en utilisant un calcul de distance exhaustif (Figure IV●6.9) est visuellement comparable à celui obtenu avec la distance itérée (Figures IV●6.7 et IV●6.8).



(a) coupe transversale

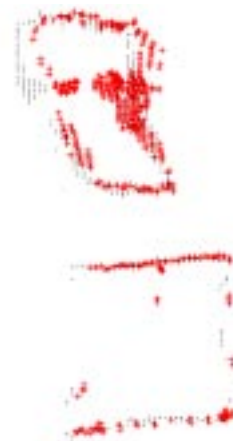


(b) coupe sagittale

FIG. IV●6.7: Éléments guidant la déformation (distance 6D).



(a) coupe transversale



(b) coupe sagittale

FIG. IV●6.8: Points de surface inférés par la déformation (distance 6D).



(a) éléments guidant la déformation, coupe transversale



(b) points de surface, coupe transversale

FIG. IV●6.9: Déformation obtenue en utilisant la distance 6d exhaustive.

### 6.2.4.3 Comparaisons quantitatives

Nous présentons l'évolution de l'erreur moyenne et du temps de calcul de chaque itération au cours du processus. Les 180 itérations se répartissent en 9 cycles de 20 itérations. Le premier correspond à la transformation rigide, le second aux déformations globales, et les suivants aux subdivisions successives de l'octree de déformation. L'écart type n'est recalculé qu'en début de cycle. Les courbes des erreurs ne sont pas strictement décroissantes, car l'algorithme de minimisation de Levenberg-Marquardt procède par essais et erreurs.

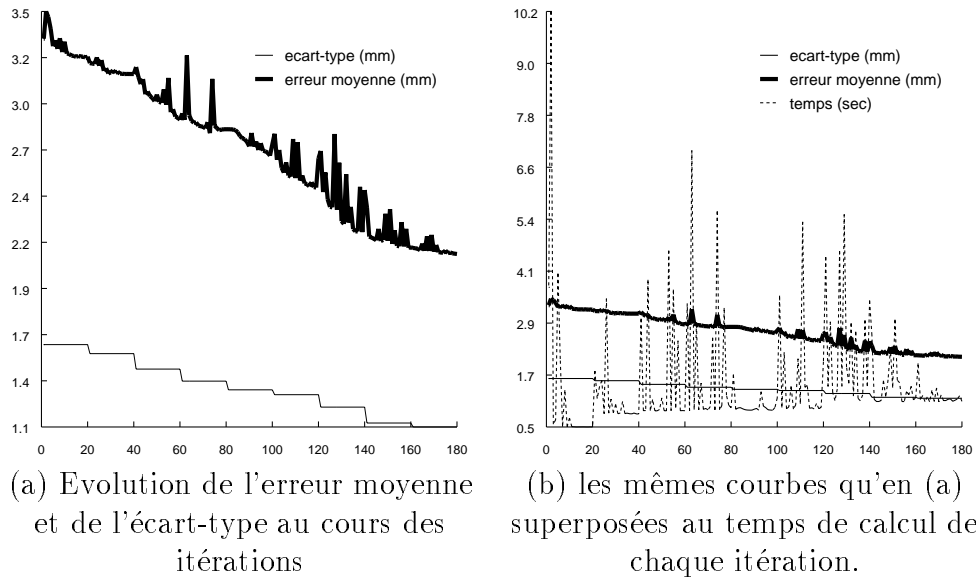


FIG. IV●6.10: Quantifications de l'erreur avec la distance  $6d$  itérée dans un arbre k-d.

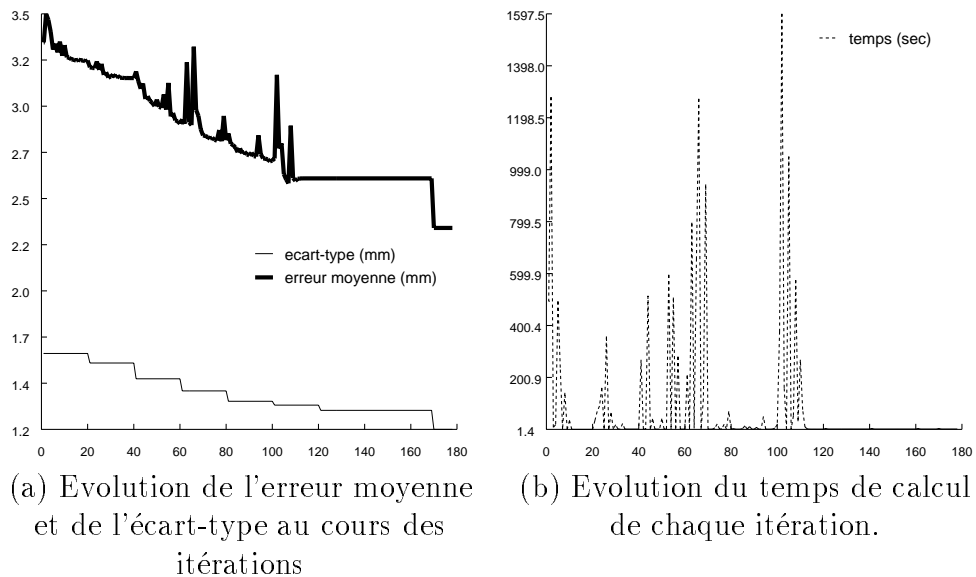


FIG. IV●6.11: Quantifications de l'erreur avec le calcul exhaustif de la distance  $6d$ .

La figure IV●6.12 montre que les histogrammes des erreurs finales pour les deux méthodes  $6d$  sont comparables.

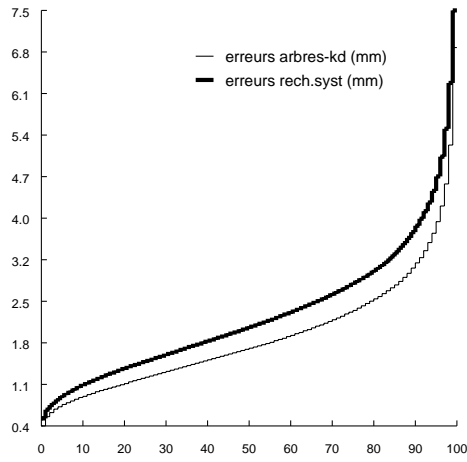


FIG. IV●6.12: Comparaison de l'histogramme des erreurs finales pour les deux méthodes 6d

#### 6.2.4.4 Analyse de l'algorithme de distance itérée

Les figures de cette sous-section comparées avec la figure IV●6.10 (b) montrent que le nombre moyen par point de parcours de nœuds dans l'arbre kd est lié avec le nombre moyen par point de calculs réels de la distance, et avec le temps de calcul. Le temps de calcul est très important au début du processus, pour la première itération, car le calcul du point le plus proche ne peut pas utiliser de point le plus proche précédent pour réduire le parcours de l'arbre. Le temps de calcul est très court quand de nombreux points conservent leur point le plus proche précédent. Les cycles de 20 itérations génèrent des pics de calculs. Entre 80 et 100 itérations, il y a une étrange zone plate.

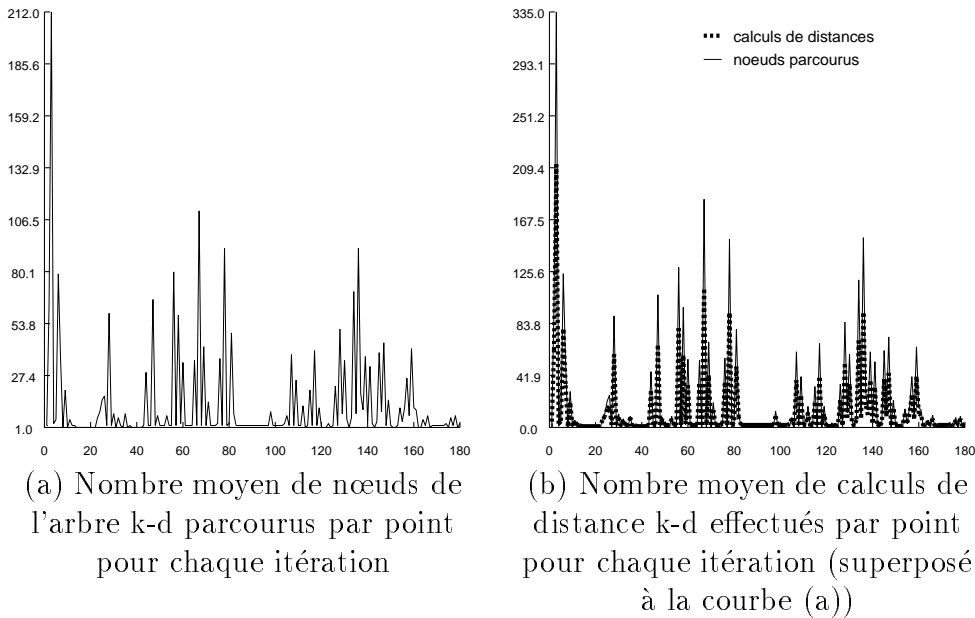


FIG. IV●6.13: Quantification de l'efficacité du calcul de distance itérée dans un arbre k-d.



### 6.2.5 Déformation utilisant un calcul de distance 3D

Nous avons comparé les résultats de la section précédente, obtenus avec la distance 6D avec ceux obtenus avec une distance 3D. Nous avons utilisé la distance octree-spline, pour l'efficacité des calculs de distance en 3D.

On remarque sur la figure IV●6.14 que sur l'apophyse transverse placée à droite dans la coupe transversale, un contour interne du modèle est superposé à un contour externe des données.

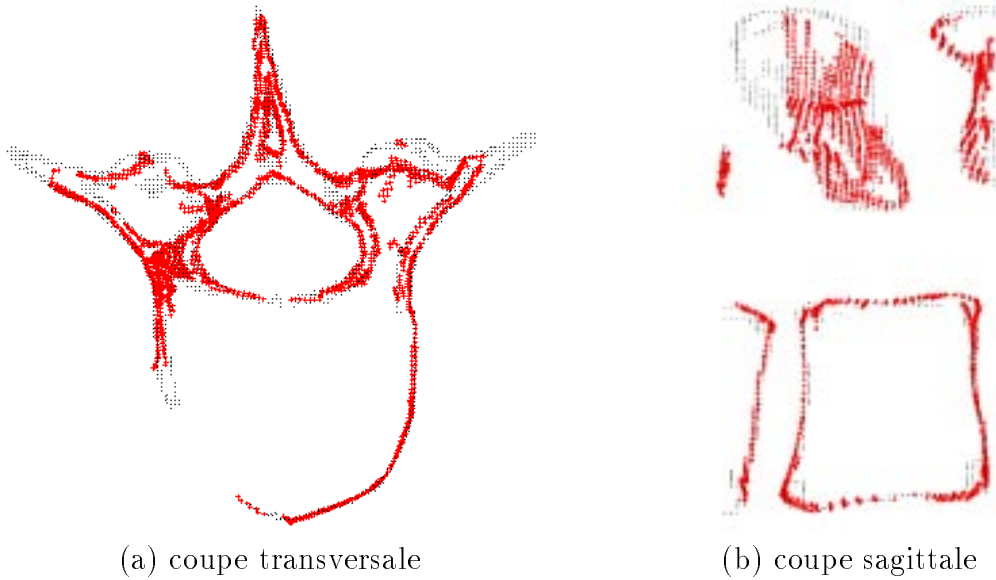


FIG. IV●6.14: Eléments guidant la déformation (distance 3D).

La figure IV●6.15 laisse apparaître une moins bonne superposition des contours que sur IV●6.8.

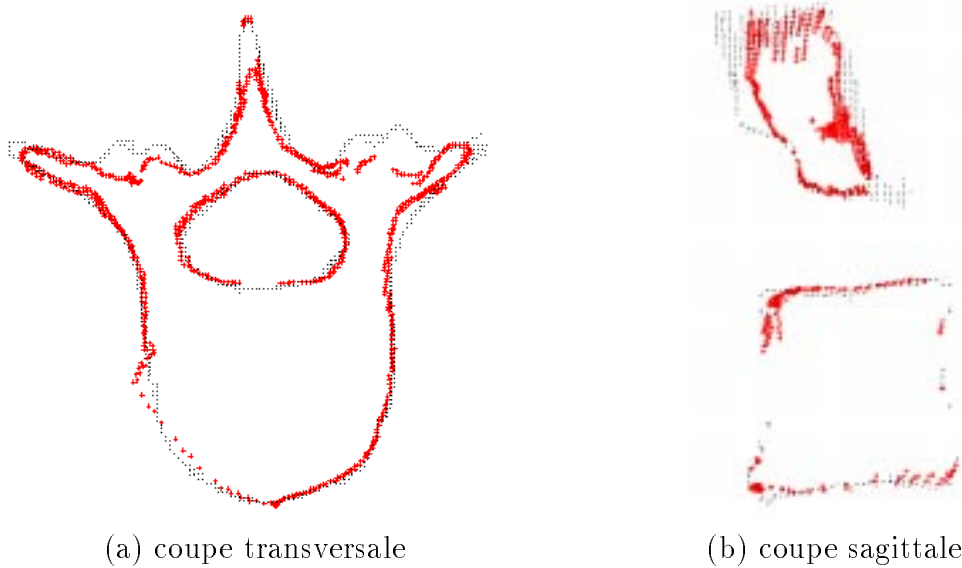


FIG. IV●6.15: Points de surface inférés par la déformation (distance 3D).

## IV•6.3 Echographie virtuelle

Delphine Henry a utilisé notre modèle pour simuler la déformation de veines et d'artères de la cuisse, afin de créer des images échographiques virtuelles [Henry 1997]. A l'aide d'une sonde échographique dont la position est repérée dans l'espace, elle crée une base de données d'images de la cuisse, localisées dans l'espace. Elle génère ensuite par interpolation des images correspondant aux positions d'une sonde virtuelle (figure IV•6.16).

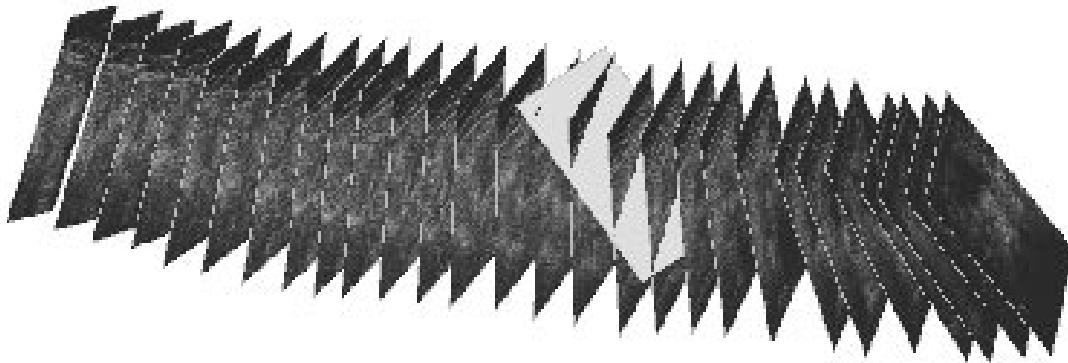
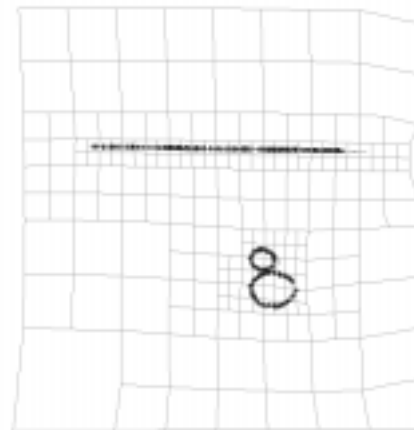


FIG. IV•6.16: Base d'images échographiques localisées dans l'espace. En gris clair la position d'une coupe virtuelle à générer

La déformation des veines et artères selon la pression exercée par la sonde virtuelle sur la cuisse est modélisée pour générer les contours des veines et artères modifiés par la sonde (figure IV•6.17 (a)). C'est alors qu'intervient notre modèle, pour inférer sur toute l'image échographique les déformations induites à partir de ces structures (figure IV•6.17 (b)).



(a) Structures à mettre en correspondance



(b) Après la transformation non-rigide

FIG. IV•6.17: Inférence de déformations à partir de structures de référence

La figure IV•6.18 montre les différentes images correspondant à ce processus. En (a), l'image sans considération de pression, qui est l'image originale. En (b), l'image virtuelle

générée par inférence de déformation, pour une pression de la sonde donnée. En (c), l'image réellement acquise en appliquant cette pression.

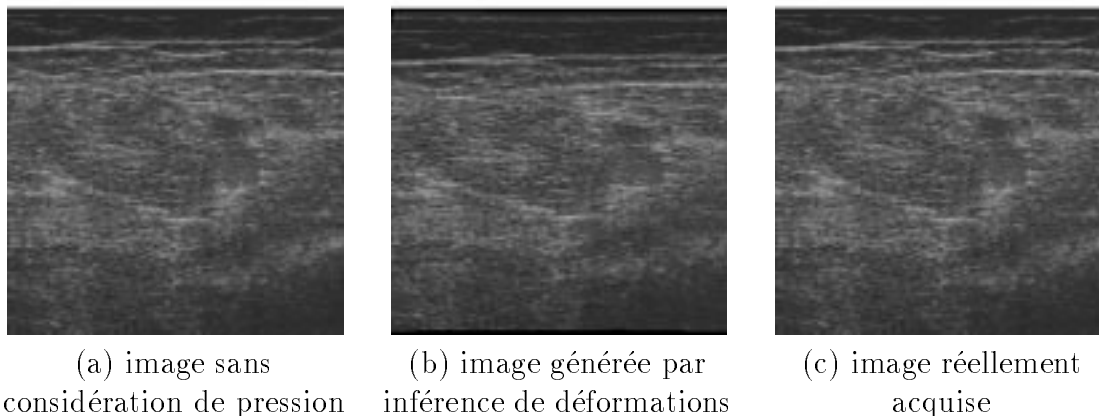
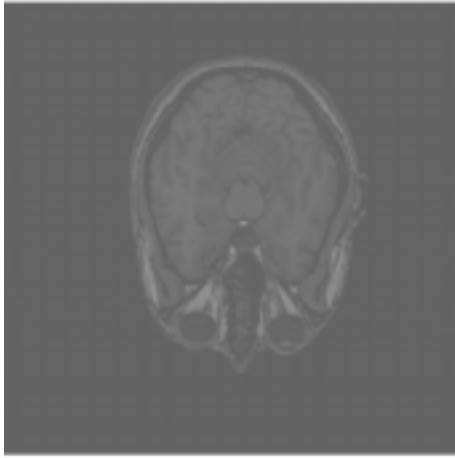


FIG. IV●6.18: Génération d'une image inférant la pression exercée par la sonde sur la patient : comparaison d'une image réelle et d'une image générée par inférence de déformation

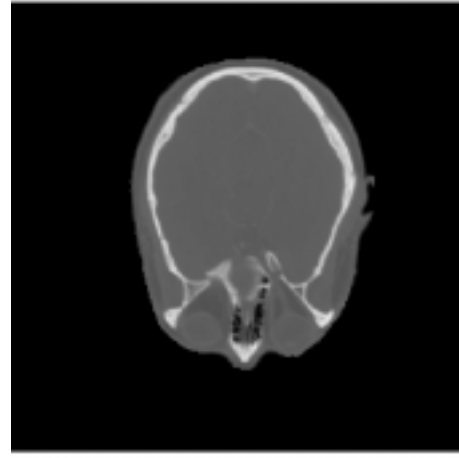
## IV●6.4 Correction des distorsion en IRM

Les résultats de cette section ont été obtenus par Jean Romanet. Ils concernent la correction des distorsions d'une image IMR en utilisant une image TDM comme référence. Un volume TDM et d'un volume IRM du cerveau d'un même patient, enregistrés le même jour, nous ont été fournis par le Dr Richard Bucholz du S<sup>t</sup> Louis Hospital, USA. Ces volumes ont les mêmes tailles de coupe soit  $256 \times 256$  et des voxels cubiques de même taille soit  $1.172 \text{ mm}$  de côté. Ils ont été nettoyés et segmentés au ras du crâne pour favoriser dans un premier temps la convergence du recalage (figure IV●6.19). Puis un filtre de Deriche  $3D$  a été appliqué. Les extrema du gradient ont été extraits, puis seuillés par hystérésis. Etant donné les différences entre les deux modalités, les contours TDM sont inclus dans ceux de l'IRM (figure IV●6.20). Cela permet de ne pas avoir trop de points aberrants lors du recalage.

Le volume TDM est ensuite déformé, pour correspondre au volume IRM. La figure IV●6.21 montre la comparaison des deux nuages de points avant et après recalage. Les figures IV●6.22 et IV●6.23 montre la comparaison entre l'image IRM reformattée après correction et l'image TDM qui sert de référence. Ce sont des coupes respectivement au niveau des images 20 et 30 du volume TDM.

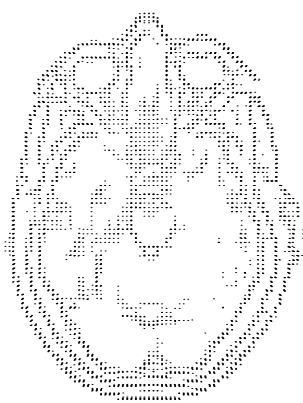


(a) IRM

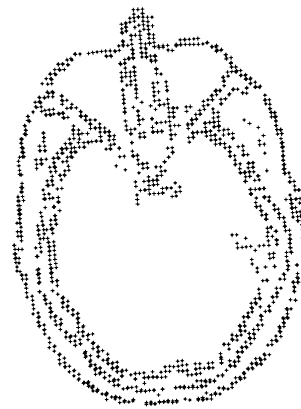


(b) TDM

FIG. IV●6.19: Coupes dans les images originales

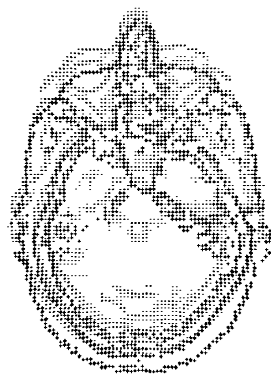


(a) IRM



(b) TDM

FIG. IV●6.20: Les extrema des gradients

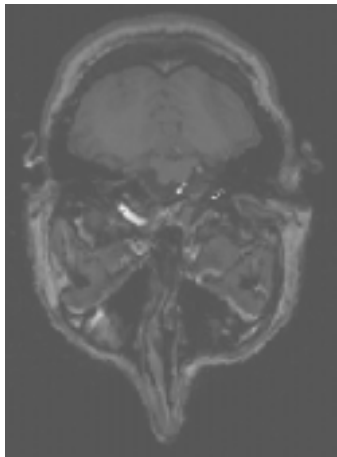


(a) attitude initiale

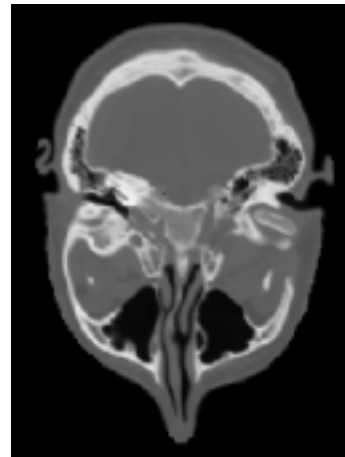


(b) après la déformation

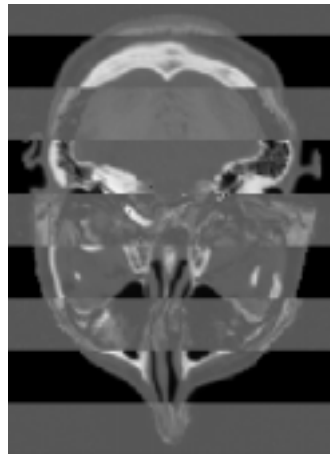
FIG. IV●6.21: Vues des superpositions des deux images



(a) coupe IRM corrigée

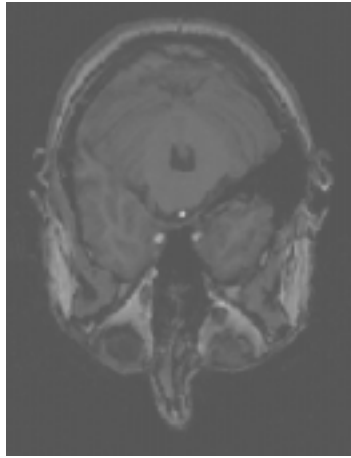


(b) coupe TDM

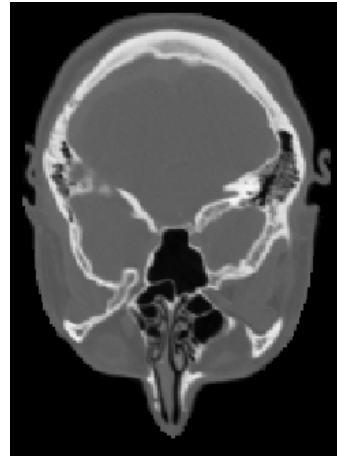


(c) comparaison entre (a) et (b)

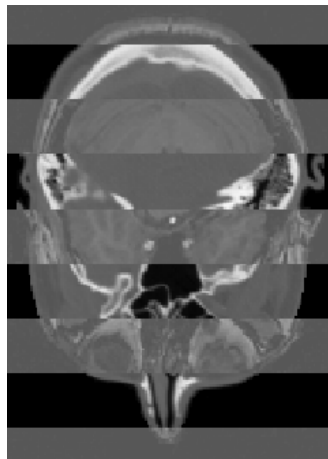
FIG. IV●6.22: Comparaison des images IRM après correction et TDM, au niveau de la  $20^{i(e)}me$  image TDM



(a) coupe IRM corrigée



(b) coupe TDM



(c) comparaison entre (a) et (b)

FIG. IV●6.23: Comparaison des images IRM après correction et TDM, au niveau de la  $30^{i(e)me}$  image TDM

### IV•7.1 Discussion

#### 7.1.1 Validation du modèle

Nous avons effectué les premiers tests de validation de notre modèle, selon une méthodologie très simple de comparaison avec un résultat obtenu par segmentation manuelle. Nous avons montré que la distance 6D permet d'obtenir de meilleurs résultats que la distance 3D, en particulier pour distinguer contours internes et externes d'une vertèbre dans une image TDM.

Le protocole de validation à mettre en œuvre devra porter sur plusieurs jeux d'images réelles. Il sera intéressant de vérifier à partir d'un échantillon suffisamment important d'images les bonnes propriétés de notre méthode. Dans un premier temps, nous pourrions nous contenter d'images TDM de vertèbres lombaires de différents patients sains. L'expérimentation devra ensuite porter sur des vertèbres de patients présentant une pathologie lombaire. Elle s'étendra à d'autres types de vertèbres, dorsales ou cervicales, afin de déterminer quelles sont les limitations inhérentes à l'emploi d'une vertèbre lombaire comme modèle. Nous passerons ensuite à l'étude d'autres modalités d'imagerie, comme l'IRM, et à d'autres organes que la vertèbre.

#### 7.1.2 Perspectives

Ce travail ouvre de nombreuses questions qui méritent des études particulières. Plusieurs éléments du modèle peuvent varier :

- Les coefficients de régularisation qui règlent les contraintes appliquées sur les déformations.
- Le seuil d'élimination des outliers, c'est-à-dire la valeur (absolue ou relative) de la distance au-delà de laquelle un point est considéré comme aberrant.
- La méthode de raffinement de l'octree de déformation en fonction de la distance aux éléments géométriques

La régularisation devra pouvoir être relâchée au cours du temps, afin de permettre l'évolution des éléments qui sont dans les zones où la correspondance n'est pas satisfaisante. Les autres éléments étant déjà proches de leurs correspondants ne risquant plus d'être déplacés.

Au cours de la minimisation, on peut aussi imaginer que les coefficients de régularisation évoluent localement en fonction de la qualité locale de la correspondance.

---

## IV•7.2 Conclusion

Nous avons présenté une méthode de segmentation 3-d consistant à inférer un modèle sur l'image traitée par déformation volumique de ce modèle. Cette approche est a priori plus robuste que les méthodes de surfaces déformables, car la déformation est calculée à partir d'éléments répartis dans un volume et non pas seulement sur une surface. Son implémentation utilise un maillage de déformation adaptatif, hiérarchique et régularisé, ainsi qu'une minimisation de distances généralisées entre des points du modèle munis de caractéristiques différentielles et des images données munis de ces mêmes caractéristiques. Nous avons présenté une méthode efficace pour le calcul de ces distances en utilisant des arbres k-d. Nous avons proposé une nouvelle méthode de recherche du point le plus proche dans un arbre k-d dans le cas d'une recherche itérée. Nous avons validé notre méthode sur un cas d'images médicales réelles, et nous l'avons appliqué à d'autres domaines que la segmentation.





# Conclusion



---

# Conclusion

*“Etre menacé de mort, ça fait naître.”*  
F.Dolto

Le but de cette thèse était d'étudier les modèles déformables et leurs applications en imagerie médicale, en particulier en ce qui concerne la reconstruction d'objets à partir de points sur leur surface et la segmentation d'images volumiques.

## Bilan

Nous avons établi une étude bibliographique des modèles déformables, classifiés selon cinq composantes, puis nous avons décrit trois modèles déformables, l'un surfacique, l'autre implicite, le dernier volumique. Notre étude nous a permis de mettre en lumière les avantages et les inconvénients des trois méthodes.

Le modèle surfacique est souple d'usage, sa convergence est très rapide, et il peut représenter les données avec un niveau de détail réglable. Il ne contient en revanche pas de possibilité de coder une forme particulière.

Le modèle implicite est très compact, et il permet de représenter des objets fermés de topologie complexe. Il fournit une représentation lisse de la surface de l'objet, et il donne également la possibilité de connaître très aisément la position de tout point par rapport à l'objet : intérieur, extérieur ou surface.

Nous avons appliqué ces modèles au problème de reconstruction de la surface d'un objet à partir de points non organisés de cette surface. Pour cette application, nous avons montré que le modèle des  $\delta$ -snakes permet d'obtenir rapidement une bonne reconstruction de la surface, si les points de données ne comportent pas de lacunes trop importantes. De plus, ce modèle se prête bien à une manipulation interactive, soit par le placement d'objets intraversables, soit par un déplacement direct des points du maillage. Nous avons ensuite développé notre modèle implicite, dans l'idée d'obtenir une modélisation d'un objet qui tienne compte de son volume et non plus seulement de sa surface, et d'utiliser cette modélisation pour animer de manière physiquement réaliste des organes, dans le but par exemple de simuler une intervention chirurgicale. Nous avons validé l'utilisation du modèle implicite déformable pour la reconstruction de surfaces.

Nous avons ensuite choisi d'appliquer les modèles déformables au problème de la seg-

mentation d'images médicales volumiques. Pour cela, nous nous sommes penchés sur la question des caractéristiques de bas niveau, et nous avons choisi d'extraire par un opérateur 3D les gradients de la fonction de niveau de gris des images. Nous avons retenu un modèle volumique qui permet d'utiliser non seulement l'information de la surface extérieure de l'objet, mais aussi toute l'information présente dans le volume. En ce qui concerne les vertèbres, par exemple, il s'agit en plus des contours ou des surfaces internes à la vertèbre (comme l'interface entre l'os spongieux et l'os cortical), et aussi des éléments des vertèbres voisines.

La déformation est soumise à un ensemble de contraintes (sous la forme d'une énergie de régularisation), pour lui garantir de bonnes propriétés. Ainsi cette déformation volumique permet d'inférer la localisation d'organes voisins qui n'apparaissent pas dans une des images. Ce modèle déformable volumique est applicable à n'importe quel type d'objet, surfacique ou non, partiel ou complet, quelle que soit sa topologie.

Nous avons obtenu des résultats dans les domaines que nous nous étions proposé d'étudier, mais ce n'est pas tout, le modèle volumique déformable a également d'autres applications, nous l'avons en particulier utilisé dans un projet visant à corriger les distorsions de l'IRM en utilisant une référence TDM.

## Perspectives

Nous avons fait le tour des différents aspects des modèles déformables, illustrés par trois modèles différents, ce qui a permis de mettre en évidence les avantages de chaque approche, mais qui laisse certaines combinaisons non encore explorées. L'interactivité dont nous avons validé l'importance sur le modèle des  $\delta$ -snakes pourrait être ajoutée aux deux autres modèles. Les caractéristiques de bas niveau et la régularisation du modèle déformable pourraient être appliquées au modèle des  $\delta$ -snakes.

Notre travail, par la décomposition des modèles déformables proposée, ouvre un espace de réflexion pour de futurs travaux. La classification des modèles déformables permet des recombinaisons inédites de composantes. Un modèle composé de plusieurs morceaux de surfaces pourrait être associé à une déformation de forme libre étendue (EFFD). Il serait également intéressant d'adjoindre à un modèle implicite généré par des primitives des déformations locales afin d'obtenir plus de précision dans la reconstruction. Un octree-spline de déformations pourrait être combiné avec des déformations modales statistiques dans un modèle qui aurait des profils de niveaux de gris ou des courbures comme caractéristiques de bas niveau.

---

# Bibliographie

- Aho (A.), Hopcroft (J.) et Ullman (J.). *Structures de données et algorithmes*. InterEditions, 1994.
- Attali (D.), Bertolino (P.) et Montanvert (A.). Using polyballs to approximate shape and skeletons. *12<sup>th</sup> ICPR*, pages 626–628, Octobre 1994.
- Aurdal (L.), Descombes (X.), Maître (H.), Bloch (I.), Adamsbaum (C.) et Kalifa (G.). Fully automated analysis of adrenoleukodystrophy from dual echo mr images : Automatic segmentation and quantification. *Computer Assisted Radiology CAR'95*, pages 35–40, Berlin, Germany, Juin 1995.
- AVS. *User's Guide*. Advanced Visual Systems Inc., 1994.
- Bainville (E.). Reconstruction d'objets tridimensionnels à partir de silhouettes. Master's thesis, Ecole Normale Supérieure de Lyon, France, Juillet 1992.
- Bainville (E.). *Modélisation géométrique et dynamique d'un geste chirurgical*. PhD thesis, Université Joseph Fourier, Grenoble, France, Mars 1996.
- Bajcsy (R.) et Kovacic (S.). Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 46, pages 1–21, 1989.
- Bardinet (E.). *Modèles déformables contraints : applications à l'imagerie cardiaque*. PhD thesis, Université Paris IX, Dauphine, France, Dcembre 1995.
- Bardinet (E.), Cohen (L.) et Ayache (N.). Fitting of iso-surfaces using superquadrics and free-form deformations. *IEEE WWorkshop on Biomedical Image Analysis*, pages 184–193, Seattle, Juin 1994. IEEE Computer Society.
- Benayoun (S.). *Calcul local du mouvement. Application à l'imagerie médicale multidimensionnelle*. PhD thesis, Université Paris 9 Dauphine, Dcembre 1994.
- Bezdek (J.), Hall (L.) et Clarke (L.). Review of mr image segmentation techniques using pattern recognition. *Medical Physics*, 20 (4), pages 1033–1048, Juillet 1993.
- Bittar (E.), Lavallée (S.) et Szeliski (R.). A method for registering overlapping range images of arbitrarily shaped surfaces for 3-D object reconstruction. *SPIE Vol. 2059, Sensor Fusion VI*, Boston, Septembre 1993. Society of Photo-Optical Instrumentation Engineers.
- Bittar (E.), Tsingos (N.) et Gascuel (M-P.). Automatic Reconstruction of Unstructured 3D Data : Combining a Medial Axis and Implicit Surfaces. *Computer Graphics forum (Eurographics'95)*, 14 (3), pages 457–468, Aot 1995.
- Blanc (C.) et Schlick (C.). Extended Field Functions for Soft Objects. Gascuel (Wyvill &), editor, *Implicit Surface'95, Eurographics Workshop*, pages 21–32, Grenoble, France, Avril 1995.

- Blinn (J.). A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, pages 235–256, Juillet 1982.
- Bloch (I.). Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account. *Pattern Recognition Letters*, 17 (8), pages 905–919, 1996.
- Bloomenthal (Jules), editor. *Introduction to implicit surfaces*. Morgan Kaufman, 1997.
- Bloomenthal (Jules) et Shoemake (Ken). Convolution surfaces. *Computer Graphics*, 25 (4), pages 251–256, Juillet 1991. Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 1991).
- Bloomenthal (Jules) et Wyvill (Brian). Interactive techniques for implicit modeling. *Computer Graphics*, 24 (2), pages 109–116, Mars 1990.
- Boissonnat (J.-D.). Geometric structures for three-dimensional shape representation. *ACM Trans. Graph.*, 3 (4), pages 266–286, 1984.
- Boissonnat (J.-D.) et Geiger (B.). Three-dimensional reconstruction of complex shapes based on the Delaunay triangulation. Report 1697, INRIA Sophia-Antipolis, Valbonne, France, 1992.
- Bookstein (F. L.). Principal warps : Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 (6), pages 567–585, Juin 1989.
- Bookstein (F.L.) et Green (W.D.K.). Edge information at landmarks in medical images. *SPIE vol. 1808 Visualization in Biomedical Computing*, pages 242–258, 1992.
- Borgefors (G.). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34, pages 344–371, 1986.
- Boujemaa (N.), Stamon (G.) et Gagalowicz (A.). Modélisation floue pour la segmentation d'images. *Actes Congrès AFCET RFIA 94*, pages 163–173, Paris, 1994.
- Bro-Nielsen (M.), Gramkow (C.) et Kreiborg (S.). Non-rigid Image Registration Using Bone Growth Model. Troccaz (J.), Grimson (E.) et Mosges (R.), editors, *CVRMed-MRCAS'97 Proceedings, LNCS Series 1205*, pages 3–12, Berlin, Mars 1997. Springer-Verlag.
- Bro-Nielsen (M.) et Gramkow (K.). Fast Fluid Registration of Medical Images. Springer, editor, *Visualization in Biomedical Computing*, pages 267–276, Septembre 1996.
- Brunie (L.). *Fusion d'images médicales multi-modales : application à l'étude tridimensionnelle dynamique de la colonne vertébrale (in french)*. PhD thesis, Grenoble University, Dcembre 1992.
- Burdea (G.) et Coiffet (P.). *La Réalité Virtuelle*. Hermes, Dcembre 1993.
- Cani-Gascuel (M-P.) et Desbrun (M.). Animation of Deformable Models Using Implicit Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 1 (3), Mars 1997.
- Canny (John). A computational approach for edge detection. *IEEE/PAMI*, 8 (6), pages 679–697, Novembre 1986.
- Chadwick (J.E.), Haumann (D.R.) et Parent (R.E.). Layered Construction for Deformable Characters. *Computer Graphics (SIGGRAPH'89)*, pages 243–252, Juillet 1989.
- Champleboux (G.). *Utilisation des fonctions splines pour la mise au point d'un capteur tridimensionnel sans contact : quelques applications médicales (in french)*. PhD thesis, Grenoble University, Juillet 1991.
- Christensen (G.E.), Joshi (S.C.) et Miller (M.I.). Individualizing Anatomical Atlases of the Head. Springer, editor, *Visualization in Biomedical Computing*, pages 343–348, Septembre 1996.

- Cinquin (P.). *Application des fonctions splines au traitement d'images numériques*, thèse d'état. PhD thesis, Grenoble University, 1987.
- Cohen (I.), Ayache (N.) et Cohen (L.). Segmenting, visualizing and characterising 3D anatomical structures with deformable surfaces. *IEEE EMBS Conference*, pages 1183–1184, Orlando, Florida, Novembre 1991.
- Cohen (L.D.) et Cohen (I.). Deformable Models for 3D Medical Images using Finite Element and Balloons. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92)*, pages 592–598, Juin 1992.
- Cohen (L.D.) et Cohen (I.). Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15 (11), pages 1131–1147, Novembre 1993.
- Collins (D.L.), Le Goualher (G.), Venugopal (R.), Caramanos (A.), Evans (A.C.) et Barillot (C.). Cortical Constraints for Non-Linear Cortical Registration. Hoehne (K.) et Kikinis (R.), editors, *Visualization and Biomedical Computing*, pages 307–316. Springer-Verlag, Septembre 1996.
- Cootes (T. F.), Taylor (C. J.), Cooper (D. H.) et Graham (J.). Active Shape Models - Their Training and Application. *Computer Vision and Image Understanding*, 61 (1), pages 38–59, 1995.
- Cootes (T.F.), Hill (A.), Taylor (C.J.) et Haslam (J.). The Use of Active Shape Models For Locating Structures in Medical Images. *Image and Vision Computing*, volume 12 (6), pages 355–365, 1994.
- Coquillard (S.). Extended Free-Form Deformation : A Sculpting Tool for 3D Geometric Modeling. *Computer Graphics (SIGGRAPH'90)*, pages 187–196, Aot 1990.
- Cotin (S.), Delingette (H.) et Ayache (N.). Real time volumetric deformable models for surgery simulation. Hoehne (K.) et Kikinis (R.), editors, *Visualization and Biomedical Computing, VBC'96*, pages 535–540. Springer-Verlag, 1996.
- Davis (M.H.), Khotanzad (A.), Flaming (D.P.) et Harms (S.E.). A Physics-Based Coordinate Transformation for 3-D Image Matching. *IEEE Transactions on Medical Imaging*, 16 (3), pages 317–328, Juin 1997.
- DeCarlo (D.) et Metaxas (D.). Blended Deformable Models. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 566–572, Seattle, Juin 1994.
- Deering (M.). HoloSketch : A Virtual Reality Sketching/Animation Tool. *ACM Transactions on Computer-Human Interaction*, 2 (3), pages 220–238, 1995.
- Delingette (H.). *Modélisation et reconnaissance d'objets tridimensionnels à l'aide de maillages simples*. PhD thesis, Ecole Centrale Paris, Juillet 1994.
- Delingette (H.). Decimation of isosurfaces with Deformable Models. Troccaz (J.), Grimson (E.) et Mosges (R.), editors, *CVRMed-MRCAS'97 Proceedings, LNCS Series 1205*, pages 83–92, Berlin, Mars 1997. Springer-Verlag.
- Dellepiane (S.), Venturi (G.) et Vernazza (G.). Model generation and model matching of real images by a fuzzy approach. *Pattern REcognition*, 25 (2), pages 115–137, 1992.
- Deriche (Rachid). Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector. *International Journal of Computer Vision*, pages 167–187, 1987.
- Deriche (Rachid). Fast Algorithms for Low-Level Vision. *IEEE/PAMI*, 12 (1), pages 78–87, Janvier 1990.
- Edelsbrunner (Herbert) et Mucke (Ernst P.). Three-dimensional alpha shapes. *1992 Workshop on Volume Visualization*, pages 75–82, 1992.



- Evans (A.C.), Collins (D.L.), Neelin (P.) et Marrett (T.S.). Correlative analysis of three-dimensional brain images. Taylor (R.), Lavallée (S.), Burdea (G.) et Mosges (R.), editors, *Computer-integrated surgery : technology and clinical applications*, pages 99–114. MIT Press, Cambridge, MA, 1996.
- Evans (A.C.), Marrett (S.), Collins (D.L.) et Peters (T.M.). Anatomical-functional correlative analysis of the human brain using three-dimensional imaging systems. *Proceedings SPIE : Medical Imaging III*, pages 264–274, 1989.
- Feldmar (Jacques.). *Recalage rigide, non rigide et projectif d'images médicales tridimensionnelles*. PhD thesis, Ecole Polytechnique, Novembre 1995.
- Ferley (E.), Cani-Gascuel (M-P.) et Attali (D.). Skeletal reconstruction of Branching Shapes. *Computer Graphics Forum (to appear)*, 1997.
- Fidrich (M.) et Thirion (J.P.). Multiscale Extraction and Representation of Features from Medical Images. Technical Report 2365, INRIA, Octobre 1994.
- Fischler (M.) et Elschlager (R.). The representation and matching of pictorial structures. *IEEE Trans. on Computers*, 22 (1), pages 67–92, 1973.
- Forsey (D. R.) et Bartels (R. H.). Hierarchical B-spline refinement. *Computer Graphics (SIGGRAPH'88)*, 22 (4), pages 205–212, Aot 1988.
- Friedman (J. H.), Bentley (J.L.) et Finkel (R. A.). An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Software*, 3 (3), pages 209–226, Septembre 1977.
- Fukunaga (K.) et Narendra (P.M.). A Branch and Bound Algorithm for Computing k-Nearest Neighbors. *IEEE Transactions on Computers*, pages 750–753, Juillet 1975.
- Gascuel (J-D.). *Fabule : un environnement de recherche pour l'animation et la simulation*. *Les Simulateurs, Troisième Séminaire du groupe de travail français Animation et Simulation*, Octobre 1994.
- Gascuel (Marie-Paule). An implicit formulation for precise contact modeling between flexible solids. *Computer Graphics*, pages 313–320, Aot 1993. Proceedings of SIGGRAPH'93.
- Gascuel (M.P.). *Déformations de surfaces complexes : techniques de haut niveau pour la modélisation et l'animation*. PhD thesis, Université Paris XI, Octobre 1990.
- Géraud (T.), Mangin (J.-F.), Bloch (I.) et Maître (H.). Segmenting internal structures in 3d mr images of the brain by markovian relaxation on a watershed based adjacency graph. *ICIP-95*, pages 548–551, Washington DC, Octobre 1995.
- Hamadeh (Ali). *Une Approche Unifiée pour la Segmentation et la Mise en Correspondance 3D/2D d'Images Multimodales*. PhD thesis, Institut National Polytechnique de Grenoble I.N.P.G, Mars 1997.
- Henry (D.). *Echographie 3D : Outils pour la modélisation de structures et la simulation d'ex-amens échographiques*. PhD thesis, Université Joseph Fourier, Grenoble, France, Octobre 1997.
- Hoppe (H.), DeRose (T.), Duchamp (T.) et Halstead (M.). Piecewise Smooth Surface Reconstruction. *Computer Graphics (SIGGRAPH'94)*, pages 295–302, Juillet 1994.
- Hoppe (H.), DeRose (T.), Duchamp (T.) and McDonald J.) et W. (Stuetzle). Surface reconstruction from unorganized points. E. (Catmull E.), editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, pages 71–78, Juillet 1992.
- Horand (R.) et Monga (O.). *Vision par Ordinateur*. Hermes, Paris, 1993.
- Hsu (W.M.), Hughes (J.F.) et Kaufman (H.). Direct Manipulation of Free-Form Deformations. *Computer Graphics (SIGGRAPH '92 Proceedings)*, pages 177–184, Juillet 1992.

- Huber (P. J.). *Robust Statistics*. John Wiley & Sons, New York, New York, 1981.
- Jacq (J.J.) et Roux (C.). Registration of non-segmented images using a genetic algorithm. *CVRMED (Computer Vision, Virtual Reality, Robotics in Medicine) Proc.*, pages 205–211. Springer, 1995.
- Jimenez (S.). *Modélisation et simulation physique d'objets volumiques déformables complexes*. PhD thesis, Institut National Polytechnique de Grenoble, Novembre 1993.
- Jolion (J.-M.) et Rosenfeld (A.). *A Pyramid Framework for Early Vision*. Kluwer Academic Publishers, 1991.
- Joukhadar (A.). *Simulation dynamique et applications robotiques*. PhD thesis, Institut National Polytechnique de Grenoble, Mai 1997.
- Kacic-Alesic (Zoran) et Wyvill (Brian). Controlled blending of procedural implicit surfaces. *Graphics Interface'91*, pages 236–245, Calgary, Canada, Juin 1991.
- Kass (M.), Witkin (A.) et Terzopoulos (D.). Snakes : Active contour models. *International Journal of Computer Vision*, 1, pages 321–331, 1988.
- Kim (B.), Boes (J.L.), Frey (K.A.) et C.R. (Meyer). Mutual Information for Automated Multimodal Image Warping. Springer, editor, *Visualization in Biomedical Computing*, pages 349–354, Septembre 1996.
- Kim (B.S.) et Park (S.B.). A Fast k Nearest Neighbor Finding Algorithm Based on the Ordered Partition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8 (6), pages 761–766, Novembre 1986.
- Lachaud (J.O.) et Bainville (E.). A discrete adaptative model following topological modifications of volumes. *Discrete Geometry for Computer Imagery, Grenoble*, Septembre 1994.
- Lavallée (S.), Cinquin (P.) et Troccaz (J.). Computer Integrated Surgery and Therapy : State of the Art. Roux (C.) et Coatrieux (J.L.), editors, *Contemporary Perspectives in Three-Dimensional Biomedical Imaging*, chapter 10, pages 239–310. IOS Press, Amsterdam, NL, 1997.
- Lavallée (S.), Szeliski (R.) et Brunie (L.). Matching 3-D smooth surfaces with their 2-D projections using 3-D distance maps. *SPIE Vol. 1570 Geometric Methods in Computer Vision*, pages 322–336, San Diego, CA, Juillet 1991.
- Lavallée (S.), Szeliski (R.) et Brunie (L.). Anatomy-based registration of 3-D medical images, range images, X-ray projections, 3-D models using Octree-Splines. Taylor (R.), Lavallée (S.), Burdea (G.) et Mosges (R.), editors, *Computer Integrated Surgery*, pages 115–143. MIT Press, Cambridge, MA, 1996.
- Lefaix (G.), Riot (X.), Haignon (P.), Collorec (R.) et Ramee (A.). 3D modeling and deformation analysis of the vertebra with spherical harmonics. *19th Annual international conference of the IEEE engineering in medicine and biology society*, pages 422–425, CHICAGO, Octobre 1997.
- Leitner (F.). *Segmentation dynamique d'images tridimensionnelles*. PhD thesis, INPG, 1993.
- Lockwood (P.). A low-cost dtw-based discrete utterance recognizer. *EICPR-86*, pages 467–469, 1986.
- Lombardo (J.C.) et Puech (C.). Oriented particles : a tool for shape memory objects modelling. *Graphics Interface'95*, pages 255–262, 1995.
- Loop (C.). Smooth Spline Surfaces over Irregular Meshes. *Computer Graphics (SIGGRAPH'94)*, pages 303–310, Juillet 1994.
- Lubiarz (S.) et Lockwood (P.). Evaluation of fast algorithms for finding the nearest neighbor. *ICASSP97*, volume 2, pages 1491–1494, Munich, Avril 1997.

- Luciani (A.) et Cadoz (C.). Utilisation de modèles mécaniques et géométriques pour la synthèse d'images animées. *Deuxième colloque Image. CESTA*, Nice, Avril 1986.
- Magnin (I.E.) et Reissman (P.J.). Modélisation et mise en correspondance avec la pyramide neuractive. *Actes 10ème Congrès AFCET - Reconnaissance des Formes et Intelligence Artificielle (RFIA '96)*, Rennes, 1996.
- Maintz (J.B.A.), Elsen (P.A.) van den et Viergever (M.A.). Comparison of edge-based and ridge-based registration of CT and MRI brain images. *Medical Image Analysis*, 1 (2), pages 151–161, 1996.
- Malladi (R.), Sethian (J.A.) et Vemuri (B.C.). Shape Modeling with Front Propagation : A Level Set Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17 (2), pages 158–175, Fvrier 1995.
- Mangin (J.F.). *Mise en correspondance d'images médicales 3D multi-modalités multi-individus pour la corrélation anatomo-fonctionnelle cérébrale*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Mars 1995.
- McInerney (T.) et Terzopoulos (D.). A finite element model for 3D shape reconstruction and nonrigid motion tracking. *Fourth International Conference on Computer Vision (ICCV'93)*, pages 518–523, Berlin, Mai 1993. IEEE Computer Society Press.
- McInerney (T.) et Terzopoulos (D.). Deformable models in medical image analysis : a survey. *Medical Image Analysis*, 2 (2), pages 91–108, 1996.
- Miller (J. V.), Breen (D. E.), E. (Lorensen W.), M. (O'Bara R.) et J. (Wozny M.). Geometrically deformed models : A method for extracting closed geometric models from volume data. W. (Sederberg T.), editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, pages 217–226, Juillet 1991.
- Monga (O.) et Benayoun (S.). Using partial derivatives of 3D images to extract typical surface features. *computer vision and image understanding*, 61 (2), pages 171–189, Mars 1995.
- Monga (O.), Deriche (R.) et Rocchisani (J-M.). 3d edge detection using recursive filtering : Application to scanner images. *Computer Vision Graphics and Image processing*, volume 53 (1), pages 76–87, 1991.
- Muraki (Shigeru). Volumetric shape description of range data using blobby model. *Computer Graphics*, 25 (4), pages 227–235, Juillet 1991.
- Nastar (C.) et Ayache (N.). Classification of Nonrigid Motion in 3D Images using Physics-Based Vibration Analysis. *IEEE Workshop on Biomedical Image Analysis*, pages 61–69, Seattle, Juin 1994.
- Nielson (G.A.). Scattered data modeling. *IEEE Computer Graphics Applications*, pages 255–262, 1993a.
- Nielson (G.A.). Scattered data modeling. *IEEE Transactions on Medical Imaging*, 13, pages 60–70, Janvier 1993b.
- Nishimura (H.), Hirai (M.), Kawai (T.), Kawata (T.), Shirakawa (I.) et Omura (K.). Objects modeling by distribution function and a method of image generation (in japanese). *Trans. IEICE Japan*, J68-D (4), pages 718–725, 1985.
- Opalach (A.) et Cani-Gascuel (M-P.). Local Deformation for Animation of Implicit Surfaces. *SCCG'97*, Bratislava, Slovakia, Juin 1997.
- Parker (A.A.), Kenyon (R.V.) et Troxel (D.E.). Comparison of Interpolating Methods for Image Resampling. *IEEE Transactions on Medical Imaging*, 2 (1), pages 31–39, Mars 1996.
- Pentland (A.) et Sclaroff (S.). Closed-Form Solutions for Physically Based Shape Modeling and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 (7), pages 715–729, Juillet 1991.

- Pentland (A.) et Williams (J.). Good Vibrations : Modal Dynamics for Graphics and Animation. *Computer Graphics (SIGGRAPH'89)*, 23 (3), pages 215–222, Juillet 1989.
- Pfaltz (J.L.) et Rosenfeld (A.). Computer representation of planar regions by their skeletons. *Comm. of ACM*, 10, pages 119–125, Fvrier 1967.
- Preparata (F.) et Shamos (M.). *Computational Geometry, An Introduction*. Springer Verlag, 1986.
- Press (W. H.), Flannery (B. P.), Teukolsky (S. A.) et Vetterling (W. T.). *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, second edition, 1992.
- Promayon (E.). *Modélisation et simulation de la respiration*. PhD thesis, Université Joseph Fourier, Grenoble, France, Novembre 1997.
- Promayon (E.), Baconnier (P.) et Puech (C.). Physically-Based Deformations Constrained in Displacements and Volume. *Computer Graphics Forum (EuroGraphics'96)*, volume 15(3), pages 155–164, Aot 1996.
- Promayon (E.), Baconnier (P.) et Puech (C.). Physically-Based Model for Simulating the Human Trunk Respiration Movements. J. Troccaz (R. Mosges), E.Grimson, editor, *CVRMed-MRCAS'97*, LNCS 1205, pages 379,388. Springer, 1997.
- Ramasubramanian (V.) et Paliwal (K.). Fast K-Dimensional Tree Algorithms for Nearest Neighbor Search with Application to Vector Quantization Encoding. *IEEE Transactions on Signal Processing*, 40 (3), pages 518–531, Mars 1992.
- Raya (S.P.) et Udupa (J.K.). Shape based interpolation of Multidimensional objects. *IEEE transactions on medical imaging*, 9 (1), pages 32–42, Mars 1990.
- Robert (A.), Schmitt (F.) et Mousseaux (E.). Modelisation of the left ventricle and its deformations using superquadrics and hyperquadrics. Lemke (H.U. et al.), editor, *CAR'95 (Computer Assisted Radiology)*. Springer, Juin 1995.
- Rohr (K.), H.S. (Stiehl), Sprengler (R.), Beil (W.), Buzug (T.M.), Weese (J.) et Kuhn (M.H.). Point-Based Elastic Registration of Medical Image Data Using Approximating Thin-Plate Splines. Springer, editor, *Visualization in Biomedical Computing*, pages 297–306, Septembre 1996.
- Romanet (J.). Mise en correspondance élastique d'images du cerveau : IRM-scannerX. Master's thesis, Université Joseph Fourier, Grenoble, D.E.A. Modèles et Instruments en Médecine et Biologie, Septembre 1997.
- Rosenfeld (A.) et J.L. (Pfaltz). Sequential operations in digital picture processing. *Journal of the ACM*, 13 (4), pages 471–494, Octobre 1966.
- Rouet (J-M.), Jacq (J-J.) et Roux (C.). Recalage élastique 3-D de surfaces numériques par optimisation génétique. *Colloque sur le Traitement du Signal et des Images (GRETSI'97)*, Grenoble University, pages 1395–1398, Septembre 1997.
- Scaroff (s.) et Pentland (A.). On Modal Modeling for Medical Images : Underconstrained Shape Description and Data Compression. *IEEE Workshop on Biomedical Image Analysis*, pages 70–79, Seattle, Juin 1994. IEEE Computer Society.
- Scaroff S. (Pentland A.). Generalized implicit functions for computer graphics. W. (Sederberg T.), editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, pages 247–250, Juillet 1991.
- Sederberg (T.W.) et Parry (S.R.). Free-form deformations of solid geometric models. *Computer Graphics (SIGGRAPH'86)*, 20 (4), pages 151–160, 1986.

- Staib (L.H.) et Duncan (J.S.). Model-Based Deformable Surface Finding for Medical Images. *IEEE Transactions on Medical Imaging*, 15 (5), pages 720–731, Octobre 1996.
- Studholme (C.), Little (J.), Penny (G.), Hill (D.) et Hawkes (D.). Automated multimodality registration using the full affine transformation : application to MR and CT guided skull base surgery. Hoehne (K.) et Kikinis (R.), editors, *Visualization and Biomedical Computing (VBC'96)*, pages 601–606. Springer-Verlag, 1996.
- Sumanaweera (T.S.), Glover (G. H.), Binford (T. O.) et Adler (J. R.). MR susceptibility misregistration correction. *IEEE trans. med. imaging*, 12 (2), pages 251–259, Juin 1993.
- Szeliski (R.). *Bayesian Modeling of Uncertainty in Low-Level Vision*. Kluwer Academic Publishers, Boston, Massachusetts, 1989a.
- Szeliski (R.). *Bayesian modeling of uncertainty in low-level vision*. Kluwer Academic Publishers, Boston, MA, 1989b.
- Szeliski (R.). Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 (6), pages 513–528, Juin 1990.
- Szeliski (R.) et Lavallée (S.). Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. *IEEE Workshop on Biomedical Image Analysis*, Seattle, Juin 1994. IEEE Computer Society.
- Szeliski (R.) et Lavallée (S.). Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. *Int. J. of Computer Vision (IJCV)*, (18) (2), pages 171–186, Mai 1996.
- Szeliski (R.) et Tonnesen (D.). Surface Modeling with Oriented Particle Systems. *Computer Graphics (SIGGRAPH '92 Proceedings)*, pages 185–194, Juillet 1992.
- Terzopoulos (D.) et Metaxas (D.). Dynamic 3D models with local and global deformations : Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 (7), pages 703–714, Juillet 1991.
- Terzopoulos (D.), Witkin (A.) et Kass (M.). Constraints on deformable models : Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36, pages 91–123, 1988.
- Thiel (E.). *Les Distances de Chanfrein en Analyse d'Images : Fondements et Applications*. PhD thesis, Université Joseph Fourier-GRENOBLE I, Septembre 1994.
- Thirion (J.P.). Extremal points : definition and application to 3D image registration. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 587–592, Seattle, Juin 1994.
- Thirion (J.P.), Monga (O.), Benayoun (S.), Guezic (A.) et Ayache (N.). Automatic registration of 3D images using surface curvature. *IEEE Int. Symp. on Optical Applied Science and Engineering*, San Diego, Juillet 1992.
- Tsingos (N.), Bittar (E.) et Gascuel (M-P.). Semi-automatic Reconstruction of Implicit Surfaces for Medical Applications. Earnshah (R.) et Vince (J.), editors, *Computer Graphics International*, pages 3–15. Academic Press, Juin 1995.
- Tsingos (Nicolas) et Gascuel (Marie-Paule). Un modeleur interactif d'objets définis par des surfaces implicites. *Secondes Journées de l'AFIG*, Toulouse, Dcembre 1994.
- Turner (R.). LEMAN : A System for Constructing and Animating Layered Elastic Characters. Earnshah (R.) et Vince (J.), editors, *Computer Graphics International*, pages 185–203. Academic Press, Juin 1995.
- Turner (R.), Gobbetti (E.) et Soboroff (I.). Head-Trackted Stereo Viewing with Two-Handed 3D Interaction for Animated Character Construction. *Computer Graphics forum (Eurographics'96)*, 15 (3), pages 197–206, 1996.

- Elsen (P.A.) van den, Maintz (J.B.A.), Pol (E.J.D.) et Viergever (M.A.). Automatic registration of CT and MR brain images using correlation of geometrical features. *IEEE Trans. Med. Imaging*, 14 (2), pages 384–396, 1995.
- Vemuri (B. C.), Guo (Y.), LAi (S.H.) et Leonard (C.M.). Fast Algorithms for fitting Multiresolution Hybrid Shape Models to Brain MRI. Springer, editor, *Visualization in Biomedical Computing*, pages 213–222, Septembre 1996.
- Vemuri (B.C.) et Radisavljevic (A.). Multiresolution stochastic hybrid shape models with fractal priors. *ACM Trans. on Graphics*, 13 (2), pages 177–207, Avril 1994.
- Viola (P.A.) et Wells (W.M.). Alignment by maximization of mutual information. *Proc of 5th Int. Conf. Computer Vision (ICCV)*, 1995.
- Wells (W.M.), Viola (P.), Atsumi (H.), Nakajima (S.) et R. (Kikinis). Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1 (1), pages 35–52, Mars 1996.
- Wells (W.M.), Viola (P.) et Kikinis (R.). Multi-modal volume registration by maximization of mutual information. *MRCAS95, Medical Robotics and Computer Assisted Surgery Proc.*, pages 55–62, Baltimore, Novembre 1995. Wiley.
- Whitaker (R.T.). Algorithms for Implicit Deformable Models. *Proc of 5th Int. Conf. Computer Vision (ICCV)*, 1995.
- Widrow (B.). The rubber mask technique, part I. *Pattern Recognition*, 5 (3), pages 175–211, 1973.
- Wilhelms (J.). Towards automatic motion control. *IEEE Computer Graphics and Applications*, 4 (7), pages 11–22, 1987.
- Witkin (A.) et Welch (W.). Fast animation and control of nonrigid structures. *Computer Graphics (SIGGRAPH'90)*, 24 (4), pages 243–252, Aot 1990.
- Wu (Y.), Thalmann (D.) et Magnenat Thalmann (N.). Deformable Surfaces using Physically Based Particle Systems. Earnshah (R.) et Vince (J.), editors, *Computer Graphics International*, pages 205–215. Academic Press, Juin 1995.
- Wyvill (Brian) et Wyvill (Geoff). Field functions for implicit surfaces. *The Visual Computer*, 5, pages 75–82, Dcembre 1989.
- Wyvill (Geoff), McPheeters (Craig) et Wyvill (Brian). Data structure for soft objects. *The Visual Computer*, pages 227–234, Aot 1986.
- Yserantant (H.). On the multi-level splitting of finite elements spaces. *Numerische Mathematik*, 49, pages 379–412, 1986.
- Zhang (Z.). Iterative point matching for registration of free-form curves. Technical Report 1658, INRIA, France, Mars 1992.



# Surface, implicit, and volumetric deformable models for medical imaging

## Abstract

The improvement of the medical images quality allows the obtention of volumetric images, which contain a great deal of information.

An efficient approach to treat these images consists in using the a priori knowledge of the shape of the objects to be analysed, and to employ intrinsically 3D methods. Deformable models meet both criterions.

We propose to formalise the deformable models and their evolution in a data image by distinguishing five components : binding characteristics, geometrical representation, deformation, deformability, and control.

We describe three deformable models.

We employ the delta-snakes surface model to reconstruct objets from unstructured points of their surface. We approximate this surface with an octree-spline distance map. We developed interactive tools to complete missing data or directly deform the surface.

Then we propose for the same kind of application an implicit model based on primitives generating a local potential field. The primitives are interactively placed, or automatically selected in the discrete medial axis of the data. The optimisation of the parameters of the primitives leads to a compact representation of the objects.

With these two models, we reconstruct objects digitized by range finders or segmented in volumetric images.

Our last model is volumetric. Its hierarchical deformation via an octree-spline minimizes a generalized distance between its characteristics and the ones of the data. It is realized under the control of the Levenberg-Marquardt algorithm, within the limits placed by a regularisation function. We developed an iterated generalised distance calculation algorithm in a kd-tree. We apply this model to volumetric images segmentation. Other kind of application have also been conducted.

## Key-words

Deformable model, surface, implicit, volumetric, generalized distance, distance map, octree-spline, optimization, interactivity, hierarchical deformation, kd-tree.



# Résumé

Les progrès des dispositifs d'imagerie médicale permettent l'obtention d'images volumiques, qui contiennent une grande quantité d'information. Une approche efficace de traitement de ces images consiste à utiliser la connaissance a priori de la forme des objets à analyser, et à employer des méthodes intrinsèquement tridimensionnelles. Les modèles déformables répondent à ces deux critères.

Nous proposons de formaliser les modèles déformables et leur évolution dans une image dite de données, en distinguant cinq composantes : caractéristiques de liaison, représentation géométrique, déformation, déformabilité, et contrôle.

Nous décrivons trois modèles déformables.

Nous employons le modèle surfacique des delta-snakes pour reconstruire des objets à partir de points répartis sur leur surface. Nous approximons cette surface par une carte de distance octree-spline. Nous avons mis au point des outils interactifs pour compléter des données manquantes ou déformer directement la surface.

Nous proposons ensuite pour ce même type d'application un modèle implicite à base de primitives générant un champ potentiel local. Les primitives sont placées interactivement, ou automatiquement sélectionnées dans l'axe médian discret des données. L'optimisation des paramètres des primitives mène à une représentation compacte des objets.

Nous reconstruisons par ces deux modèles des objets numérisés par des capteurs de distance ou segmentés dans des images volumiques.

Notre dernier modèle est volumique. Sa déformation hiérarchique par un octree-spline minimise la distance généralisée entre ses caractéristiques et celles des données, sous le contrôle de l'algorithme de Levenberg-Marquardt, et dans les limites imposées par une fonction de régularisation. Nous avons établi un algorithme de calcul de distance généralisée itérée dans un arbre k-d. Nous appliquons ce modèle à la segmentation d'images volumiques. D'autres types d'applications ont également été réalisées.

## Mots-Clef

Modèle déformable, surfacique, implicite, volumique, distance généralisée, carte de distance, octree-spline, interactivité, optimisation, déformation hiérarchique, arbre k-d.