

# Modélisation et indexation d'objets 3D à l'aide de superellipsoïdes

## THÈSE

présentée et soutenue publiquement le 6 Juillet 2004

pour l'obtention du

Doctorat de l'université Claude Bernard — Lyon 1  
(spécialité informatique)

par

Laurent Chevalier

### Composition du jury

<i>Rapporteurs :</i>	Kadi BOUATOUCH	IRISA, Université de Rennes 1
	Touradj EBRAHIMI	École Polytechnique Fédérale de Lausanne
<i>Examineurs :</i>	Mohammed DAOUDI	LIFL - Université des Sciences et Technologies de Lille
	Pascal GUITTON	LaBRI - Université de Bordeaux 1
	Atila BASKURT	LIRIS - INSA Lyon
	Fabrice JAILLET	LIRIS - Université Claude Bernard Lyon 1
<i>Invité :</i>	Bernard Péroche	LIRIS - Université Claude Bernard Lyon 1

Mis en page avec la classe thloria.

# Table des matières

<b>Introduction Générale</b>	<b>1</b>
<b>I Description d'objets tridimensionnels à l'aide de superellipsoïdes</b>	<b>5</b>
<b>Introduction</b>	<b>7</b>
<b>1 Représentation tridimensionnelle</b>	<b>11</b>
1.1 Maillages surfaciques tridimensionnels . . . . .	11
1.2 Modèles de surfaces paramétriques . . . . .	13
1.3 Modèles de surfaces implicites . . . . .	15
1.4 Les superquadriques . . . . .	18
1.5 Conclusion . . . . .	21
<b>2 Superellipsoïdes</b>	<b>23</b>
2.1 De la superellipse à la superellipsoïde . . . . .	23
2.2 Superellipsoïdes : équations générales . . . . .	28
2.3 Paramétrisation régulière d'une superellipsoïde . . . . .	29
2.4 Vecteur normal . . . . .	30
2.5 Distance d'un point à la surface d'une superellipsoïde . . . . .	31
2.6 Les autres classes de superquadriques . . . . .	39
2.7 Conclusion . . . . .	41
<b>3 Approximation d'un ensemble de points 3D par une superellipsoïde</b>	<b>43</b>
3.1 Estimation des paramètres . . . . .	44
3.1.1 Ellipsoïde d'inertie . . . . .	44

3.1.2	Estimation directe en se basant sur l'approximation directe d'ellipse . . . . .	47
3.2	Minimisation de l'énergie . . . . .	50
3.3	Conclusion . . . . .	59
<b>4</b>	<b>Segmentation des données tridimensionnelles</b>	<b>61</b>
4.1	Local à global : croissance de région . . . . .	61
4.1.1	Initialisation des germes . . . . .	62
4.1.2	Phase de croissance . . . . .	62
4.1.3	Phase de sélection . . . . .	65
4.1.4	Résultats . . . . .	66
4.1.5	Conclusion . . . . .	68
4.2	Global à local : découpage-fusion (split and merge) . . . . .	69
4.2.1	Phase de découpage (split) . . . . .	70
4.2.2	Phase de fusion (merge) . . . . .	73
4.2.3	Résultats . . . . .	76
4.2.4	Structure du descripteur . . . . .	80
4.2.5	Commentaires . . . . .	82
4.3	Conclusion . . . . .	82
	<b>Conclusion</b>	<b>83</b>
<b>II</b>	<b>Utilisation du descripteur d'objets pour l'indexation et la comparaison d'objets 3D</b>	<b>85</b>
	<b>Introduction</b>	<b>87</b>
<b>5</b>	<b>État de l'art sur l'indexation et la comparaison d'objets 3D</b>	<b>89</b>
5.1	Approches statistiques . . . . .	89
5.1.1	Approches statistiques locales . . . . .	90
5.1.2	Approches statistiques globales . . . . .	91
5.2	Approches multi-vues . . . . .	92
5.3	Approches structurales . . . . .	92
5.4	Conclusion . . . . .	94

<b>6 Comparaisons de graphes</b>	<b>95</b>
6.1 Généralités . . . . .	95
6.2 La distance d'édition . . . . .	103
6.2.1 Comparaison de chaînes de caractères . . . . .	103
6.2.2 Adaptations aux graphes . . . . .	104
6.3 Conclusion . . . . .	107
<b>7 Comparaison d'objets tridimensionnels</b>	<b>109</b>
7.1 Choix et coût des fonctions d'édition . . . . .	109
7.1.1 Relabélisation ou mesure de similarité entre superellipsoïdes . . . . .	111
7.1.2 Fusion de superellipsoïdes . . . . .	122
7.2 Test de validations . . . . .	128
7.3 Conclusion . . . . .	136
<b>Conclusion</b>	<b>137</b>
<b>Conclusion générale</b>	<b>141</b>
<b>Publications</b>	<b>145</b>
<b>Bibliographie</b>	<b>147</b>



# Introduction Générale

L'informatique a désormais investi la plupart des occupations humaines. Les applications nécessitant une représentation précise ou fictive du monde physique sont nombreuses : jeux vidéo, applications médicales ou industrielles, guidage de véhicule ou de robots intelligents, etc. Les technologies permettant l'acquisition de cette réalité tridimensionnelle ont considérablement évolué que ce soit en accessibilité, rapidité ou précision. Il est donc légitime que leur utilisation soit de plus en plus courante. Ainsi, le volume des données 3D que ce soit en nombre d'objets ou en taille de chacun d'entre-eux, a considérablement augmenté.

Ces données se présentent sous la forme d'un ensemble de points  $P = \{P_1, P_2, P_3 \dots P_N\} \in \mathbb{R}^3$  échantillonnés à la surface de l'objet considéré. La régularité de la densité de cet échantillon ainsi que son organisation (par exemple la connaissance de relation de voisinage entre les points) dépendent de la méthode d'acquisition utilisée. Toutefois, pour la plupart des applications, ne serait-ce que pour la visualisation, elles ne sont généralement pas exploitables directement. Une étape de reconstruction ou modélisation de la surface de l'objet, esquissée par cet ensemble de points, est nécessaire. La méthode la plus courante est le maillage de cette surface par un ensemble de polygones ; ce qui correspond, dans sa version la plus directe, en une interpolation linéaire des points échantillonnés. Cette représentation est certes performante pour l'affichage rapide, mais possède les défauts de n'être que très bas niveau et donc de ne fournir que peu d'informations sur l'objet et surtout d'être très volumineuse. Ce problème du volume de la représentation est important non seulement en ce qui concerne le stockage bien sûr, mais aussi et surtout parce que parallèlement à l'amplification de l'utilisation de la 3D, s'est développé tout un ensemble de services réseaux aujourd'hui omniprésents. Une partie de la problématique est de concilier ces deux domaines. Une application représentative de ce phénomène est sans aucun doute la recherche de modèles tridimensionnels dans des bases de données à travers internet. En effet, il s'agit alors à la fois de transmettre les modèles, de les visualiser (peut être aussi de les pré-visualiser grâce à un modèle simplifié) et d'avoir une méthode permettant de les comparer pour la recherche.

Ainsi, il est désormais important de disposer de descripteurs d'objets tridimensionnels performants, à la fois compacts, permettant un affichage rapide de l'objet qui peut être,

au besoin, simplifié (affichage basse ou moyenne résolution) et aptes à être exploités pour la comparaison des modèles entre eux que ce soit pour des applications de reconnaissance ou de recherche dans des bases de données.

L'approche proposée dans cette étude est de décrire les objets tridimensionnels par un assemblage de primitives : les superellipsoïdes. Outre des propriétés intéressantes liées à sa formulation mixte, ce modèle a pour principal avantage une grande capacité de description pour un nombre très réduit de paramètres. Cinq paramètres seulement vont permettre de définir des ellipsoïdes, des cylindres, des parallélépipèdes, des octoèdres et toutes les formes intermédiaires et de toutes tailles.

N'ayant pas de connaissance *a priori* sur les données, le passage de l'ensemble de points tridimensionnels à l'assemblage de superellipsoïdes sera nécessairement précédé d'une étape de segmentation. Cette dernière a pour but de distinguer les parties (i.e. les sous-ensembles de points) dont se composent l'objet 3D. Ces parties incarnent des entités au sens physique et sémantique. En effet, cette segmentation correspond à l'interprétation des données. Le choix d'une description par une combinaison de primitives n'est pas innocent. Il nous permet à la fois de traiter des problèmes de visualisation comme pour toutes les méthodes de reconstruction, mais aussi d'aborder grâce à des informations de haut niveau sur la structure des données, des applications de type reconnaissance d'objet.

Ce mémoire est divisé en deux parties bien distinctes. Les détails de notre méthode permettant d'obtenir cette description des données tridimensionnelles font l'objet de la première partie. Dans celle-ci, nous présentons successivement une étude détaillée des superellipsoïdes, la brique élémentaire de notre descripteur, puis un moyen d'approximer des données 3D par une superellipsoïde et enfin nous proposons une méthode de segmentation applicable à des ensembles de points non-organisés et sans condition d'une quelconque régularité dans l'échantillonnage. Comme annoncé, nous obtenons un descripteur de surfaces composé d'un ensemble de superellipsoïdes. Cet assemblage a de plus l'avantage d'être structuré grâce à un graphe témoignant des relations topologiques entre les primitives : les sommets représentent les superellipsoïdes et les arêtes notifient une relation d'adjacence entre ces dernières.

Nous n'avons pas voulu obtenir un descripteur dédié à une application précise mais apte, au besoin après une étape de spécialisation, pour des exploitations diverses. Les applications que nous visons sont celles que nous avons évoquées dans un paragraphe précédent. Elles concernent les domaines de la visualisation simplifiée ou progressive, adaptée à des applications multimédia sur des réseaux à faible ou moyen débit, mais aussi à des applications de type reconnaissance ou vision, où le descripteur est utilisé en tant que

discriminateur d'objets. La recherche dans des bases de données d'objets tridimensionnels et l'indexation font donc partie de cette seconde catégorie.

Ainsi, la seconde partie de ce travail traite de l'indexation d'objets 3D à l'aide de notre descripteur. Les deux parties de ce mémoire forment à la fois la continuité d'une chaîne complète et suffisante, mais sont aussi relativement indépendantes. En effet, d'une part, comme nous l'avons déjà évoqué, nous avons souhaité un descripteur adaptable à des domaines divers. La première partie peut en effet constituer le point de départ pour un travail futur dans un autre domaine traitant des objets 3D. D'autre part, nous ne supposons en aucune façon, dans cette seconde partie, que le descripteur ait été obtenu par notre méthode. De plus, il peut avoir été retouché manuellement par l'utilisateur, cela est encore parfois le cas en indexation où l'interaction avec un expert est souvent requise. Aussi, on peut imaginer une application de recherche d'objet où la requête de l'utilisateur n'est pas de trouver un objet proche d'un objet déjà existant dont il possède une version numérique, mais simplement une esquisse qu'il construira manuellement dans le format de notre descripteur. Dans ce dernier cas, cet utilisateur sera sans doute très heureux de disposer d'une approche de description intuitive comme celle que nous proposons, car comme nous le verrons dans le chapitre 5, de nombreuses méthodes d'indexations utilisent des descripteurs basés, par exemple, sur des distributions statistiques de courbures ou des transformées rendant ce genre de requêtes beaucoup plus difficiles et indirectes.

Comme nous avons déjà le descripteur qui nous servira à l'indexation d'objets tridimensionnels, il ne reste plus dans cette deuxième partie qu'à trouver un moyen nous permettant d'établir une mesure de similarité entre deux de nos descripteurs et, ce faisant, de comparer des objets. Notre description est obtenue par une décomposition spatiale de l'objet, il est donc évident que le moyen de comparaison sera basé sur une analyse structurale. Ainsi, la mesure de similarité entre les descripteurs se transforme en un problème d'appariement des graphes sous-jacents. Les méthodes classiques de mise en correspondance, telles que les isomorphismes de graphes ou de sous-graphes ont le défaut d'être la plupart du temps trop contraignantes et strictes, elles ne s'accordent que peu avec la comparaison d'objets qui se doit de rester un peu lâche. En effet, les structures de deux objets similaires ne sont pas forcément exactement les mêmes. Ainsi, pour cette comparaison, nous utilisons une méthode basée sur le calcul du coût minimum en modifications permettant de transformer un graphe en un autre. Après l'étude détaillée de cette méthode appelée distance d'édition, la définition des opérations de modifications de graphe nécessaires et l'instauration de leurs coûts constituent le propos principal de cette seconde partie du mémoire. Bien sûr, nous présentons aussi les résultats portant sur la comparaisons d'objets et l'indexation.

Une des difficultés et donc un des intérêts de cette étude est d'aborder des domaines

très différents. En effet, la plupart des travaux portant sur les thèmes que nous explorons sont souvent relativement dissociés. Certains vont se focaliser sur le problème de la reconstruction de surface. D'autres se concentrent sur la comparaison d'objets et l'indexation. Cela pour la bonne raison que les approches portant sur l'indexation 3D s'intéressent essentiellement aux objets représentés sous forme de maillages (déjà reconstruits donc) qui constituent la plus grande proportion des bases de données tridimensionnelles, ou alors ont l'indexation pour unique objectif et donc utilisent des méthodes se passant de reconstruction. Nous avons essayé de considérer toute la chaîne de traitement en partant de données brutes le plus générales possibles. Nous traitons à la fois des problèmes bas niveau tels que l'approximation de surfaces, la segmentation de données tridimensionnelles, mais aussi d'idées de plus haut niveau sémantique avec la comparaison de structures d'objets.

## Première partie

# Description d'objets tridimensionnels à l'aide de superellipsoïdes



# Introduction

Le sujet de cette première partie est d'obtenir un descripteur de surface, compact et apte à diverses exploitations (analyse, segmentation, modélisation, visualisation), basé sur une décomposition de données tridimensionnelles en un ensemble de primitives.

Diverses technologies peuvent être utilisées pour acquérir une représentation numérique d'objets du monde réel. Une présentation assez courante de ces données est appelée image de profondeur et est obtenue par un système de capteurs et de lasers. À la différence d'une simple image, chaque pixel ne contient pas une information de couleur, mais de profondeur. Les données ne renseignent donc que sur une face de l'objet et sont régulièrement réparties suivant une grille bidimensionnelle. Différentes méthodes permettent de mettre en correspondance ces données tirées de différentes vues, et ainsi d'obtenir une représentation complète de l'objet mais avec perte de régularité due à la grille. L'acquisition des données 3D peut aussi être faite point par point grâce à un capteur mécanique. Le point à numériser est mis en contact avec l'extrémité d'un stylet tactile, on mesure alors la différence de la position du stylet et d'un point de référence. Dans le domaine médical, des informations sur les organes internes peuvent être obtenues grâce aux techniques de tomographie (CT) ou d'imagerie à résonance magnétique (IRM). Les données ainsi acquises ne sont pas simplement surfaciques mais volumiques, et sont présentées sous la forme d'un ensemble de voxels. Dans différents domaines mais surtout en ce qui concerne le médical, les données peuvent aussi se présenter structurées en un empilement de contours de sections 2D de l'objet considéré. Dans cette étude, nous nous intéressons à la présentation de ces données sous leur forme la plus générale possible. Nos données sont un ensemble de points de la surface de l'objet. Nous ne supposons pas d'organisation particulière de ces données (comme la connaissance du voisinage) ni d'une quelconque régularité de la densité d'échantillonnage. Nous avons donc affaire à un ensemble quelconque de points de  $\mathbb{R}^3$ .

Nous proposons de décrire ces données tridimensionnelles par un ensemble de primitives, cette description nécessite donc une étape de segmentation. La segmentation dans sa généralité consiste à étiqueter chaque point des données de telle façon que des points appartenant à une même partie de l'objet reçoive la même étiquette. Les parties sont caractérisées comme homogènes suivant un certain critère. L'interprétation des données

que nous allons segmenter, la notion de forme et les informations sémantiques que nous allons en déduire sont directement induites par le choix du critère d'homogénéité.

Avant même de considérer les différents algorithmes possibles, notons que plusieurs approches de segmentation existent. Une première, que l'on peut qualifier d'approche frontière, s'applique à déterminer uniquement les limites des parties des objets. Typiquement, les critères d'homogénéité permettant de déterminer ces frontières peuvent être basés sur l'étude des lignes de courbures de la surface de l'objet. La seconde approche est celle que nous employons. Pour décomposer les données en sous-parties, elle utilise directement la forme de ces parties comme critère d'homogénéité. Ainsi, durant la segmentation, on ne fait pas qu'un étiquetage des points et une localisation des sous-parties, on en profite pour déterminer leur forme en même temps. La capacité de description de nos primitives est donc cruciale car elle influe à la fois sur la précision de notre descripteur mais aussi sur la segmentation.

La première partie de ce mémoire est divisée en cinq chapitres.

Le premier est consacré au tour d'horizon des différents modèles de modélisation surfacique. Nous y présentons bien sûr leurs capacités pour la reconstruction de données tridimensionnelles. Mais, nous désirons que notre descripteur d'objets ne soit pas limité à un seul domaine d'application, nous mettons donc en avant les avantages et inconvénients de leur exploitation dans différents domaines.

Dans le second chapitre, nous présentons en détail le modèle que nous utilisons : les superellipsoïdes. Leurs différentes propriétés sont analysées, et notamment la mesure de la distance d'un point de l'espace tridimensionnel à la surface d'une superellipsoïde étant cruciale pour l'approximation des données, nous exposons une étude détaillée des différentes méthodes ayant été utilisées en soulignant leurs qualités et leurs faiblesses.

Avant de pouvoir décomposer nos données en une structure de primitives élémentaires, nous devons apprendre comment la surface que dessine un ensemble de points 3D peut être approximée par une seule superellipsoïde. Le troisième chapitre de cette partie est dédié à cette étude. La méthode que nous utilisons est une approximation au sens des moindres carrés de la distance des points à la surface. Elle nécessite deux étapes principales : une première estimation de la superellipsoïde correspondante aux données, puis une méthode de régression non-linéaire permettant d'optimiser la solution. Pour ces deux étapes, plusieurs approches sont expérimentées. Ainsi, nous comparons différentes méthodes d'optimisations : algorithme génétique, méthode du simplex et algorithme de Levenberg-Marquardt.

La segmentation des données et l'obtention du descripteur proprement dit sont l'objet

du quatrième chapitre. Nous proposons deux méthodes de segmentation. La première, inspirée de la méthode de Leonardis [LJS97], est une approche *local à global* basée sur un algorithme de croissance de régions. La seconde est une méthode originale de type *global à local* et suit un procédé de découpage-fusion. Comme nous venons de le voir, notre approche de segmentation détermine simultanément la décomposition en parties et la forme de ces parties (i.e. l'approximation par une superellipsoïde). À la fin de ce chapitre, nous obtenons donc complètement notre descripteur de surface. Nous exposons les résultats obtenus par ces deux approches.

Enfin, le dernier chapitre de cette première partie est consacré à la conclusion sur l'obtention de notre descripteur mais aussi aux perspectives et notamment à ses exploitations possibles.



# Chapitre 1

## Représentation tridimensionnelle

Dans ce chapitre sont présentés les différents modèles de représentation tridimensionnelle pouvant être utilisés dans le cadre d'une reconstruction d'objet à partir d'un ensemble de points de sa surface. Pour chacun d'entre-eux, nous en exposerons les avantages et les inconvénients ainsi que leurs utilisations classiques. Pour notre problématique, rappelons que le modèle recherché doit être à la fois compact et fonctionnel pour des applications diverses. Nous nous contenterons donc des modèles relativement généralistes et n'aborderons que le cas des modèles géométriques.

Ainsi, la première partie est consacrée aux incontournables maillages tridimensionnels. Les surfaces paramétriques ainsi que les surfaces implicites seront ensuite successivement présentées. Enfin, nous aborderons un modèle mixte : les superquadriques.

### 1.1 Maillages surfaciques tridimensionnels

Un modèle tridimensionnel surfacique représenté sous forme de maillage est un ensemble de polygones plans défini par une liste de sommets (i.e. des points de l'espace 3D) et une liste d'arêtes reliant ces derniers (i.e. des couples de sommets). Ces polygones sont contigus : ils se partagent des arêtes. Les polygones peuvent être de tout type, mais le plus utilisé est le triangle ; on parle alors de maillage triangulaire.

Ce type de représentation est de loin le plus courant en informatique. La principale raison à cela est sa capacité à être affichée en temps réel même dans le cas d'une scène complexe. En effet, ce type de représentation est géré directement par les cartes graphiques 3D maintenant présentes dans la plupart des ordinateurs. Ces cartes intègrent même différents filtres et autres procédés parfois assez élaborés afin d'améliorer la qualité du rendu tout en gardant un affichage très rapide.

En conséquence, les maillages tridimensionnels sont incontournables. Pour preuve, l'affichage de l'ensemble des modèles existants de représentation tridimensionnelle se fera

par l'intermédiaire d'une facétisation. Pour un affichage rapide, la capacité de ces modèles à être échantillonnés et polygonalisés est donc essentielle.

Ce modèle de surface étant un assemblage de polygones plans, il ne pourra décrire exactement que les objets composés uniquement de surfaces planes. La représentation de surfaces courbes n'en sera qu'une approximation. Bien sûr, il est toujours possible d'améliorer cette approximation en augmentant le nombre de facettes mais la taille du modèle augmentera elle aussi en conséquence. De plus, on peut remarquer que si la qualité de représentation paraît correcte à un certain niveau d'échelle, ce ne sera plus forcément le cas si on se rapproche du modèle. La représentation par maillage tridimensionnel a donc ce gros défaut d'être extrêmement volumineuse.

Ainsi, la compression de maillages est un enjeu important de la recherche. Pour un état de l'art détaillé sur le sujet, on peut se référer à l'article de Rossignac [TR98] ou celui de Alliez [AD01]. Succinctement, l'idée est de coupler le codage des sommets et leur connectivité. Les coordonnées des sommets sont, en général, prédites grossièrement des coordonnées des voisins puis corrigées. Seule la correction est effectivement codée. La compression de maillages se divise en deux écoles (Figure 1.1).

La première que l'on peut qualifier de compression classique mono-résolution se base généralement sur un parcours spécifique et déterminant du maillage découpé en bandes.

La seconde approche, introduite par Hoppe [Hop96], propose une compression multi-résolution. Elle s'intéresse plus à la transmission des modèles et suggère un codage permettant une décompression basée sur le raffinement progressif de l'objet. Pour ce faire, ces méthodes s'appuient sur différentes méthodes de simplification de maillage en supprimant des arêtes, supprimant ou fusionnant des sommets. Vallette [S02], par exemple, utilise une décomposition en ondelettes pour l'analyse multi-résolution du maillage.

Une partie du standard international MPEG-4 (MPEG-4Systems, MPEG-4Visual) a trait à la représentation tridimensionnelle et offre des technologies de codage de maillages 3D. Celles-ci incluent des techniques de compression aussi bien mono que multi-résolution. L'approche monorésolution adoptée est celle de Taubin et Rossignac [TR98] pour le codage de la connexité, combinée au schéma de prédiction par règle du parallélogramme proposée par Touma et Gotsman [TG98] pour le codage de la géométrie et des attributs. La méthode de compression multirésolution adoptée par MPEG-4 s'appuie sur la technique de *Progressive Forest Split* proposée par Taubin.

De nombreuses méthodes permettent le maillage d'ensembles de points non-organisés. Si cette facétisation est généralement tout à fait suffisante et satisfaisante dans le cadre d'applications de type affichage, indépendamment du fait que la représentation est très lourde dans le cas de modèle de haute précision, elle ne contient, en l'état, aucune information de plus haut niveau, et est donc très peu intéressante en ce qui nous concerne.

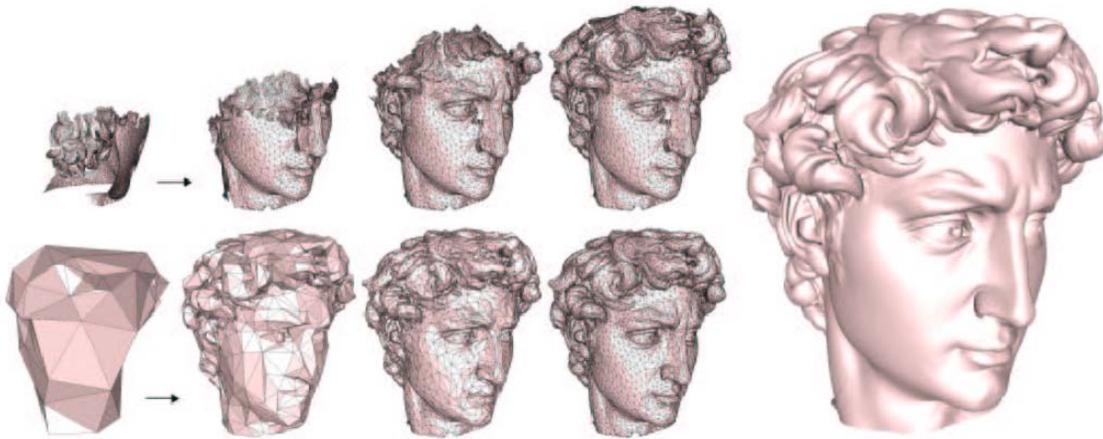


FIG. 1.1 – Différentes étapes de la transmission d'un modèle suivant les deux approches de la compression de maillages. En haut, la compression mono-résolution progresse de haut en bas. En bas, l'approche multirésolution progresse par raffinement progressif du modèle (figure extraite de [AD01])

Malgré ses défauts, son omniprésence dans le monde de la représentation tridimensionnelle fait que le maillage surfacique 3D est à l'origine de nombreuses études dans des domaines très variés. Nous verrons dans le chapitre consacré à un aperçu de l'indexation 3D (partie 2 chapitre 5) qu'il constitue la base de nombreux travaux de recherche.

## 1.2 Modèles de surfaces paramétriques

La représentation paramétrique est caractérisée par la définition de chaque point  $S(\eta, \mu)$  de la surface par une équation du type :

$$S(\eta, \mu) = \begin{bmatrix} f_x(\eta, \mu) \\ f_y(\eta, \mu) \\ f_z(\eta, \mu) \end{bmatrix}$$

$\eta$  et  $\mu$  étant appelées les variables paramétriques et  $f_x$ ,  $f_y$  et  $f_z$  étant des fonctions de  $\mathbb{R}^2$  dans  $\mathbb{R}$ .

Les modèles paramétriques sont une grande famille de surfaces. Les fonctions qui les définissent peuvent être diverses. Cependant, nous n'allons pas rester dans une présentation du cas général, nous nous concentrerons sur la sous-famille des courbes de Bézier et de leurs extensions (splines), qui est la plus connue et dont l'utilisation est très répandue dans le domaine de la CFAO/CAO entre autres.

Pierre Bézier introduit les surfaces paramétriques dans le domaine de la conception informatique au début des années 60 [Béz87]. Les courbes et surfaces de Bézier sont

caractérisées par une série de points appelés points de contrôle ou pôles. La combinaison linéaire de ces pôles suivant deux directions  $\eta$  et  $\mu$  fournit tous les points  $S(\eta, \mu)$  de la surface :

$$S(\eta, \mu) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} B_i^n(\eta) B_j^m(\mu)$$

avec  $P_{i,j}$  définissant le réseau des points de contrôle de la surface et  $B_k^n(t) = C_n^k t^k (1-t)^{n-k}$  le polynôme de Bernstein (avec  $t \in [0, 1]$ ).

Ces courbes et surfaces ont permis de remplacer les méthodes traditionnelles de modélisation. Le modèle de Bézier a longtemps été utilisé mais possède deux inconvénients majeurs :

- le déplacement d'un point de contrôle entraîne des modifications sur l'ensemble de la courbe ou de la surface ;
- l'augmentation du degré du polynôme avec le nombre de pôles rend très coûteux l'utilisation de polygones de contrôle complexes.

Une autre forme de cette famille sont les B-Splines, où le polynôme de Bernstein est abandonné au profit d'un autre polynôme en guise de fonction de base. Les B-Splines permettent alors des modifications locales de la surface (le déplacement d'un point de contrôle ne bouleverse pas l'ensemble de la surface) et la complexité du polynôme de base ne dépend plus du nombre de points.

L'évolution importante suivante sera les NURBS (Non Uniform Rational B-Spline). La fonction de base est une fonction rationnelle, et une pondération est associée à chaque point de contrôle. Le potentiel de description devient alors vraiment très grand. Elles permettent toujours de définir des surfaces de Bézier ou des B-splines, mais aussi des surfaces rationnelles comme des morceaux de sphères ; ce qui était impossible jusqu'alors. Pour cette raison, les NURBS sont incluses dans différents formats d'échange de données de CAO comme IGES. Récemment, elles ont été ajoutées au format VRML et X3D (eX-tensible 3D) dédié à internet, ce qui atteste bien de leur pertinence. Elles font aussi partie de la palette d'outils de représentation tridimensionnelle proposée par MPEG4 avec AFX (Animation eXtension Framework) au côté d'un autre type de surfaces paramétriques : les surfaces de subdivision.

La compression de surfaces paramétrées est un domaine qui commence à émerger dans la littérature, parallèlement à l'utilisation grandissante de ce type de surfaces. Un schéma de compression de surface NURBS est par exemple présenté par Santa Cruz et Ebrahimi dans [Cru03]. Les noeuds et les points de contrôle sont codés de manière prédictive (codage DCPM) et les erreurs de prédiction sont codées par entropie en utilisant le codeur MQ de JPEG2000. Le taux de compression obtenu est d'environ 5 par rapport au maillage codé par la méthode présente dans MPEG4 avec une meilleure qualité visuelle pour une erreur

donnée.

Outre leur pouvoir descriptif, le modèle de surface de type NURBS (et ses prédécesseurs) a plusieurs avantages. Tout d'abord, il est relativement compact. De plus, même en réduisant nombre de points de contrôle, la surface reste lisse (il n'y a donc aucun problème de visualisation lors de changement de facteur d'échelle). Par définition, tous les points de la surface sont connus et directement accessibles. Pour échantillonner la surface, afin d'obtenir un maillage lors de l'affichage par exemple, il suffit d'échantillonner les paramètres. Même si le calcul de ces points peut parfois s'avérer coûteux, la méthode est beaucoup plus simple que dans le cas des modèles implicites que nous verrons plus tard (paragraphe 1.3).

Cependant, un défaut de ce type de surfaces est que le réseau de points de contrôle sous-jacent est forcément rectangulaire, ce qui les rend difficilement plaquables sur un objet de topologie complexe (objet avec des trous par exemple). Il est bien sûr possible de les utiliser par carreaux mais il faut alors prévoir le raccordement des patches [Per92].

En ce qui concerne les techniques d'approximation et de reconstruction par courbes et surfaces de type spline, on peut se référer à l'état de l'art de Dierckx [Die93]. On peut aussi consulter le livre faisant référence en matière de NURBS [Far93].

## 1.3 Modèles de surfaces implicites

La caractéristique des surfaces implicites est d'être définies comme la solution d'une équation de  $\mathbb{R}^3$  dans  $\mathbb{R}$ . L'ensemble des points  $P = (x, y, z)$  de la surface implicite  $S$  définie par  $f$  est celui qui vérifie :

$$f(x, y, z) = 0$$

Des formes basiques très communes ont souvent l'habitude d'être exprimées sous cette forme, par exemple la droite :  $ax + by + cz + d = 0$  ou bien la sphère  $x^2 + y^2 + z^2 - r^2 = 0$ .

Grâce à cette formulation, une propriété de ce modèle de représentation est de donner de manière directe des informations de relation entre tout point de l'espace tridimensionnel et la surface. Ainsi, se renseigner pour savoir si un point est à l'intérieur ou à l'extérieur du modèle s'avère très simple. Le signe de la fonction caractérisant la surface implicite définit un véritable partitionnement de l'espace et un point  $P = (x, y, z)$  tel que :

- $f(x, y, z) > 0$  sera à l'extérieur de l'objet modélisé ;
- $f(x, y, z) < 0$  sera à l'intérieur de l'objet modélisé ;
- $f(x, y, z) = 0$  sera sur la surface de l'objet modélisé.

Ainsi, ce modèle de surfaces n'est pas limité à la représentation surfacique comme les modèles paramétriques présentés précédemment, mais peut aussi être utilisé dans le cadre de représentation volumique.

Il va en résulter que les algorithmes d'intersection, de collision ou de relation entre un point et le modèle vont pouvoir être simplifiés. Afin d'obtenir un objet plus complexe, la combinaison de différentes surfaces grâce à des opérations ensemblistes de type union, différence ou intersection de surfaces est aisée (Figure 1.2).

Par exemple, pour obtenir l'intersection  $S_I$  de deux formes implicites définies par  $f_1$  et  $f_2$ , il suffit de remarquer que les points de  $S_I$  sont ceux à la fois sur une des surfaces ( $f_1(x, y, z) = 0$  ou  $f_2(x, y, z) = 0$ ) et à l'intérieur de l'autre ( $f_1(x, y, z) < 0$  ou  $f_2(x, y, z) < 0$ ). Les points de  $S_I$  sont donc les solutions de :

$$\max(f_1(x, y, z), f_2(x, y, z)) = 0$$

Notons que le résultat est une fonction implicite, les opérations vont donc pouvoir se faire en cascade, de la même manière.

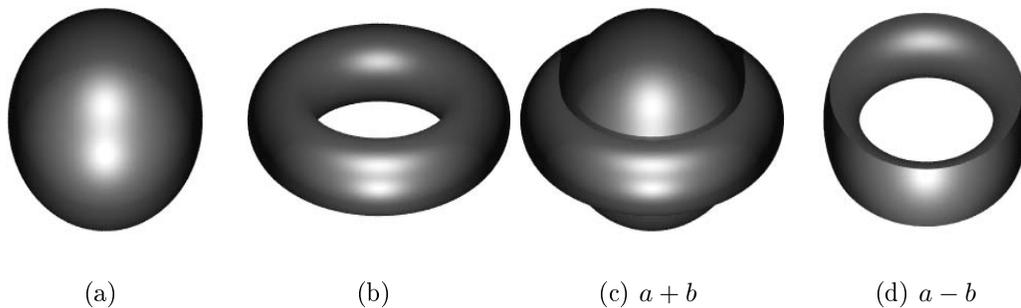


FIG. 1.2 – Exemples de combinaisons de surfaces implicites

La construction d'objets à partir d'une ou plusieurs fonctions implicites très générales et plus ou moins complexes est possible (on parle alors de modélisation algébrique), mais d'autres types de formulation ont été développés afin de rendre la modélisation plus intuitive.

Ainsi en 1982, Blinn propose l'utilisation d'un ensemble d'objets élémentaires très simples [Bli82]. Ces derniers appelés *blobs* sont définis par un point et une fonction 'potentiel' qui va décroître de manière exponentielle lorsqu'on s'éloigne de l'objet. Chaque point de l'espace 3D se voit alors attribué une valeur égale à la somme des potentiels en ce point pour chacun des blobs. La surface de l'objet est définie par l'ensemble des points de l'espace ayant un même potentiel donné (surface iso-potentiel) (Figure 1.3). Sur le même principe, il existe des variantes proposées par Nishimura (metaballs) [NHK<sup>+</sup>85] et les 'soft

objects' de Wywill [WMW86] où, entre autres, les fonctions de potentiel sont modifiées pour être moins coûteuses. En 1991, Muraki [Mur91] propose une méthode (relativement lourde) de reconstruction de surface à l'aide de blobs.

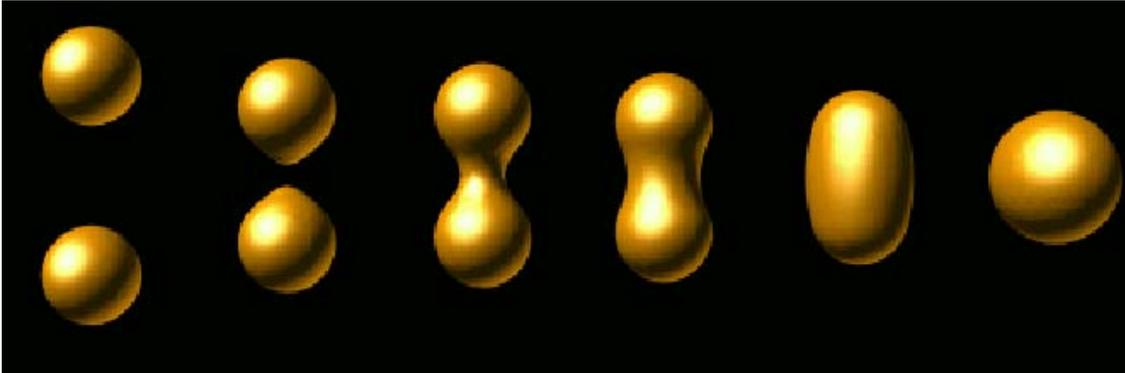


FIG. 1.3 – Effet de la distance sur le mélange des blobs.

Les blobs peuvent être considérés comme le cas particulier d'une formulation plus générale à base de fonctions implicites : la modélisation par squelette. On ne considère alors plus uniquement les points pour définir le noyau des particules élémentaires mais des primitives quelconques. Néanmoins, en pratique la représentation par squelette se limite en général à l'utilisation de points, de segments et de facettes planes.

Pour plus de liberté et de facilité pour la modélisation, des variantes du modèle original vont permettre aux éléments squelettiques de contribuer de manière négative au champ de potentiel permettant ainsi de creuser dans le modèle, et d'autre part des graphes de contribution vont être mis en place pour contrôler l'influence du potentiel des éléments squelettiques entre-eux.

La modélisation par squelette est très intuitive et peut être utilisée pour la représentation d'objets très divers de topologies simples ou complexes (le problème des objets à trous ne se pose pas dans le cas d'une modélisation implicite). Outre leur propension aux mélanges et aux combinaisons, les surfaces implicites offrent de grandes facilités dans le domaine des déformations. Cela en fait un modèle très utilisé dans le domaine de l'animation et la simulation. Et plus précisément, du fait de leur capacité à définir un volume, elles sont appréciées dans le domaine de la modélisation d'organes et la simulation médicale [Pon96, Amr02]. On peut aussi remarquer que ce type de représentation est particulièrement compact.

Le principal défaut des modèles implicites reste leur visualisation très lourde. Nous ne parlons pas ici d'un rendu de type lancer de rayons où, dans ce cas, bien entendu les modèles implicites sont avantagés par leurs facilités dans le domaine du calcul d'intersec-

tions et de partitionnement de l'espace, mais d'un affichage à base de facettes. En effet, à l'inverse des surfaces paramétriques où les points de la surface sont obtenus directement par la définition de la surface, l'échantillonnage, et par extension le maillage, de surfaces implicites est difficile et passe par des algorithmes lourds comme le "Marching Cube" [LC87, GA00].

## 1.4 Les superquadriques

Les superquadriques ont été introduites en informatique graphique par Barr [Bar81] au début des années 80 et plus spécifiquement dans le domaine de la reconstruction et de la vision par Pentland [Pen91], Solina et Bajski [BS87]. Comme leur nom le laisse présager, elles sont une extension des quadriques. Deux paramètres supplémentaires  $\epsilon_1$  et  $\epsilon_2$  vont permettre d'en contrôler la courbure latitudinale et longitudinale. Cette extension est appliquée aux quatre classes des quadriques, on pourra donc trouver des superparaboloïdes, des superhyperboloïdes, des superellipsoïdes et des supertoroïdes. Nous étudierons en détails les propriétés et la construction des superquadriques et notamment des superellipsoïdes dans le chapitre suivant.

Une particularité intéressante des superquadriques est d'être définissable à la fois sous forme implicite (elle se classe dans la catégorie des surfaces algébriques) mais aussi sous forme paramétrique. Par exemple, une superellipsoïde est définie comme la solution de :

$$F(x, y, z) = \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left( \frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1$$

et sa formulation paramétrique est :

$$S(\eta, \mu) = \begin{bmatrix} a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\mu) \\ a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\mu) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix}, \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \mu \leq \pi \end{array}$$

Les paramètres  $a_1, a_2, a_3$  contrôlent la taille de la superellipsoïde comme dans le cas des ellipsoïdes tandis que  $\epsilon_1$  et  $\epsilon_2$  en contrôlent la courbure. Les superquadriques sont donc un modèle très compact et de plus, cette compacité ne les prive pas d'une grande capacité de description. Ainsi défini avec seulement 5 paramètres, ce modèle de surface va permettre la représentation d'un éventail très large de formes. Les superellipsoïdes pourront décrire des cubes, des cylindres, des octoèdres, des ellipsoïdes de taille quelconque (Figure 1.4).

Dans les deux formulations, les mêmes paramètres sont utilisés pour les définir, on peut donc passer d'une forme à l'autre de manière directe. Comme nous avons vu que les surfaces paramétriques et les surfaces implicites ont leurs avantages et leurs inconvénients dans des domaines plus ou moins opposés, cette propriété est très intéressante. Suivant

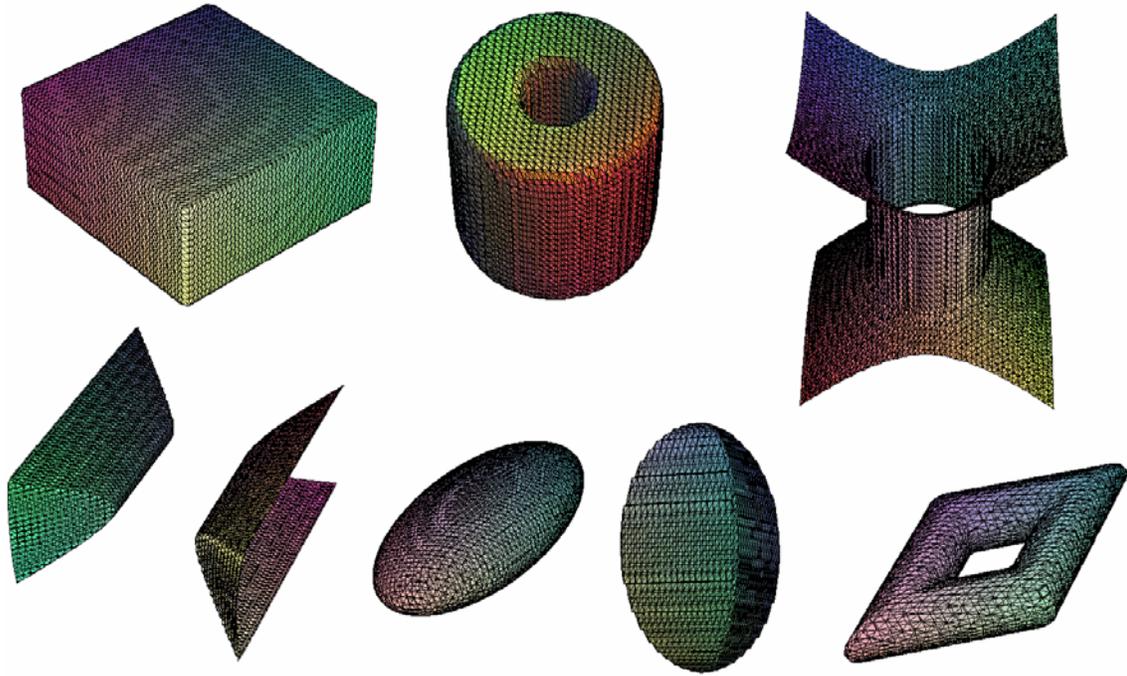


FIG. 1.4 – Exemples de superquadriques.

l'application, on va donc, sans contrainte, choisir l'une ou l'autre forme : le calcul d'intersection et de calcul de position de point sera facile avec l'équation implicite tandis que la formule paramétrique permettra un échantillonnage, un maillage et donc un affichage très simple et rapide.

L'idée de déformer les superquadriques apparaît dès les premiers articles traitant de ce modèle : Barr [Bar84], par exemple, propose des déformations globales : torsions, pliages... En effet, malgré un potentiel de description assez important, les formes obtenues restent assez simples, et surtout très symétriques. Les déformations définies par Barr seront reprises par Solina et Bajski [SB90] dans l'article proposant la méthode aujourd'hui la plus commune pour approximer des données tridimensionnelles avec des superellipsoïdes. Terzopoulos et Metaxas [TM91] proposent d'ajouter aux déformations globales une déformation locale.

Une superquadrique déformée est utilisée pour modéliser et étudier le suivi d'un ventricule par Park et Metaxas [PMY94] et Bardinet [BCA95]. Dans le premier cas, la déformation est produite par l'utilisation de fonctions à la place des paramètres constants. Dans le second, après avoir fait une première approximation avec une superellipsoïde, Bardinet lui ajoute une déformation de type FFD (Free Form Deformation [SP86]) (Figure 1.5).

Plus récemment, Zhou [ZK01] utilise des paramètres de courbure qui sont fonctions des variables paramétriques (Figure 1.6).

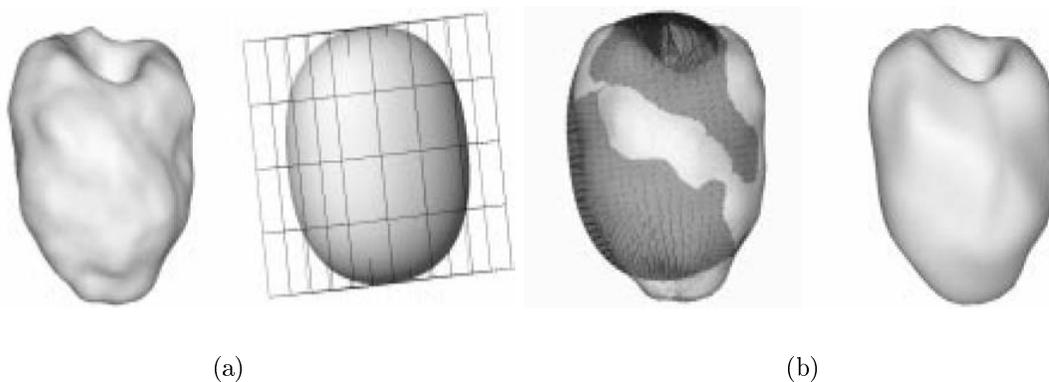


FIG. 1.5 – Approximation de données par superellipsoïde par une FFD [BCA95]. (a) l’objet original et la première approximation par une simple superellipsoïde dans la grille de contrôle de la FFD (b) le calcul du champ de déplacement et le modèle final

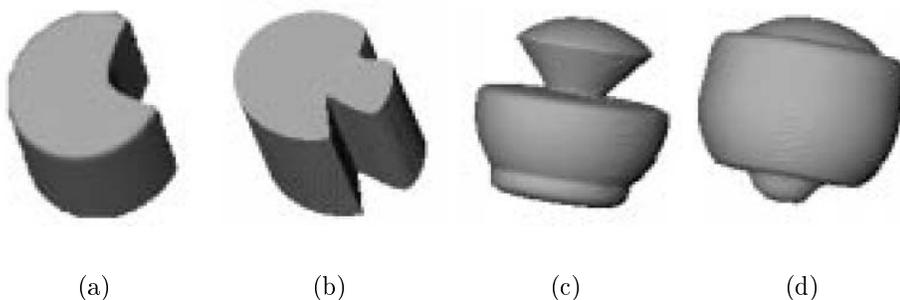


FIG. 1.6 – Exemples de superellipsoïdes étendues par Zhou [ZK01]

On peut aussi noter l’existence des hyperquadriques qui sont une extension des superquadriques (ou plutôt une généralisation). Leur équation générale est de la forme :

$$\sum_i S_i + e^{\sum_j S_j}$$

où  $S$  est l’équation d’une superquadrique.

Ce modèle devient assez délicat à manipuler et l’idée assez intuitive dans les superquadriques de trois paramètres pour la taille et deux pour la courbure est assez loin. De plus, il n’existe plus de forme paramétrique dans le cas des hyperquadriques qui perdent alors beaucoup de leur intérêt. Néanmoins, on peut noter que Cohen [CC94] propose une méthode d’approximation de données 2D et 3D par des hyperquadriques.

Une seconde approche pour augmenter les possibilités descriptives des superquadriques est, comme pour les autres surfaces implicites, de les utiliser en tant que primitives élémentaires et de les combiner afin de modéliser des objets plus complexes. Cette approche est plus particulièrement utilisée dans le cadre de la reconnaissance d’objets 3D (cf. cha-

pitre 5). L'initiateur de cette tendance fût Pentland [PD90]. Gupta [GB93] et Leonardis [LSM94, LJS97] segmentent et modélisent des images de profondeur avec des superellipsoïdes. Ce dernier utilise un algorithme de type croissance de régions dont on aura l'occasion d'étudier les détails dans la suite de ce mémoire (chapitre 4).

## 1.5 Conclusion

Les différents modèles de représentations détaillés, les maillages, les surfaces paramétriques ou implicites, ont leur domaine de prédilection. C'est pour cette raison, que tous existent et continuent d'être utilisés dans des domaines assez différents.

En ce qui concerne notre étude, les fonctions implicites semblent très intéressantes. Leur forte compacité et leur modélisation à base de volumes élémentaires combinés offrant intuitivement un niveau sémantique supplémentaire à la description sont tout à fait ce que nous recherchons. Malheureusement, leur affichage est très pénalisant pour certaines applications. Ainsi, les superellipsoïdes ont une des caractéristiques, outre le faible nombre de paramètres, est de posséder à la fois une forme paramétrique et une forme implicite, semble être toutes désignées. D'autant plus que si leurs capacités de description, déjà assez importantes, venaient à devenir insuffisantes, les méthodes permettant de les déformer ne manquent pas.

Dans le prochain chapitre, nous étudierons de manière détaillée les superellipsoïdes dans l'optique de notre prochaine problématique, l'approximation et la segmentation de données tridimensionnelles.



# Chapitre 2

## Superellipsoïdes

Comme nous venons de le voir dans le chapitre précédent, les superquadriques sont une extension de la famille des quadriques et permettent de représenter un éventail assez large de formes élémentaires tout en étant particulièrement compactes en terme de nombre de paramètres. Comme les quadriques, les superquadriques sont divisées en quatre classes. La superellipsoïde est la classe la plus largement utilisée car elle permet le plus naturellement, le plus souvent par assemblage, de modéliser des objets tridimensionnels.

Dans ce chapitre, nous allons étudier en détail ce modèle de surface. Nous comprendrons notamment en étudiant la construction d'une superellipsoïde à partir de primitives 2D quelles sont exactement les formes qu'elle peut décrire. La paramétrisation de la surface est aussi abordée. Enfin, différentes méthodes permettant la mesure de la distance d'un point de l'espace 3D à une superellipsoïde sont présentées.

### 2.1 De la superellipse à la superellipsoïde

Les quadriques résultent du produit sphérique de deux coniques. Les superquadriques sont obtenues de manière similaire. Ainsi, les superellipsoïdes peuvent être définies comme le produit sphérique de deux superellipses [Bar81].

Les superellipses sont un modèle de courbe 2D introduit en 1960 par le designer danois Piet Hein qui avait besoin de formes “entre le carré et le cercle” (figure 2.1). Elles sont une extension des ellipses, et peuvent être définies par la formule paramétrique<sup>1</sup> :

$$s(\eta) = \begin{cases} x(\eta) = a_1 \cos^{\epsilon_1}(\eta) \\ y(\eta) = a_2 \sin^{\epsilon_1}(\eta) \end{cases}, -\pi \leq \eta \leq \pi$$

---

<sup>1</sup>Pour cette équation et le reste du rapport  $\cos^\epsilon(\eta)$  est une notation simplifiée, il faut comprendre  $\text{signe}(\cos(\eta)) * |\cos(\eta)|^\epsilon$ . Et pareillement pour  $\sin^\epsilon(\eta)$

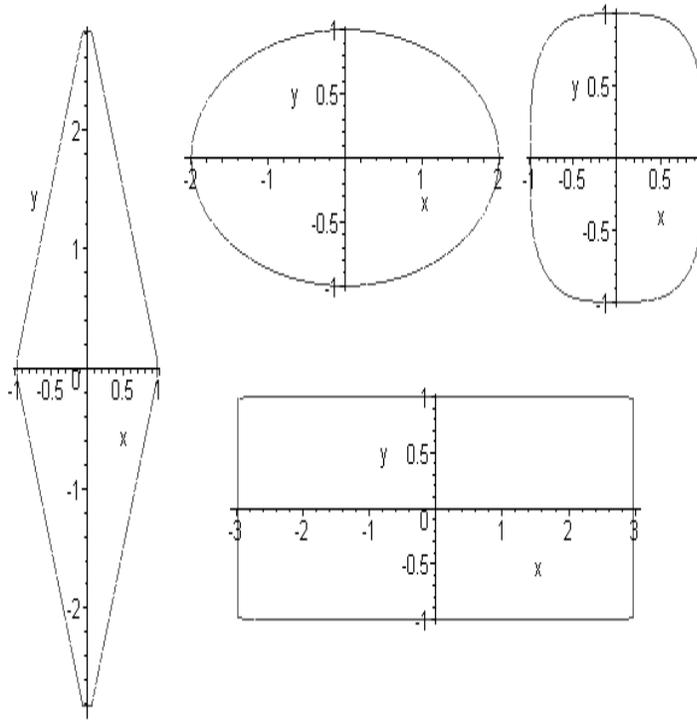


FIG. 2.1 – Exemples de superellipses

de laquelle on peut déduire une formulation implicite<sup>2</sup> :

$$\begin{cases} x(\eta) &= a_1 \cos^\epsilon(\eta) \\ y(\eta) &= a_2 \sin^\epsilon(\eta) \end{cases}$$

$$\begin{cases} x^2(\eta) &= a_1^2 \cos^{2\epsilon}(\eta) \\ y^2(\eta) &= a_2^2 \sin^{2\epsilon}(\eta) \end{cases}$$

$$\begin{cases} \left(\frac{x(\eta)}{a_1}\right)^2 &= \cos^{2\epsilon}(\eta) \\ \left(\frac{y(\eta)}{a_2}\right)^2 &= \sin^{2\epsilon}(\eta) \end{cases}$$

$$\begin{cases} \left(\frac{x(\eta)}{a_1}\right)^{\frac{2}{\epsilon}} &= \cos^2(\eta) \\ \left(\frac{y(\eta)}{a_2}\right)^{\frac{2}{\epsilon}} &= \sin^2(\eta) \end{cases}$$

en utilisant  $\cos^2 \alpha + \sin^2 \alpha = 1$ , on a :

---

<sup>2</sup>Pour cette équation et le reste du rapport  $x^{\frac{2}{\epsilon}}$  est une notation simplifiée, il faut comprendre  $(x^2)^{\frac{1}{\epsilon}}$

$$\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon}} = 1$$

On arrive à reconnaître aisément qu'il s'agit de l'équation d'une ellipse standard légèrement modifiée. Dans ces deux équations, les paramètres  $a_1$  et  $a_2$  définissent, de la même façon que pour les ellipses, la taille sur chacun des deux axes de la forme, tandis que le paramètre supplémentaire  $\epsilon$ , strictement positif, permet de contrôler sa courbure. Si  $\epsilon = 1$  l'équation redevient celle d'une simple ellipse, si  $\epsilon$  est proche de 0, la forme obtenue sera proche d'un rectangle, si  $\epsilon = 2$  la forme sera proche d'un losange. La courbe obtenue sera convexe si et seulement si  $\epsilon \leq 2$  (figure 2.2).

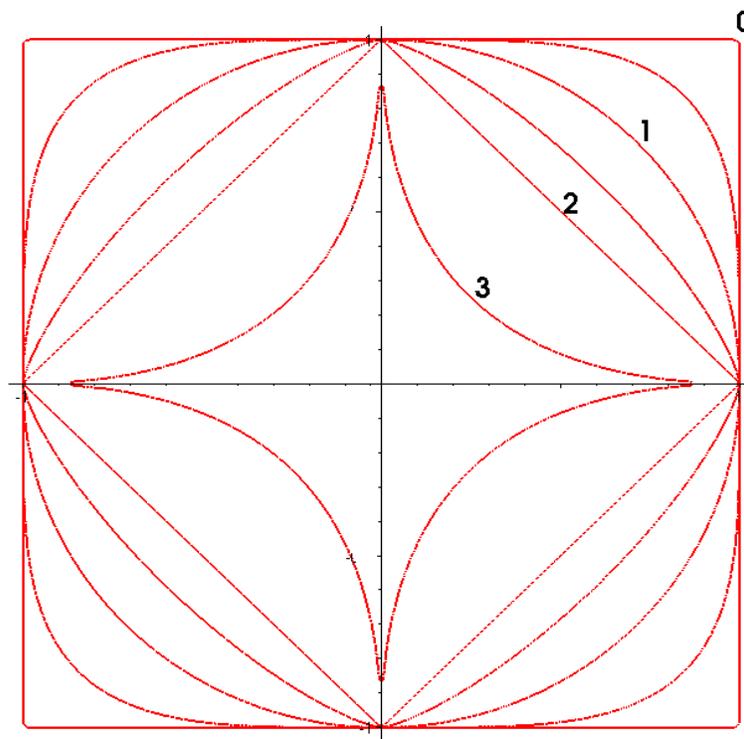


FIG. 2.2 – Effet du paramètre  $\epsilon$  sur la géométrie de la superellipse.

On peut remarquer qu'à une rotation près, plusieurs triplets de paramètres peuvent donner la même forme. Le même carré peut être obtenu avec  $a_1 = a_2 = 1$  et  $\epsilon$  proche de 0, et avec  $a_1 = a_2 = \sqrt{2}$  et  $\epsilon = 2$ . On pourra faire la même remarque pour les superellipsoïdes.

Le produit sphérique par lequel on obtient les superellipsoïdes est défini ainsi [Bar81].

Soit deux courbes 2D  $u(\eta)$  et  $v(\mu)$  :

$$u(\eta) = \begin{bmatrix} u_1(\eta) \\ u_2(\eta) \end{bmatrix} \text{ avec } -\eta_{min} \leq \eta \leq \eta_{min}$$

$$v(\mu) = \begin{bmatrix} v_1(\mu) \\ v_2(\mu) \end{bmatrix} \text{ avec } -\mu_{min} \leq \mu \leq \mu_{min}$$

leur produit sphérique est :

$$p(\eta, \mu) = u(\eta) \otimes v(\mu) = \begin{bmatrix} u_1(\eta) \\ u_2(\eta) \end{bmatrix} \otimes \begin{bmatrix} v_1(\mu) \\ v_2(\mu) \end{bmatrix} = \begin{bmatrix} u_1(\eta)v_1(\mu) \\ u_1(\eta)v_2(\mu) \\ u_2(\eta) \end{bmatrix}, \quad \begin{array}{l} -\eta_{min} \leq \eta \leq \eta_{min} \\ -\mu_{min} \leq \mu \leq \mu_{min} \end{array}$$

On remarque que la participation des deux courbes au produit n'est pas la même. Le produit sphérique n'est pas commutatif. Intuitivement, la courbe  $m(\eta)$  définit la forme des courbes de latitude de la surface tandis que  $n(\mu)$  définit celle des courbes de longitude.

On peut donc déterminer la forme paramétrique d'une superellipsoïde comme le produit de deux superellipses :

$$S(\eta, \mu) = \begin{bmatrix} \cos^{\epsilon_1}(\eta) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix} \otimes \begin{bmatrix} a_1 \cos^{\epsilon_2}(\mu) \\ a_2 \sin^{\epsilon_2}(\mu) \end{bmatrix} = \begin{bmatrix} a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\mu) \\ a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\mu) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix}, \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \mu \leq \pi \end{array} \quad (2.1)$$

Les paramètres  $a_1$ ,  $a_2$  et  $a_3$  définissent la taille (facteur d'échelle) de la superellipsoïde suivant les axes  $x$ ,  $y$  et  $z$  respectivement, tandis que les paramètres  $\epsilon_1$  et  $\epsilon_2$  caractérisent, respectivement, la courbure latitudinale et longitudinale de la forme. Les paramètres  $\epsilon_1$  et  $\epsilon_2$ , n'ont donc pas une influence équivalente sur la forme de la superellipsoïde. Par exemple, un "cylindre" s'obtient avec  $\epsilon_1$  proche de 0 et  $\epsilon_2 = 1$ , mais pas l'inverse.

Grâce à ces 5 paramètres, les superellipsoïdes vont nous permettre de modéliser un grand nombre de formes élémentaires de toutes tailles, allant de la simple ellipsoïde au parallélépipède rectangle en passant par le cylindre ou l'octoèdre et bien sûr toutes les formes intermédiaires (figure 2.4).

Comme évoqué dans le chapitre précédent et tout comme les superellipses, on peut exprimer les superellipsoïdes sous forme implicite. Le passage de la forme paramétrique à implicite se fait de même manière que pour les superellipses. On élimine d'abord  $\mu$  puis  $\eta$ . La surface de la superellipsoïde est l'ensemble des points solution de :

$$F(x, y, z) = \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left( \frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1 \quad (2.2)$$

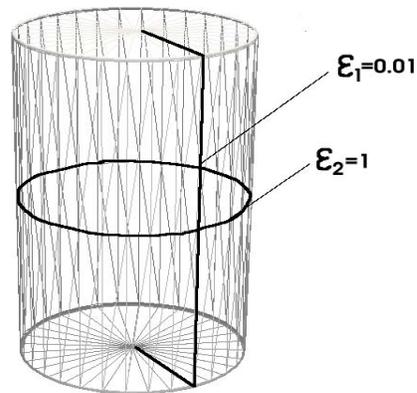


FIG. 2.3 – La superellipsoïde proche du cylindre résulte du produit sphérique d’une superellipse proche du rectangle avec une autre définissant un cercle.

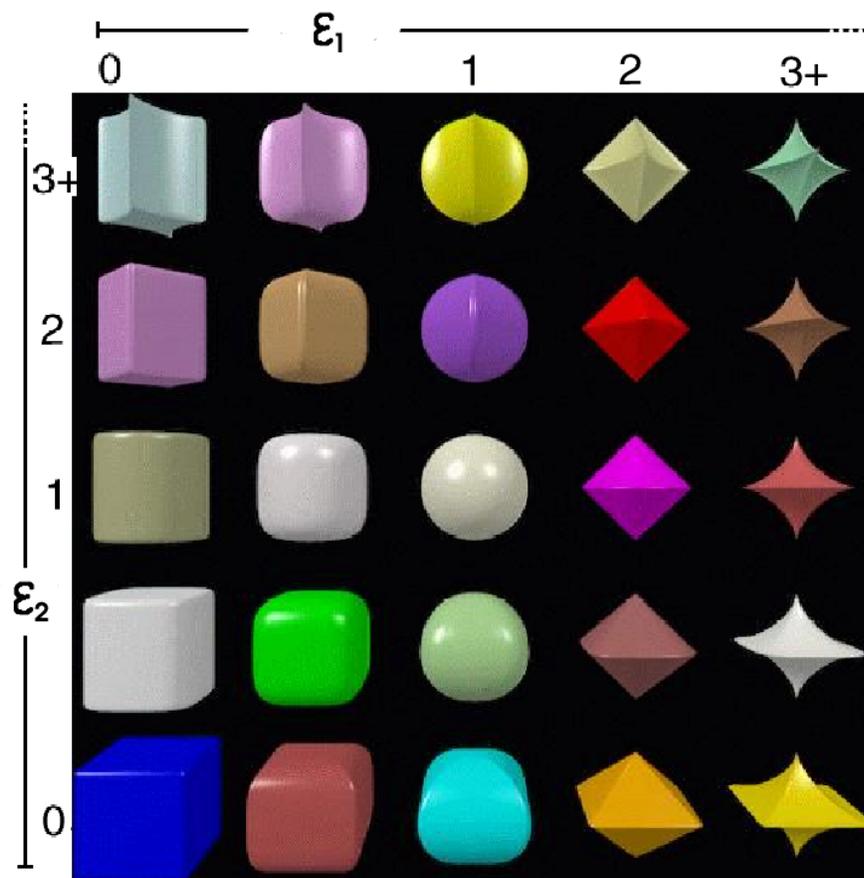


FIG. 2.4 – Exemples de superellipsoïdes

Nous avons expliqué dans la partie sur les modèles implicites<sup>3</sup> que leur formule per-

<sup>3</sup>Nous avons alors défini les surfaces implicites comme solution d’une équation du type  $f(x, y, z) = 0$ .

mettait de partitionner très simplement l'espace tridimensionnel en trois lieux. Les superellipsoïdes ne dérogent pas à la règle. On rappelle donc que pour tout point  $P$  de coordonnées  $(x, y, z)$ , on aura :

- $F(x, y, z) = 1$ , si  $P$  est sur la surface de la superellipsoïde ;
- $F(x, y, z) > 1$ , si  $P$  est à l'extérieur de la surface de la superellipsoïde ;
- $F(x, y, z) < 1$ , si  $P$  est à l'intérieur de la surface de la superellipsoïde.

La particularité des superellipsoïdes à pouvoir être définies à la fois sous forme paramétrique et implicite est intéressante et c'est une des raisons majeures de notre choix. En effet, la forme implicite permet d'obtenir à moindre coût (en terme de calculs) des informations sur la position d'un point par rapport à la surface, ce qui nous sera utile pour l'approximation. La forme paramétrique, quant à elle, permet un échantillonnage immédiat de la surface autorisant ainsi un affichage rapide, ce qui est loin d'être le cas avec une forme implicite pure.

Comme pour les superellipses, plusieurs paramétrages différents permettent d'obtenir sensiblement la même surface. En premier lieu, parce que les superellipsoïdes contiennent des axes de symétrie les rendant invariant par rotation. Par exemple, les paramètres  $a_1$  et  $a_2$  peuvent être échangés sans modifier la forme de la surface (à une rotation près). Mais aussi parce que, comme pour les superellipses (puisqu'elles en dérivent), on pourra obtenir un cube avec deux paramétrages, en utilisant pour la courbe définissant la latitude, soit la première méthode pour obtenir un carré avec une superellipse, soit la seconde. Cela ne pose pas vraiment de problème lorsque les superellipsoïdes sont employées dans un but d'approximation d'objets ; mais cela doit être pris en compte s'il on veut comparer des superellipsoïdes entre-elles en se basant sur leurs paramètres, pour des applications de reconnaissance de formes, par exemple.

## 2.2 Superellipsoïdes : équations générales

Les superellipsoïdes définies dans les équations (2.1) et (2.2) sont centrées à l'origine d'un repère tridimensionnel et leurs axes coïncident avec les axes de ce même repère. Afin de pouvoir décrire des superellipsoïdes dans n'importe quelle position, il est nécessaire d'ajouter une matrice de rotation  $R$  et un vecteur de translation  $\mathbf{t}$  habituellement

---

Pour retrouver cette formulation ici, il suffit, bien sûr de poser  $F'(x, y, z) = F(x, y, z) - 1 = 0$ .

représentés ensemble dans la matrice  $4D$  de transformation  $T$  :

$$T = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & R & \cdot & \mathbf{t} \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos \phi \cos \theta \cos \psi - \sin \phi \sin \psi & -\cos \phi \cos \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \sin \theta & t_x \\ \sin \phi \cos \theta \cos \psi + \cos \phi \sin \theta & -\sin \phi \cos \theta \sin \psi + \cos \phi \cos \theta & \sin \phi \sin \theta & t_y \\ -\sin \theta \cos \psi & \sin \theta \sin \psi & \cos \theta & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

où  $t_x$ ,  $t_y$  et  $t_z$  représentent les translation suivant les axes  $x$ ,  $y$  et  $z$  respectivement, et où  $\phi$ ,  $\theta$ , et  $\psi$  sont les angles de rotation suivant les axes  $x$ ,  $y$  et  $z$  respectivement.

La forme implicite d'une superellipsoïde dans une position générale est alors définie par :

$$\hat{F}(x, y, z; a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_x, t_y, t_z, \phi, \theta, \psi) = F \left( R^\top * \begin{pmatrix} x - t_x \\ y - t_y \\ z - t_z \end{pmatrix} \right) \quad (2.3)$$

sa forme paramétrique générale :

$$\hat{S}(\eta, \mu; a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_x, t_y, t_z, \phi, \theta, \psi) = R * S(\eta, \mu) + t \quad (2.4)$$

Une superellipsoïde dans une position générale peut donc être représentée par 11 paramètres : trois de facteurs d'échelle, deux paramètres permettant de définir sa courbure, trois de rotation et trois de translation.

## 2.3 Paramétrisation régulière d'une superellipsoïde

La forme paramétrique de la superellipsoïde, comme on l'a définie par le produit sphérique de deux superellipses, a la particularité de ne pas produire un échantillonnage régulier, mais au contraire de concentrer les points dans les zones de fortes courbures. Cette paramétrisation est donc très intéressante pour les applications de type affichage puisque plus précise là où il y en a le plus besoin. Néanmoins, d'autres applications peuvent nécessiter une répartition uniforme des points de l'échantillon à la surface de la superellipsoïde (Fig. 2.5). Bardinet propose une méthode de paramétrisation permettant de garder quelle que soit la valeur des coefficients de courbure  $\epsilon_1$  et  $\epsilon_2$ , une répartition régulière des courbes latitudinales et longitudinales [BCA95].

Remarquant que l'échantillon de points sur la surface d'une sphère obtenu grâce à sa forme paramétrique standard est régulièrement réparti, l'idée est de projeter ces points

sur la superellipsoïde. Bardinnet obtient alors une paramétrisation de la superellipsoïde permettant un échantillonnage bien plus régulier :

$$S'(\eta, \mu) = \begin{bmatrix} a_1 \rho(\eta, \mu) \cos(\eta) \cos(\mu) \\ a_2 \rho(\eta, \mu) \cos(\eta) \sin(\mu) \\ a_3 \rho(\eta, \mu) \sin(\eta) \end{bmatrix}, \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \mu \leq \pi \end{array}$$

avec :

$$\rho(\eta, \mu) = \left( \left( |\cos(\mu) \cos(\eta)|^{\frac{2}{\epsilon_2}} + |\sin(\mu) \cos(\eta)|^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + |\sin(\eta)|^{\frac{2}{\epsilon_1}} \right)^{-\frac{\epsilon_1}{\epsilon_2}}$$

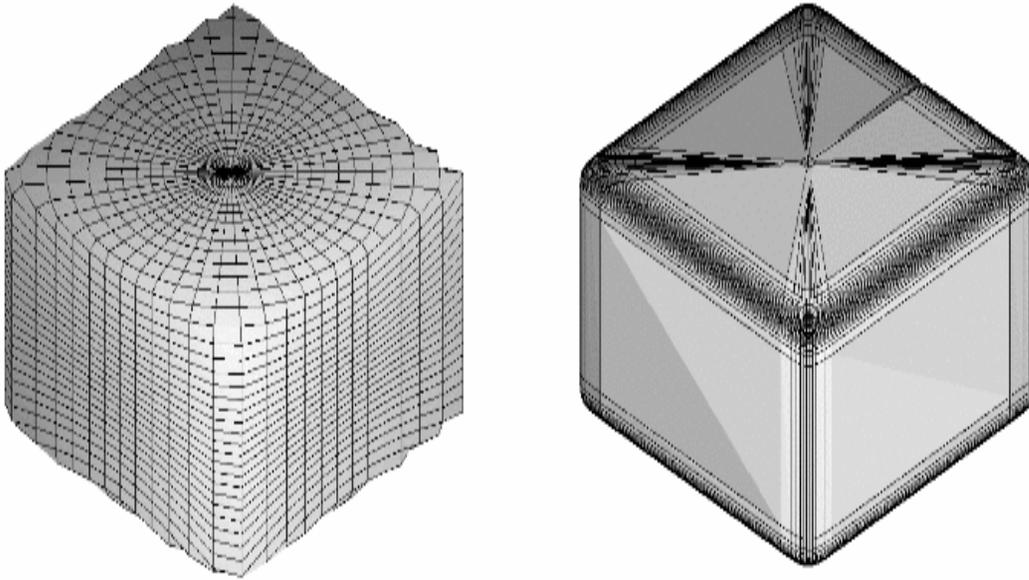


FIG. 2.5 – Paramétrisation régulière et paramétrisation standard.

Précisons que, comme le montre la figure 2.5, l'échantillonnage de la superquadrique n'est pas complètement régulier si on considère l'intersection des lignes latitudinales et longitudinales. Il y a une concentration d'intersections et donc de points au niveau des pôles. Mais l'espacement entre ces lignes, lui, est régulier.

## 2.4 Vecteur normal

La connaissance du vecteur normal en tout point à une surface est très utile en synthèse d'image. Il peut aussi l'être en approximation pour calculer la distance d'un point à la surface, par exemple. Nous allons voir que l'expression du vecteur normal à une superellipsoïde possède une forme remarquable et peut-être calculé très simplement [Bar81, JLS00].

Le vecteur normal à une surface en un point est défini par le produit vectoriel des tangentes en ce point :

$$\begin{aligned}
N(\eta, \mu) &= \frac{\delta S_\eta(\eta, \mu)}{\delta \eta} \wedge \frac{S_\mu(\eta, \mu)}{\delta \eta} \\
&= \begin{bmatrix} -a_1 \epsilon_1 \sin(\eta) \cos^{\epsilon_1-1}(\eta) \cos^{\epsilon_2}(\mu) \\ -a_2 \epsilon_1 \sin(\eta) \cos^{\epsilon_1-1}(\eta) \sin^{\epsilon_2}(\mu) \\ a_3 \epsilon_1 \cos(\eta) \sin^{\epsilon_1-1}(\eta) \end{bmatrix} \wedge \begin{bmatrix} -a_1 \epsilon_2 \sin(\mu) \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2-1}(\mu) \\ a_2 \epsilon_2 \cos(\mu) \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2-1}(\mu) \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} -a_2 a_3 \epsilon_1 \epsilon_2 \cos(\mu) \sin^{\epsilon_1-1}(\eta) \cos^{\epsilon_1+1}(\eta) \sin^{\epsilon_2-1}(\mu) \\ -a_1 a_3 \epsilon_1 \epsilon_2 \sin(\mu) \sin^{\epsilon_1-1}(\eta) \cos^{\epsilon_1+1}(\eta) \cos^{\epsilon_2-1}(\mu) \\ -a_1 a_2 \epsilon_1 \epsilon_2 \sin(\eta) \cos^{2\epsilon_1-1}(\eta) \sin^{\epsilon_2-1}(\mu) \cos^{\epsilon_2-1}(\mu) \end{bmatrix}
\end{aligned}$$

Ainsi, la direction du vecteur normal est de forme remarquable, puisque si on simplifie le résultat par le terme commun :

$$f(\eta, \mu) = -a_1 a_2 a_3 \epsilon_1 \epsilon_2 \sin^{\epsilon_1-1}(\eta) \cos^{2\epsilon_1-1}(\eta) \sin^{\epsilon_2-1}(\mu) \cos^{\epsilon_1-1}(\mu)$$

on obtient :

$$\begin{cases} N'_x(\eta, \mu) = \frac{1}{a_1} \cos^{2-\epsilon_1}(\eta) \cos^{2-\epsilon_2}(\mu) \\ N'_y(\eta, \mu) = \frac{1}{a_2} \cos^{2-\epsilon_1}(\eta) \sin^{2-\epsilon_2}(\mu) \\ N'_z(\eta, \mu) = \frac{1}{a_3} \sin^{2-\epsilon_1}(\eta) \end{cases}, \quad \begin{matrix} \eta \in [-\frac{\pi}{2}, \frac{\pi}{2}], \\ \mu \in [-\pi, \pi] \end{matrix}$$

et pour  $\epsilon_1$  et  $\epsilon_2$  inférieurs à 2, l'équation des normales définit une autre superellipsoïde (sa duale) de paramètres de courbure  $\epsilon'_1 = 2 - \epsilon_1$  et  $\epsilon'_2 = 2 - \epsilon_2$  et de dimension  $a'_1 = \frac{1}{a_1}$ ,  $a'_2 = \frac{1}{a_2}$  et  $a'_3 = \frac{1}{a_3}$ .

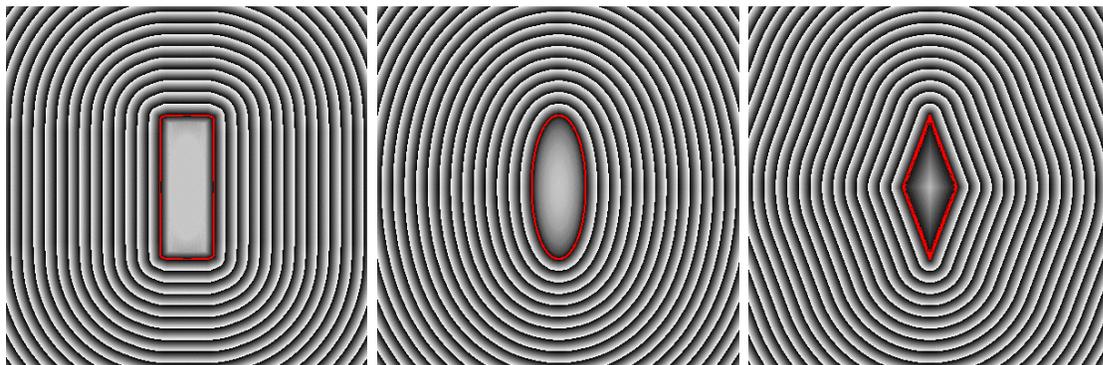
## 2.5 Distance d'un point à la surface d'une superellipsoïde

La capacité de pouvoir relativement facilement exprimer la distance d'un point tridimensionnel à la surface d'une superellipsoïde est une condition nécessaire pour l'application de modélisation de données à laquelle nous destinons la superellipsoïde. En effet, c'est grâce à cette distance que nous pourrions juger de la qualité d'une approximation.

Il n'est pas envisageable en terme de temps de calcul de calculer la distance euclidienne d'un point à la surface d'une superellipsoïde. Son évaluation nécessite un procédé de régression itératif. Nous devons donc nous contenter d'une approximation de cette distance.

Dans cette partie, nous étudierons différentes approches d'approximation de la distance d'un point à une superellipsoïde. Pour chacune d'entre-elles, ont été tracées les courbes d'équidistance de coupes de différentes superellipsoïdes dans le but de leur donner un côté plus visuel. Pour obtenir ces champs de distances, nous avons calculé la distance de chaque pixel à la surface de la superellipsoïde. Les pixels rouges sont ceux que l'on considère sur la surface (distance du pixel à la surface négligeable). Le niveau de gris de chaque pixel est proportionnel à la distance de ce même pixel à la surface modulo une constante qui dépend de l'approximation de la distance considérée. Ce modulo explique que le même niveau de gris revient périodiquement lorsqu'on s'éloigne de la surface.

Afin de pouvoir comparer ces courbes avec la distance euclidienne, ou ce à quoi ces courbes devraient tendre, nous avons tracé une approximation de cette dernière (figure 2.6) que l'on a approchée en calculant la distance minimum existant entre le point et un triangle de la surface préalablement triangulée (une méthode très gourmande donc). Bien entendu, la même échelle a été utilisée pour toutes ces courbes.



(a) Cas d'un rectangle

(b) Cas d'une ellipsoïde

(c) Cas d'un octoèdre

FIG. 2.6 – Distance euclidienne à une superellipsoïde

Dans toute cette partie, nous considérerons que nous essayons de déterminer la distance d'un point  $P = (x_1, y_1, z_1)$  par rapport à une superellipsoïde  $S$  dans son repère intrinsèque de centre  $O$ . On nommera  $P'$  la projection orthogonale de  $P$  sur  $S$ . La distance euclidienne est donc  $\|\overrightarrow{PP'}\|$ . Soit  $P'' = (x_2, y_2, z_2)$  l'intersection entre la droite  $OP$  et la surface (figure 2.7).

## Fonction potentiel

Beaucoup d'approximations de la distance se basent sur la fonction potentiel de la superellipsoïde. Il est vrai que cela est à la fois simple et peu coûteux.

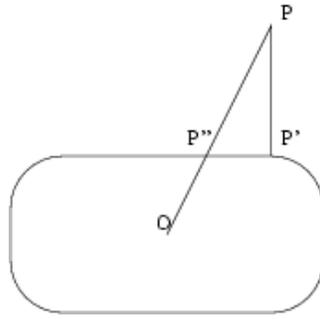
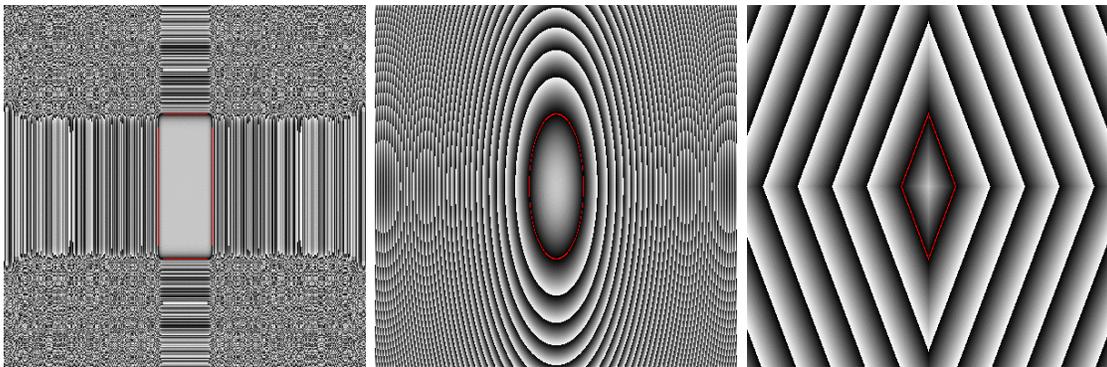


FIG. 2.7 – Notations. On utilisera dans ce chapitre  $O$  comme le centre de la superellipsoïde  $S$ ,  $P$  le point dont on désire connaître la distance à la surface,  $P'$  sa projection orthogonale sur  $S$  et  $P''$  l'intersection entre la droite  $OP$  et la surface  $S$

Mis à part le fait que cette fonction a la propriété d'augmenter quand on s'éloigne du centre de la superellipsoïde dans une direction, la figure 2.8 montre clairement qu'elle est loin de la distance euclidienne.



(a) Cas d'un rectangle

(b) Cas d'une ellipse

(c) Cas d'un octoèdre

FIG. 2.8 – Fonction potentiel d'une superellipsoïde

Si l'on désire utiliser cette fonction potentiel pour estimer la distance d'un point à la surface, plusieurs problèmes sont à résoudre. Le premier problème est que les valeurs de ce potentiel pour un point à même distance euclidienne varient énormément suivant les valeurs des coefficients de courbure. Le potentiel d'un point à une distance donnée d'une ellipse, est beaucoup plus élevé que le potentiel de ce même point à un parallélépipède de même taille et plus faible que le potentiel à un octoèdre. Si l'on se place dans le contexte de l'utilisation d'une telle approximation de la distance dans le but d'approximation, les superellipsoïdes de type octoèdres (figure 2.8c) seront donc anormalement favorisées. Comme l'a déjà fait remarquer Solina [SB90], dans le cas d'un  $\epsilon_1$  petit, le potentiel suivant

l'axe des  $z$  va croître très rapidement.

Deux autres problèmes sont bien visibles dans le cas d'une ellipsoïde (figure 2.8b). D'une part, le potentiel ne croît pas linéairement. Ce qui n'est évidemment pas le cas de la distance euclidienne. D'autre part, la croissance du potentiel n'est pas la même dans toutes les directions. Un point dans le "sillage" de l'axe principal d'inertie aura un potentiel plus faible que celui, à même distance euclidienne, qui se trouve dans la direction d'un des deux autres axes. Ce phénomène sera d'autant plus fort, que le rapport entre la taille de l'axe principal d'inertie et cet autre axe sera grand.

Il faudra donc résoudre au maximum ces problèmes afin de pouvoir utiliser le potentiel pour approximer la distance d'un point à la surface.

## Fonction standard d'approximation de distance

Solina propose, en 1990, une méthode pour approximer les ensembles de points par une superellipsoïde [SB90]. Pour juger de la distance d'un point à la surface, il utilise, tout d'abord, le critère d'erreur suivant :

$$f(x, y, z)^{\epsilon_1} - 1 \tag{2.5}$$

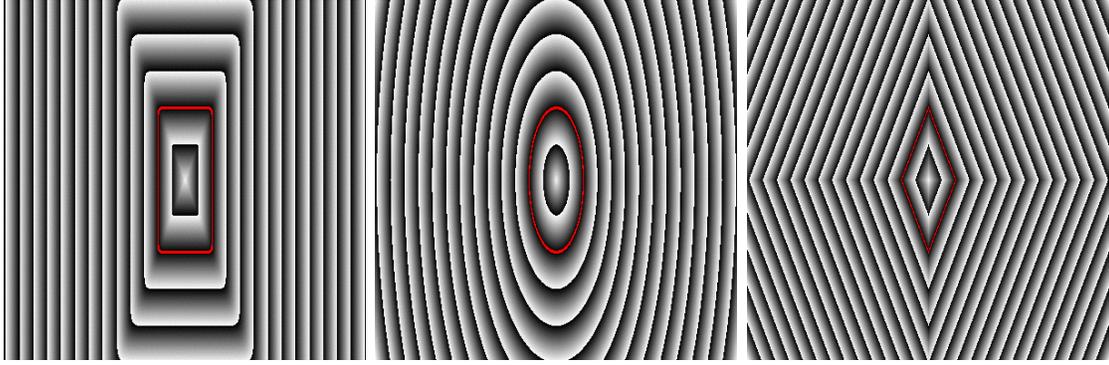
Par la suite, cette fonction sera légèrement modifiée et deviendra l'approximation de la distance la plus généralement utilisée dans le cas d'approximations :

$$F_1(x, y, z) = f(x, y, z)^{\frac{\epsilon_1}{2}} - 1 \tag{2.6}$$

Ce critère d'erreur supprime le problème lié au  $\epsilon_1$  petit. De plus, sa valeur semble la même pour une distance donnée quels que soient les valeurs des coefficients de courbure (figure 2.9). Il a une valeur qui croît de façon linéaire quand on s'éloigne du centre dans une direction donnée.

Le seul problème persistant est qu'il ne croît pas à la même vitesse dans toutes les directions. Deux points à même distance euclidienne de la surface, n'auront donc pas forcément la même valeur.

On remarque visuellement sur la figure 2.9, que les courbes décrivent la même superellipse à un facteur d'échelle près (on en aura la preuve mathématique dans le paragraphe suivant). L'erreur d'approximation de la distance sera donc d'autant plus grande que le rapport des paramètres de tailles sur les trois axes sera différent de 1. Car en effet, dans le cas d'une distance euclidienne, les courbes tendent toutes vers le cercle ; c'est à dire que les rapports entre les paramètres de taille et de courbure devraient peu à peu tendre vers 1 et non rester constants.



(a) Cas d'un rectangle

(b) Cas d'une ellipsoïde

(c) Cas d'un octoèdre

FIG. 2.9 – Critère d'erreur Solina :  $F_1$ 

### Distance “euclidienne radiale”

Pour corriger ces problèmes, d'autres méthodes peuvent être utilisées. La distance “euclidienne radiale” est la distance du point  $P$  au point d'intersection entre la surface et la droite passant par  $P$  et le centre de la superellipsoïde, c'est-à-dire  $\|\overrightarrow{P''P}\|$  [BCA95]. Cette distance est égale à la distance euclidienne si  $P'$  et  $P''$  sont confondus, c'est-à-dire si la droite  $(OP)$  est confondue avec l'un des axes de la superellipsoïde.

Puisque  $O$ ,  $P$  et  $P''$  sont alignés,  $\overrightarrow{OP''} = k * \overrightarrow{OP}$ . On a ainsi :

$$\overrightarrow{PP''} = \overrightarrow{OP} - \overrightarrow{OP''} = \overrightarrow{OP} - k * \overrightarrow{OP} = (1 - k) * \overrightarrow{OP}$$

Or,  $O$  et  $P$  sont deux points dont nous avons les coordonnées, donc pour déterminer la distance “euclidienne radiale” ( $\|\overrightarrow{P''P}\|$ ), il suffit de trouver  $k$ .  $O$ ,  $P$  et  $P''$  étant alignés, on a :

$$\begin{aligned} F(P) &= F(k * P'') = \left( \left( \frac{k * x_2}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{k * y_2}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left( \frac{k * z_2}{a_3} \right)^{\frac{2}{\epsilon_1}} \\ &= \left( k^{\frac{2}{\epsilon_2}} \left( \frac{x_2}{a_1} \right)^{\frac{2}{\epsilon_2}} + k^{\frac{2}{\epsilon_2}} \left( \frac{y_2}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + k^{\frac{2}{\epsilon_1}} \left( \frac{z_2}{a_3} \right)^{\frac{2}{\epsilon_1}} = k^{\frac{2}{\epsilon_1}} F(x_2, y_2, z_2) = k^{\frac{2}{\epsilon_1}} F(P'') \end{aligned}$$

On a donc :

$$F^{\frac{\epsilon_1}{2}}(P) = kF(P'')$$

Ce qui est la preuve que l'on avait annoncée dans le paragraphe précédent : les courbes d'équi-distance du critère proposée par Solina ( $F^{\frac{\epsilon_1}{2}}(P)$ ) définissent toutes la même superellipsoïde à un facteur d'échelle près ( $k$ ). Comme  $P''$  est sur la surface et on a donc  $F(P'') = 1$ ,

on en déduit :

$$k = F^{\frac{\epsilon_1}{2}}(P)$$

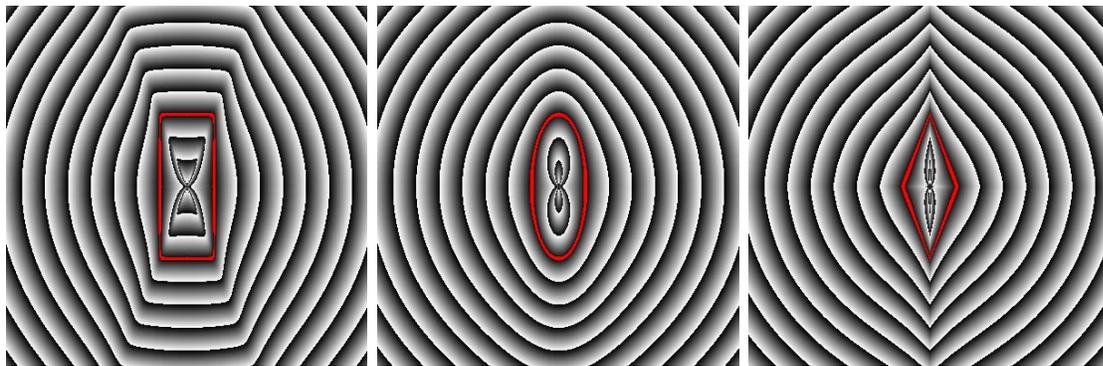
On a donc :

$$\overrightarrow{PP''} = |1 - F^{\frac{\epsilon_1}{2}}(P)| * \overrightarrow{OP} \quad (2.7)$$

En nommant  $F_2$  cette fonction de “distance” :

$$F_2(x_1, y_1, z_1) = \|\overrightarrow{PP''}\| = \|\overrightarrow{OP}\| * |1 - F^{\frac{\epsilon_1}{2}}(P)| \quad (2.8)$$

Cette approximation n’est pas beaucoup plus coûteuse que le critère d’erreur de Solina. Il suffit, en plus, de calculer la distance euclidienne de  $O$  à  $P$  qui sont deux points dont nous connaissons les coordonnées. Elle a pour avantage d’effacer l’effet “dilatation” du potentiel (figure 2.10). Sa croissance est similaire quelle que soit la direction de  $(OP)$ .



(a) Cas d’un rectangle

(b) Cas d’une ellipse

(c) Cas d’un octoèdre

FIG. 2.10 – Distance euclidienne radiale :  $F_2$

On peut aussi remarquer que la formule 2.7 nous permet de démontrer la propriété que l’on avait admis pour la forme implicite :

- $F(x, y, z) = 1$ , si  $P$  est sur la surface de la superellipsoïde ;
- $F(x, y, z) > 1$ , si  $P$  est à l’extérieur de la surface de la superellipsoïde ;
- $F(x, y, z) < 1$ , si  $P$  est à l’intérieur de la surface de la superellipsoïde.

### Critère d’erreur de Boulton et Gross (1987)

Cette approximation de la distance, proposée par Boulton et Gross (  $E_3$  dans [BG87] ), a la particularité de ne pas utiliser la forme implicite de la superellipsoïde. C’est pour cette raison que nous l’exposons ici, mais s’agissant en fait d’une seconde méthode de calcul de la distance euclidienne radiale, elle n’a que peu d’intérêt. Notons que cette approche est antérieure à celle déjà présentée.

Le critère d'erreur de Boulton et Gross est défini ainsi :

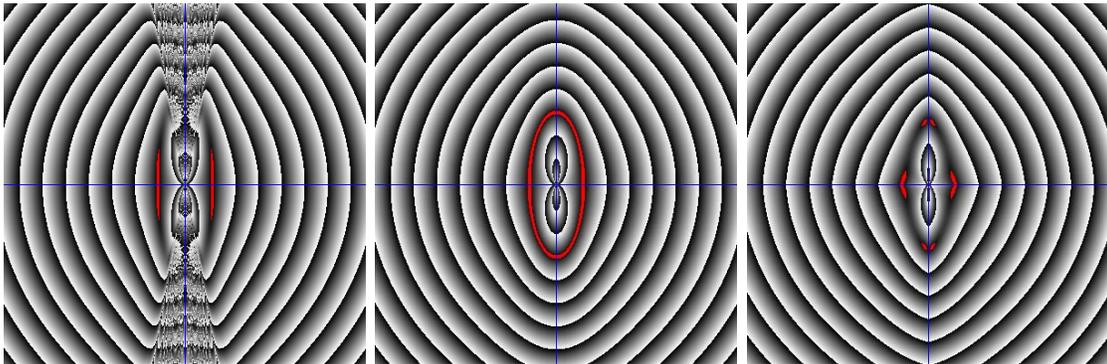
$$F(x, y, z) = \|\overrightarrow{PP''}\| = \sqrt{(P_x - P''_x)^2 + (P_y - P''_y)^2 + (P_z - P''_z)^2} \quad (2.9)$$

L'originalité de la méthode est de calculer les coordonnées de  $P''$  grâce à la formule paramétrique de la superellipsoïde. Pour cela il s'agit en fait de déterminer les angles :  $\eta$  et  $\mu$ . Boulton et Gross les calculent ainsi :

$$\begin{aligned} P'' &= S(\eta, \omega) \\ \omega &= \left( \tan^{-1} \left( \frac{a_1 v_y}{a_2 v_x} \right)^{\frac{1}{\epsilon_2}} \right) \\ \eta &= \left( \tan^{-1} \left( \frac{a_1 v_z}{a_3 v_x \sin^{\epsilon_2}(\omega)} \right)^{\frac{1}{\epsilon_1}} \right) \end{aligned}$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = R^T * \left( \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \right)$$

Le vecteur  $\mathbf{v}$  correspond aux coordonnées de  $P$  rapporté dans le repère intrinsèque à la superellipsoïde. On peut remarquer que les angles  $\eta$  et  $\mu$  qui permettent de déterminer  $P''$  ne peuvent pas être calculés uniquement avec le rapport des distances. Ces angles ne sont pas ceux de la droite  $(OP)$ . Ils sont aussi fonction des coefficients de courbure  $\epsilon_1$  et  $\epsilon_2$ . Cela est dû à la non régularité de l'échantillonnage obtenu avec la forme paramétrique standard qui est utilisée pour obtenir  $P''$ .



(a) Cas d'un rectangle

(b) Cas d'une ellipsoïde

(c) Cas d'un octoèdre

FIG. 2.11 – Distance euclidienne radiale par Boulton et Gross : E3

La figure 2.11 montre que ce critère est effectivement très proche de la distance euclidienne radiale. On peut cependant noter certains problèmes pour les petites valeurs de  $\epsilon_2$

(figure 2.11a). De plus, les axes ressortent sur la figure car le critère n'est pas défini pour ces valeurs (car  $v_x = 0$  ou  $v_y = 0$ ).

Yokoya propose lui aussi une distance n'utilisant pas la forme implicite [YKY92]. Son approche est assez proche de l'approximation de Boulton mais il utilise en plus les normales à la surface et a besoin des normales en chaque point des données pour améliorer la qualité de l'approximation de la distance (la distance euclidienne radiale n'est pas la distance euclidienne d'un point à la surface). Comme nous supposons ne pas avoir d'information sur la normale aux données, nous ne nous attarderons pas sur cette approche ici.

## Distance “euclidienne approchée”

Une approche permettant d'obtenir une approximation plus précise de la distance est d'utiliser la méthode de Taubin [Tau91]. Si le point considéré est proche de la surface, on peut, en effet, utiliser l'approximation au premier ordre en série de Taylor [BCA95, LSM94] :

$$F(P) = F(P') + F'(P') \cdot \overrightarrow{P'P} + O(\|\overrightarrow{P'P}\|^2)$$

d'où

$$\|\overrightarrow{P'P}\| = \frac{|F(P) - 1|}{\|\overrightarrow{\nabla} F(P')\|}$$

Ne connaissant pas  $P'$ , mais le sachant proche de  $P$ , on approxime  $\overrightarrow{\nabla} F(P')$  par  $\overrightarrow{\nabla} F(P)$  :

$$\|\overrightarrow{P'P}\| = \frac{|F(P) - 1|}{\|\overrightarrow{\nabla} F(P)\|}$$

d'où une nouvelle distance  $F_3$  définie par :

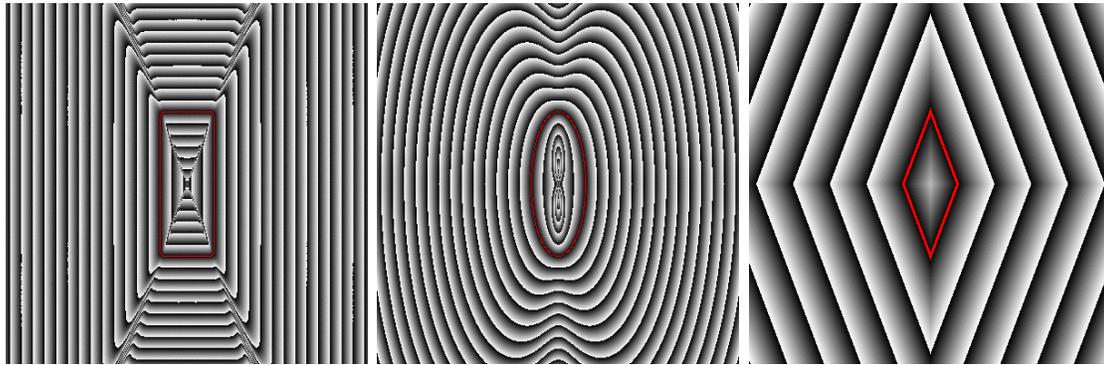
$$F_3(x, y, z) = \frac{|F(x, y, z) - 1|}{\|\overrightarrow{\nabla} F(x, y, z)\|} \quad (2.10)$$

N'oublions pas que cette approximation n'est valable que si  $P$  est suffisamment proche de  $S$ . Dans ce cas, comme le montre la figure 2.12 son comportement est proche de la distance euclidienne dans le cas d'une ellipsoïde et l'effet “dilatation” n'existe plus.

Notons qu'à cause du calcul du gradient, le coût de cette fonction est environ quatre fois plus important que pour les deux précédentes  $F_1$  et  $F_2$  (équations 2.6 et 2.8) .

## Commentaire sur les fonctions de distance :

Nous avons exposé un certain nombre de méthodes permettant d'approcher la distance d'un point à la surface d'une superellipsoïde, il en existe certainement bien d'autres. Parmi



(a) Cas d'un rectangle

(b) Cas d'une ellipse

(c) Cas d'un octoèdre

FIG. 2.12 – Distance euclidienne approchée :  $F_3$ 

ces approximations, bien que celle proposée par Solina soit de loin la plus utilisée, la distance euclidienne radiale est la plus séduisante. Elle paraît, si on s'en tient aux courbes d'équidistance, de qualité semblable à la distance euclidienne approchée pour un coût en temps de calcul bien inférieur. Tout cela dépend bien sûr de l'utilisation que l'on veut faire de cette distance et reste assez théorique. Nous comparerons concrètement ces distances dans la section consacrée à l'approximation d'ensembles de points par une superellipse.

## 2.6 Les autres classes de superquadrriques

On sait que les superellipsoïdes ne sont qu'une classe parmi celles que forment les superquadrriques. Nous allons ici rappeler brièvement les formes paramétriques et implicites des trois autres classes. Les équations sont très proches de celles d'une superellipse. Pour cette raison la plupart des résultats ou remarques concernant les superellipsoïdes évoqués précédemment sont adaptables à l'ensemble des superquadrriques.

### Les superhyperboloïdes à une nappe

Elles résultent du produit d'une superhyperbole et d'une superellipse<sup>4</sup>.

Forme paramétrique :

$$x(\eta, \mu) = \begin{bmatrix} a_1 \sec^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\mu) \\ a_2 \sec^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\mu) \\ a_3 \tan^{\epsilon_1}(\eta) \end{bmatrix}, \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \mu \leq \pi \end{array}$$

Forme implicite :

---

<sup>4</sup>*sec* est la sécante définie comme l'inverse du cosinus  $\sec(\alpha) = \frac{1}{\cos(\alpha)}$

$$F(x, y, z) = \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} - \left( \frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1$$

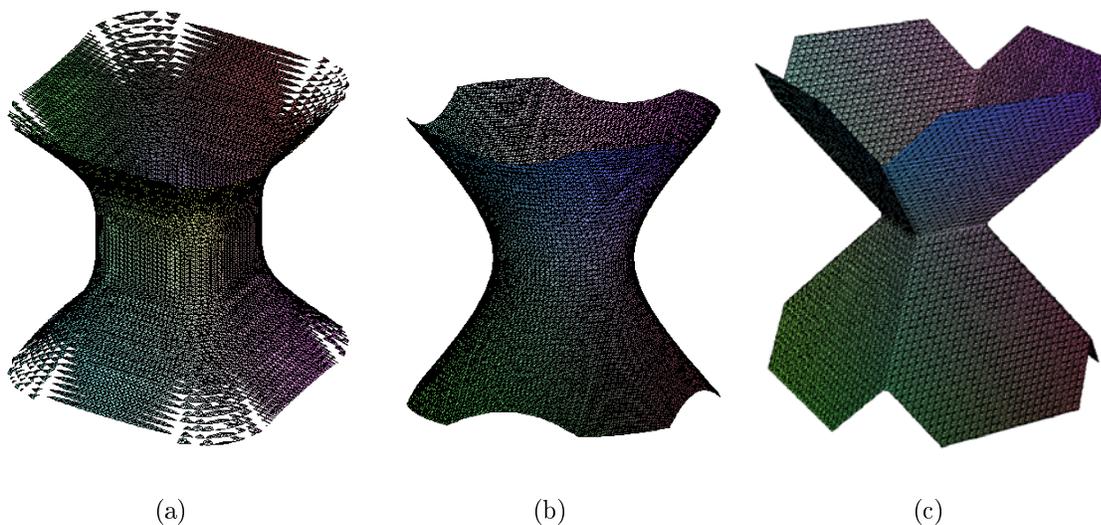


FIG. 2.13 – Exemples de superhyperboloïdes à une nappe

### Les superhyperboloïdes à deux nappes

Elles résultent du produit deux superhyperboles.

Forme paramétrique :

$$x(\eta, \mu) = \begin{bmatrix} a_1 \sec^{\epsilon_1}(\eta) \sec^{\epsilon_2}(\mu) \\ a_2 \sec^{\epsilon_1}(\eta) \tan^{\epsilon_2}(\mu) \\ a_3 \tan^{\epsilon_1}(\eta) \end{bmatrix}, \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \mu \leq \pi \end{array}$$

Forme implicite :

$$F(x, y, z) = \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} - \left( \frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} - \left( \frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1$$

### Les supertoroïdes

Ils résultent du produit d'un supertore et d'une superellipse. Ce sont, avec les superellipsoïdes, les seuls types des superquadriques fermées. C'est pourquoi ce sont les deux

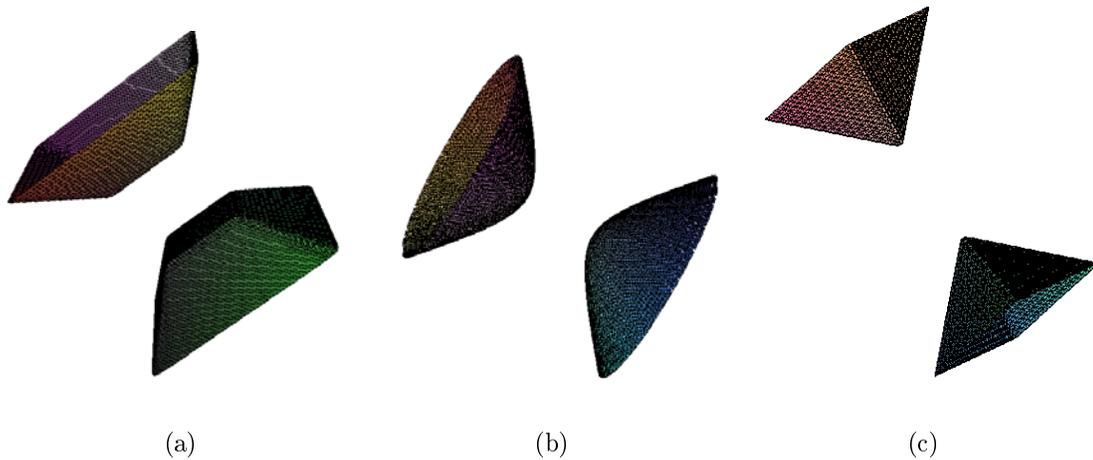


FIG. 2.14 – Exemples de superhyperboloïdes à deux nappes

classes facilement utilisables pour la modélisation. Pour les autres, il est nécessaire de limiter la surface, en contraignant  $\eta$  et  $\mu$  par exemple, ce qui rend les choses un peu plus compliquées.

Forme paramétrique :

$$x(\eta, \mu) = \begin{bmatrix} a_1(a_4 + \cos^{\epsilon_1}(\eta)) \cos^{\epsilon_2}(\mu) \\ a_2(a_4 + \cos^{\epsilon_1}(\eta)) \sin^{\epsilon_2}(\mu) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix}, \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \mu \leq \pi \end{array}$$

Forme implicite ;

$$F(x, y, z) = \left( \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{2}} - a_4 \right)^{\frac{2}{\epsilon_1}} + \left( \frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1$$

Les supertoroïdes sont très proches des superellipsoïdes. Ces dernières sont même un cas particulier de supertoroïde. En effet, dans le cas  $a_4 = 0$ , on retrouve exactement les équations d'une superellipsoïde.

## 2.7 Conclusion

La superellipsoïde constitue un modèle de surface généraliste et compact tout à fait adapté à notre problématique. Elle possède, de plus, certaines propriétés qui rendent son utilisation assez simple. La cohabitation de la forme implicite et de la forme paramétrique facilite à la fois les applications de type approximation et celles de rendu. Elles permettent la représentation de formes assez variées incluant presque toutes les formes basiques :

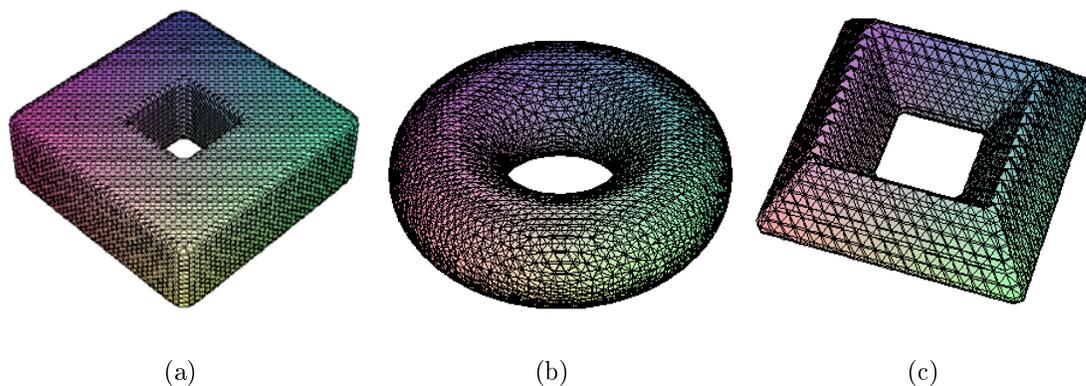


FIG. 2.15 – Exemples de supertoroïdes

cubes, ellipsoïdes, cylindres. Elles pèchent peut-être par une symétrie inévitable et donc l'impossibilité de décrire avec une seule primitive des formes de type pyramides.

Outre les superellipsoïdes, et de la même façon que pour les quadriques, plusieurs classes de superquadriques existent. Elles sont toutes dotées de nombreux atouts et la plupart des résultats exposés ici sont en fait généraux aux quatre classes. De plus, si nous avons décidé d'utiliser les superellipsoïdes comme primitives de notre descripteur, on peut facilement imaginer l'élargissement des méthodes de modélisation aux supertoroïdes qui sont, en fait, une extension des superellipsoïdes avec un paramètre supplémentaire pour contrôler le rayon intérieur.

En ce qui concerne l'approximation de données, que nous allons étudier dans le chapitre suivant, une information quantitative sur la position d'un point à la surface est nécessaire. Si le calcul de la distance euclidienne n'est pas envisageable, il existe plusieurs méthodes peu coûteuses permettant de l'approcher. Parmi les distances que nous avons présentées, et après l'étude approfondie que nous avons faite, la distance euclidienne radiale, bien qu'elle ne soit pas celle la plus communément utilisée semble la plus intéressante. Nous verrons ce qu'il en est, en pratique, dans le chapitre suivant.

# Chapitre 3

## Approximation d'un ensemble de points 3D par une superellipsoïde

La première étape pour obtenir un descripteur de données tridimensionnelles à base de superquadriques, ou plus exactement de superellipsoïdes, est de savoir approximer un ensemble de points 3D par une superellipsoïde. C'est-à-dire, étant donné un ensemble de points, être capable de déterminer l'ensemble des 11 paramètres permettant de définir la superellipsoïde modélisant le mieux les données ; car bien sûr il est très rare qu'un ensemble de points 3D puisse correspondre exactement à la surface d'une seule superellipsoïde.

La méthode que nous utilisons est assez classique. Elle fût proposée par Solina et Bajski [SB90]. Il s'agit d'une approximation au sens des moindres carrés. Nous considérons que la superellipsoïde qui approche le plus vraisemblablement les données est celle dont la somme des distances au carré de chaque point à sa surface est la plus faible. Pour un ensemble de  $N$  points tridimensionnels, il s'agit de trouver le vecteur des 11 paramètres  $\mathbf{a} = [a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_x, t_y, t_z, \phi, \theta, \psi]$  qui minimise :

$$\sum_{i=1}^N d(\mathbf{x}_i, \mathbf{a})^2$$

avec  $d$  la distance d'un point à la superellipsoïde (classiquement issue de l'équation 2.6 mais pouvant aussi provenir de 2.8).

Solina et Bajski travaillaient sur des données de type images de profondeur. Les objets qu'ils approximaient étaient par conséquent incomplets (une seule face visible). Beaucoup de superellipsoïdes peuvent alors correspondre aux données. Pour résoudre ce problème, ils vont introduire le facteur  $(a_1 * a_2 * a_3)^{\frac{1}{3}}$  dans la formule à minimiser privilégiant ainsi les petites superquadriques. Nous serons nous aussi confrontés à des données incomplètes et utiliserons donc ce facteur.

Finalement, le problème à résoudre est donc de trouver  $\mathbf{a}$  minimisant :

$$\Delta(\mathbf{a}) = (a_1 * a_2 * a_3)^{\frac{1}{3}} \sum_{i=1}^N d(\mathbf{x}_i, \mathbf{a})^2 \quad (3.1)$$

La résolution ne peut se faire de manière directe (i.e. non-itérativement, analytiquement). Il est nécessaire d'utiliser un algorithme de régression. La méthode se décompose alors en deux étapes. La première est de trouver un ensemble de paramètres qui sera le point de départ pour l'algorithme de minimisation. Ce point de départ doit être le plus proche possible de la solution pour que la régression soit la plus efficace. La seconde étape est la régression elle-même. Après avoir exposé l'approche d'estimation des paramètres, nous comparons quelques méthodes de régression applicables, en notant que l'algorithme utilisé par Solina était celui de Levenberg-Marquardt.

### 3.1 Estimation des paramètres

Cette étape de première estimation des paramètres est très importante pour la suite de l'approximation. En effet, la plupart des méthodes de régression restent souvent piégées par les minima locaux. Il s'agit donc d'estimer les paramètres le plus près possible de la solution afin de donner le meilleur point de départ à l'algorithme de régression.

Nous n'allons pas essayer d'estimer les coefficients de courbure de la superellipsoïde. Il n'est pas possible de le faire simplement. C'est pourquoi, on fixe arbitrairement les paramètres  $\epsilon_1$  et  $\epsilon_2$  à 1. Cela revient à déterminer la simple ellipsoïde la plus proche des données.

#### 3.1.1 Ellipsoïde d'inertie

L'approche de Solina et Bajski est d'utiliser le centre de gravité des données et la matrice des moments centraux afin de déterminer la position et l'orientation de l'ellipsoïde. Comme nous pouvons le pressentir et comme nous allons le voir, cette approche statistique d'estimation fait l'hypothèse de données régulièrement réparties sur la surface. N'oublions pas que Solina et Bajski, qui ont proposé la méthode, travaillaient sur des données de type images de profondeur qui à défaut d'être complètes, sont des données régulièrement réparties.

Le centre de la superellipsoïde (permettant d'évaluer les paramètres de translation) est initialement le centre de gravité de l'ensemble de points :

$$\begin{aligned} t_x &= \bar{x} \\ t_y &= \bar{y} \\ t_z &= \bar{z} \end{aligned}$$

Pour estimer l'orientation de la superellipsoïde (les paramètres de rotation) et sa taille ( $a_1, a_2, a_3$ ), on utilise la matrice des moments centraux d'inertie d'ordre 2 :

$$M = \frac{1}{N} \sum_{i=1}^N \begin{pmatrix} (y_i - \bar{y})^2 + (z_i - \bar{z})^2 & -(y_i - \bar{y})(x_i - \bar{x}) & -(z_i - \bar{z})(x_i - \bar{x}) \\ -(x_i - \bar{x})(y_i - \bar{y}) & (x_i - \bar{x})^2 + (z_i - \bar{z})^2 & -(z_i - \bar{z})(y_i - \bar{y}) \\ -(x_i - \bar{x})(z_i - \bar{z}) & -(y_i - \bar{y})(z_i - \bar{z}) & (x_i - \bar{x})^2 + (y_i - \bar{y})^2 \end{pmatrix}$$

On cherche alors la matrice de rotation  $R$  qui diagonalise  $M$  :

$$D = R^{-1}MR$$

$R$  est la matrice dont les colonnes sont les vecteurs propres de  $M$  :  $e_1, e_2$  et  $e_3$  et  $D$  la matrice diagonale contenant les valeurs propres de  $M$  :  $\lambda_1, \lambda_2$  et  $\lambda_3$ .  $R$  est la matrice de rotation permettant de passer du repère absolu au repère intrinsèque de l'ellipsoïde. On va donc déduire de cette matrice l'estimation des paramètres de rotation en l'identifiant à une matrice de rotation standard. On aura donc, par exemple :

$$\begin{aligned} \phi &= -\arcsin(R_{1,3}) \\ \theta &= \arctan\left(-\frac{R_{2,3}}{\cos(\theta)}, -\frac{R_{3,3}}{\cos(\theta)}\right) \\ \psi &= \arctan\left(-\frac{R_{1,2}}{\cos(\theta)}, -\frac{R_{1,1}}{\cos(\theta)}\right) \end{aligned}$$

avec  $\phi, \theta$  et  $\psi$  les angles de rotation respectivement sur les axes  $x, y$  et  $z$ .

On a donc tous les paramètres de translation et de rotation, c'est-à-dire tous ceux contenus dans la matrice de transformation  $T$ . Il nous faut maintenant estimer les trois derniers paramètres de l'ellipsoïde, c'est-à-dire ceux définissant sa taille ( $a_1, a_2, a_3$ ). Rappelons que nous nous bornons à estimer les paramètres d'une simple ellipsoïde ( $\epsilon_1 = \epsilon_2 = 1$ ).

Dans l'approche originale, Solina utilisait la distance du centre de gravité au point le plus éloigné suivant chaque axe pour déterminer les paramètres de taille. Bien sûr cette méthode est extrêmement sensible aux points aberrants. C'est pourquoi nous utiliserons la méthode de Bardinet [BCA95] qui en comparant la matrice diagonale des valeurs propres et la matrice d'inertie  $J$  d'une ellipsoïde :

$$J = \frac{1}{3} \begin{pmatrix} a_2^2 + a_3^2 & 0 & 0 \\ 0 & a_1^2 + a_3^2 & 0 \\ 0 & 0 & a_1^2 + a_2^2 \end{pmatrix}$$

permet de calculer :

$$\begin{aligned} a_1^2 &= \frac{3}{2}(\lambda_2 + \lambda_3 - \lambda_1) \\ a_2^2 &= \frac{3}{2}(\lambda_1 + \lambda_3 - \lambda_2) \\ a_3^2 &= \frac{3}{2}(\lambda_1 + \lambda_2 - \lambda_3) \end{aligned}$$

On a ainsi estimé les neuf paramètres de l'ellipsoïde. Mais les valeurs obtenues ont été attribuées de manière arbitraire à chaque axe. Cela n'a pas d'importance dans le cas d'une ellipsoïde puisque que cette dernière est symétrique suivant les trois axes. Mais pour une superquadrique, il y a alors un choix à faire sur l'orientation du modèle et décider quel axe  $x$ ,  $y$  ou  $z$  on attribuera la plus grande taille (i.e. l'axe d'inertie). Le choix se limite en fait à  $x$  ou  $z$ , puisque  $x$  et  $y$  ont une influence complètement équivalente sur la forme de la superellipsoïde. Rappelons nous par exemple, que les cylindres ne peuvent être construits qu'avec l'axe des  $z$  pour axe de révolution. Ne pouvant estimer les paramètres de courbure de la superellipsoïde, nous ne pouvons que faire un choix arbitraire. Nous décidons, comme Solina, que  $z$  serait l'axe d'inertie ce qui privilégie les cylindres de type “tuyau” à l'encontre des “boîtes de camembert”.

De toutes façons, ce choix n'est théoriquement pas définitif puisque, lors de l'ajustage des paramètres, la rotation a une liberté de  $2\pi$  sur tous les axes. Néanmoins, certains algorithmes nécessitent un minimum déjà proche de l'estimation pour converger correctement.

Le résultat est relativement proche de la solution dans la plupart des cas, mais étant calculé entièrement grâce à des études de répartition des données, il peut s'en éloigner fortement dans le cas de données irrégulières. Il faut se rappeler que Solina travaillait sur des données de type image de profondeur qui bien qu'incomplètes sont beaucoup mieux réparties que celles auxquelles nous voulons nous confronter.

## Résultats

Nous allons maintenant présenter quelques résultats d'estimation obtenus grâce à cette méthode.

Les données sur lesquelles nous allons faire ces tests sont toutes issues de la surface d'une superellipsoïde. Nous avons sélectionné : une sphère, une ellipsoïde, un parallélépipède rectangle et deux cylindres (table 3.1). Chacun de ces objets a été échantillonné de deux manières différentes : un échantillonnage régulier tout à fait adapté à la méthode, et un échantillonnage grâce à la forme paramétrique régulière qui va concentrer les points dans les pôles des superquadriques. L'objet nommé *Sphere3* est particulier, il s'agit d'une sphère échantillonnée régulièrement mais coupée aux deux-tiers.

Les résultats de l'estimation sont présentés dans le tableau 3.1.1. Ils montrent à quel point la méthode est sensible à la régularité de l'échantillonnage. Dans tous les cas, la méthode est plus efficace pour des données de densité constante. Cela est particulièrement visible pour les objets Sphere. Bien entendu, cela était complètement prévisible puisque la méthode fait l'hypothèse de cette régularité pour déduire les paramètres.

Nom	Échantillonnage	Nombre de points	paramètres de la superellipsoïde				
			$a_1$	$a_2$	$a_3$	$e_1$	$e_2$
Sphère 1	paramétrique	1116	1.00	1.00	1.00	1.00	1.00
Sphère 2	régulier	1000	1.00	1.00	1.00	1.00	1.00
Cylindre XY 1	paramétrique	1116	3.00	3.00	1.00	0.01	1.00
Cylindre XY 2	régulier	1000	3.00	3.00	1.00	0.01	1.00
Cylindre Z 1	paramétrique	1116	1.00	1.00	3.00	0.01	1.00
Cylindre Z 2	régulier	1000	1.00	1.00	3.00	0.01	1.00
Ellipsoïde 1	paramétrique	1116	3.00	1.00	1.00	1.00	1.00
Ellipsoïde 2	régulier	1000	3.00	1.00	1.00	1.00	1.00
Parallélépipède 1	paramétrique	1116	1.00	1.00	3.00	0.01	0.01
Parallélépipède 2	régulier	1000	1.00	1.00	3.00	0.01	0.01

TAB. 3.1 – Différentes données synthétiques utilisées dans les tests

On peut aussi noter que ces résultats sont biaisés par le fait que ces données synthétiques sont obtenues de manière complètement symétrique par rapport aux trois axes. Cela explique la grande exactitude des paramètres de translation qui correspondent au barycentre de l'ensemble des points. Les autres paramètres qui sont déduits de la matrice de covariance, calculée à partir de ce barycentre, sont donc obtenus dans des circonstances très favorables. En fait, ces résultats montrent donc uniquement la sensibilité à la régularité de la densité des paramètres de facteur d'échelle et de rotation.

### 3.1.2 Estimation directe en se basant sur l'approximation directe d'ellipse

La méthode précédente d'estimation des paramètres initiaux est très commune et sera celle que nous utilisons, mais évoquons tout de même l'idée d'une estimation des paramètres basée sur une méthode d'approximation d'ellipsoïdes au sens des moindres carrés, non-itérative, qui serait l'extension de celle existant pour les ellipses proposée par Fitzgibbon [FPF99].

Son approche est la suivante : une conique générale peut être représentée par l'équation implicite appelé "distance algébrique" :

$$F(\mathbf{a}, \mathbf{x}) = \mathbf{a} \cdot \mathbf{x} = ax^2 + bxy + cy^2 + dx + ey + f = 0$$

avec  $\mathbf{a} = [a, b, c, d, e, f]$  et  $\mathbf{x} = [x^2, xy, y^2, x, y, 1]$ .

Si l'on désire approximer un ensemble de  $N$  points 2D, une approche est de minimiser

Objet	Distorsion	paramètres de la superellipsoïde													
		$a_1$	$a_2$	$a_3$	$e_1$	$e_2$	$t_x$	$t_y$	$t_z$	$r_x$	$r_y$	$r_z$			
Sphere 1	0.003217	0.866	0.866	1.224	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.394
Sphere 2	0.000439	0.999	0.999	0.999	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Sphere 3	0.005582	0.689	1.058	1.058	1.000	1.000	0.308	0.000	0.000	0.000	0.104	0.000	0.000	0.000	
Cylindre XY 1	0.007368	1.367	2.976	2.976	1.000	1.000	0.000	0.000	0.000	0.000	-0.016	1.570	0.000	0.000	
Cylindre XY 2	0.005283	1.707	2.614	2.614	1.000	1.000	0.000	0.000	0.000	0.000	0.000	1.570	0.009	0.000	
Cylindre Z 1	0.007058	0.992	0.992	4.103	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
Cylindre Z 2	0.005095	1.171	1.190	3.348	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.001	0.451	0.000	
Ellipsoïde 1	0.004259	1.204	0.880	2.640	1.000	1.000	0.000	0.000	0.000	0.000	0.000	1.570	0.000	0.000	
Ellipsoïde 2	0.001205	1.008	1.008	2.891	1.000	1.000	0.000	0.002	0.000	0.004	1.571	0.604	0.000	0.002	
Parallélépipède 1	0.008893	1.089	1.089	4.205	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	
Parallélépipède 2	0.006888	1.362	1.364	3.386	1.000	1.000	0.000	0.000	0.030	0.000	0.000	0.000	0.000	-0.382	

TAB. 3.2 – Estimation obtenus grâce à la méthode de l'ellipsoïde d'inertie

la somme des carrés de la “distance algébrique” d’un point à la courbe :

$$E(a) = \sum_{i=1}^N F(\mathbf{a}, \mathbf{x}_i)^2$$

sous la contrainte que la conique représentée par  $\mathbf{a}$  soit une ellipse, c’est-à-dire que :

$$4ac - b^2 > 0$$

soit :

$$a^T C a = a^T \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} a > 0$$

Notons que cette contrainte interdit aussi la solution triviale à savoir  $\mathbf{a}$  est un vecteur nul.

Si on considère maintenant la matrice de donnée  $D = [\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_N]^T$ , le problème devient : minimiser  $E(\mathbf{a}) = |D\mathbf{a}|^2$  avec  $\mathbf{a}^T C \mathbf{a} > 0$ .

Si on devait en rester là, le problème serait complexe, mais puisque toute conique définie par l’ensemble de paramètres  $\mathbf{a}$  peut aussi l’être avec les paramètres  $\mathbf{a}' = k * \mathbf{a}$  pour tout  $k \in \mathbb{R}^*$ , on peut simplifier la contrainte par  $\mathbf{a}^T C \mathbf{a} = 1$ .

En définitive, on doit minimiser  $E(\mathbf{a}) = |D\mathbf{a}|^2$  avec  $\mathbf{a}^T C \mathbf{a} = 1$ , et cette simplification nous permet d’utiliser le théorème de Lagrange. On a :

$$2D^T D a - 2\lambda C a = 0$$

$$D^T D a = \lambda C a$$

$$\frac{1}{\lambda} * a = (D^T D)^{-1} C a$$

On résout le système grâce à une simple extraction des vecteurs/valeurs propres.

## Tentative de passage en 3D

Il serait très intéressant de pouvoir utiliser une telle méthode pour l’estimation des paramètres de la superellipsoïde. La régularité de la répartition des données jouerait beaucoup moins sur le résultat et le critère (minimisation au sens des moindres carrés) est tout à fait adapté pour la suite de l’approximation.

Mais l'adaptation à la 3D n'est pas immédiate. Le problème est la contrainte des paramètres. En effet, pour une quadrique, la formule permettant d'assurer que la solution définit bien une ellipsoïde n'est pas quadratique. Elle ne peut donc pas être exprimée sous la forme  $\mathbf{a}^T C \mathbf{a}$ . Le problème ne peut alors plus être résolu de manière simple avec une extraction de valeurs et vecteurs propres et perd beaucoup de son intérêt.

## 3.2 Minimisation de l'énergie

Maintenant que nous savons obtenir une première estimation des paramètres de la superellipsoïde, nous devons sélectionner un algorithme de minimisation. La méthode de régression devra être applicable à une fonction non-linéaire et pourra utiliser ses dérivées puisque nous pouvons les calculer à la fois numériquement et analytiquement. L'algorithme utilisée par Solina est celui de Levenberg-Marquardt.

Bien que cet algorithme soit un classique de la régression non-linéaire, il a ses limites. Ainsi, certains essayèrent de réaliser cette minimisation à l'aide d'autres algorithmes et en particulier d'algorithmes stochastiques : algorithme génétique, recuit simulé.

Nous avons implémenté et testé différentes méthodes de régression : Levenberg-Marquardt, DSM, algorithmes génétiques. Après un bref rappel du fonctionnement de ces méthodes, nous compareront leurs performances en terme de temps de calcul et de qualité du résultat en les utilisant pour notre problème et en testant différentes approximations de la distance d'un point à la surface.

### Downhill simplex method (DSM)

La méthode DSM (Downhill Simplex Method) a été proposée par Nelder et Mead en 1965 [NM65]. Cette méthode n'a besoin, pour minimiser une fonction, que de son évaluation. Elle est souvent très efficace pour les fonctions non-dérivables (ce qui n'est pas notre cas) mais assez gourmande en temps de calcul.

Un simplex est défini comme une figure à  $N + 1$  sommets dans un espace de recherche de dimension  $N$ . Dans le cas de la minimisation d'une fonction  $f(x)$ , le simplex sera un segment ; dans celui de  $g(x, y)$ , ce sera un triangle. Rappelons que l'énergie que nous désirons minimiser a onze paramètres (équation 3.1), notre simplex sera donc une figure à douze sommets.

Le simplex initial peut être défini par un point initial  $P_0$  provenant d'une première estimation de la solution, puis pour  $i$  variant de 1 à  $N + 1$  :

$$P_i = P_0 + \lambda_i * e_i$$

où  $e_i$  est le  $i$ -ème vecteur unitaire de l'espace de recherche et  $\lambda_i$  un scalaire fonction de la variabilité des paramètres pour le problème traité.

Le principe du DSM est ensuite d'opérer une série successive d'étapes consistant à déformer le simplexe en modifiant un de ses points  $P_i$ . Si on note  $\bar{P}$  le barycentre de l'ensemble des points du simplexe privé de  $P_i$ , les opérations possibles de modifications sont alors les suivantes :

- la **réflexion** :  $P'_i = (1 + \alpha)\bar{P} - \alpha P_i$  avec  $\alpha$  une constante positive ;
- l'**expansion** :  $P'_i = (1 - \gamma)\bar{P} - \gamma P_i$  avec  $\gamma$  une constante positive ;
- la **contraction** :  $P'_i = (1 - \beta)\bar{P} - \beta P_i$  avec  $\beta$  une constante comprise entre 0 et 1.

La séquence des modifications opérées est alors la suivante :

1. On détermine dans un premier temps le meilleur  $P_l$  et le moins bon point  $P_h$  suivant la fonction d'évaluation à minimiser.
2. Une réflexion du point le moins bon est tentée. Si l'évaluation du point obtenu est comprise entre les évaluations de  $P_l$  et de  $P_h$ , on conserve ce nouveau point et on recommence en 1.
3. Sinon, si la valeur du point obtenu est meilleure que la valeur de  $P_l$ , on tente une expansion. On conserve les meilleurs des deux points (avant ou après expansion) et on reprend en 1.
4. Si après réflexion  $P_h$  reste le moins bon point, alors on conserve le meilleur des deux points (avant ou après réflexion) et on opère une contraction de ce point avant de recommencer en 1.

Une séquence appropriée de telles opérations conduira à la convergence du simplexe vers un minimum.

L'algorithme de DSM, que nous allons tester, possède en plus une opération de mutation d'un point du simplexe qui va permettre en cas de faible progression de redynamiser le système en lui réinjectant de l'énergie.

## Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt est une méthode classique de la régression non-linéaire [PTVF92, BCA95]. Il nécessite les dérivées du premier ordre de la fonction à minimiser.

Cet algorithme va osciller entre deux méthodes de minimisation. La première est une simple descente de gradient, qui peut être formulée ainsi :

$$\mathbf{a}' = \mathbf{a} - C\nabla\Delta(\mathbf{a}) \quad (3.2)$$

avec  $\mathbf{a}$  un ensemble de onze paramètres définissant une superquadrique et  $C$  une constante définissant le pas de descente.

La seconde méthode est une méthode de descente quadratique, qui va supposer que, lorsque l'on est proche de la solution, la fonction à minimiser peut être approximée par une forme quadratique. Le minimum de l'énergie à minimiser est alors le minimum de la forme quadratique qui peut se calculer directement :

$$\mathbf{a}_{min} = \mathbf{a} + \mathbf{D}^{-1} \cdot [-\nabla\Delta(\mathbf{a})] \quad (3.3)$$

avec  $D$  la matrice Hessienne de  $\Delta(\mathbf{a})$ .

L'idée de l'algorithme de Levenberg-Marquardt est de combiner ces approches en remarquant que quand  $\mathbf{D}$  est diagonale la méthode quadratique devient une simple descente de gradient. Si on pose alors  $\mathbf{D}'$  telle que :

$$\begin{aligned} \mathbf{D}'_{jj} &= \mathbf{D}_{jj} * (1 + \lambda) \\ \mathbf{D}'_{ij} &= \mathbf{D}_{ij} \quad (j \neq i) \end{aligned}$$

La formule permettant la régression devient :

$$\mathbf{a}_{min} = \mathbf{a} + \frac{1}{\lambda} \mathbf{D}'^{-1} \cdot [-\nabla\Delta(\mathbf{a})]$$

Ainsi, quand  $\lambda$  sera grand, la matrice  $D'$  sera presque diagonale et donc on sera proche de l'équation 3.2 et de la méthode de descente de gradient, tandis que quand  $\lambda$  sera proche de 0, la regression se fera à la manière d'une descente quadratique (équation 3.3)

Si  $\mathbf{a}$  est un ensemble de paramètres de la fonction à minimiser à une étape donnée de la regression, l'étape suivante se déroule alors ainsi :

1. Calculer  $\mathbf{a}'$
2. Si  $\Delta(\mathbf{a}') > \Delta(\mathbf{a})$  alors  $\lambda = 10 * \lambda$
3. Si  $\Delta(\mathbf{a}') \leq \Delta(\mathbf{a})$  alors  $\lambda = 0.1 * \lambda$  et  $\mathbf{a} = \mathbf{a}'$ .

On répète itérativement ce procédé jusqu'à ce que  $\Delta(\mathbf{a})$  ne régresse plus, ce qui revient à s'arrêter quand  $\lambda$  est trop grand.

L'algorithme de Levenberg-Marquardt converge généralement assez rapidement mais se laisse piéger par les minima locaux. Il est toujours possible, comme dans le cas du DSM de se sortir de ces minima locaux en secouant les paramètres après une première convergence puis de relancer la régression. Nous l'avons fait, mais les résultats présentés ici, ont été obtenus grâce à une simple application de l'algorithme de Levenberg-Marquardt.

## Algorithme Génétique

Un algorithme génétique est une approche d'optimisation stochastique qui se base sur les lois de la sélection naturelle de l'évolution des espèces.

Une population  $P$  d'individus va évoluer, chacun d'entre-eux étant muni d'un ensemble de gènes. Un individu est en fait un point dans l'espace de recherche. Cette évolution conduira à une amélioration de la population au sens de la fonction à minimiser. À chaque étape, le meilleur individu sera une solution possible de l'optimisation.

Initialement, la population est générée de façon aléatoire dans l'espace de recherche. Pour nous, il s'agira d'un ensemble de superellipsoïdes et les gènes seront les onze paramètres. L'évolution de cette population est, en fait, une succession de générations jusqu'à ce que la convergence ait lieu.

Chaque génération se décompose en deux phases. La première phase est celle dite de sélection. Tous les individus vont être évalués par la fonction à optimiser puis triés en fonction de cette évaluation, pour nous la valeur de l'équation 3.1 de la page 44. Ensuite un ensemble de couples de même taille que la population va être choisi. Plus le classement d'un individu est bon, plus il aura de chances d'être dans un couple. Bien sûr, un même individu peut être dans plusieurs couples à la fois.

Ensuite, durant la phase dite de reproduction, les couples vont donner naissance à une nouvelle population. Compte tenu du mode de sélection des couples, les individus les moins bons ne se reproduiront presque jamais (c'est la sélection naturelle). De chaque couple sortira un nouvel individu dont les gènes seront un mélange aléatoire de ceux de ses parents. De plus, chaque gène pourra aléatoirement :

1. être transmis intact ;
2. résulter d'un croisement des gènes des deux parents, c'est-à-dire un mélange qui pourra se faire soit au niveau binaire (les  $n$  (aléatoire) premiers bits venant du père, les  $n$  seconds de la mère), soit en restant avec les réels (si les paramètres sont réels) avec un moyennage pondéré (aléatoire) par exemple ;
3. subir une mutation, c'est-à-dire une altération aléatoire.

Cette approche va faire évoluer une population qui sera de plus en plus proche de la solution. Pour garantir la progression, une technique simple est de garder le(s) meilleur(s) individu(s) d'une génération sur l'autre.

Cet algorithme ne pose pas le problème des minima locaux. En revanche, il nécessite le réglage de nombreux paramètres : le taux de mutation des gènes, leur taux de croisement, la taille de la population, etc.

## Comparaison des résultats

Nous avons essayé d'utiliser ces trois algorithmes de minimisation pour la phase de régression de la méthode d'approximation d'un ensemble de points par une superellipsoïde. Pour chacun de ces algorithmes, les tests ont été réalisés avec deux méthodes d'approximation de la distance : la fonction "standard"  $F_1$  d'approximation et la distance "euclidienne radiale"  $F_2$ .

Nous présentons ici les résultats sur trois exemples d'objets simples : un parallélépipède rectangle constitué d'un échantillon de 1013 points, un cylindre (1059 points) et une ellipsoïde tronquée aux trois quarts dans le sens de la longueur (1070 points) (figure 3.1).

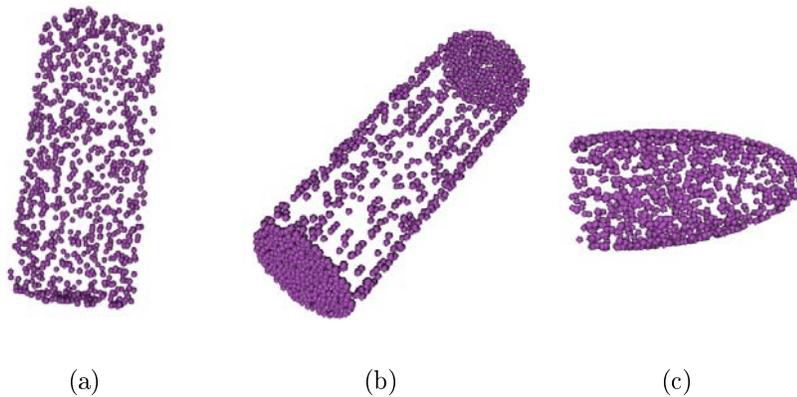


FIG. 3.1 – Les ensembles de points non bruités utilisés pour les tests d'approximation (a) L'objet Rectangle (b) L'objet Cylindre (c) L'objet Ellipsoïde tronquée

Les figures 3.2 à 3.4 montrent l'évolution de la distorsion en fonction du temps lors de l'utilisation des trois algorithmes. Le temps est exprimé en secondes. La distorsion est calculée avec la fonction de distance utilisée ; c'est la valeur de la fonction énergie minimisée. Pour toutes ces figures, la régression par Levenberg-Marquardt est représentée par un trait plein, celle par algorithme génétique l'est par des tirets tandis que les pointillés correspondent au DSM.

Cette expression de l'évolution de la distorsion en secondes est utile pour comparer concrètement l'efficacité des méthodes dont nous disposons. Néanmoins, il faut garder à l'esprit qu'elle n'est valable que pour les implémentations des algorithmes que nous utilisons et qu'il peut donc y avoir des variations importantes suivant les versions et optimisations apportées à tel ou tel algorithme. Pour avoir une vision plus indépendante de la situation on peut noter que sur les exemples d'environ 1000 points, nos versions du DSM et l'algorithme de Levenberg-Marquardt progresse d'environ 40 itérations par seconde alors que l'algorithme génétique progresse à une vitesse de 5 itérations par seconde pour

une population de 100 individus. La vitesse de ces trois algorithmes varie de manière globalement linéaire en fonction du nombre de points.

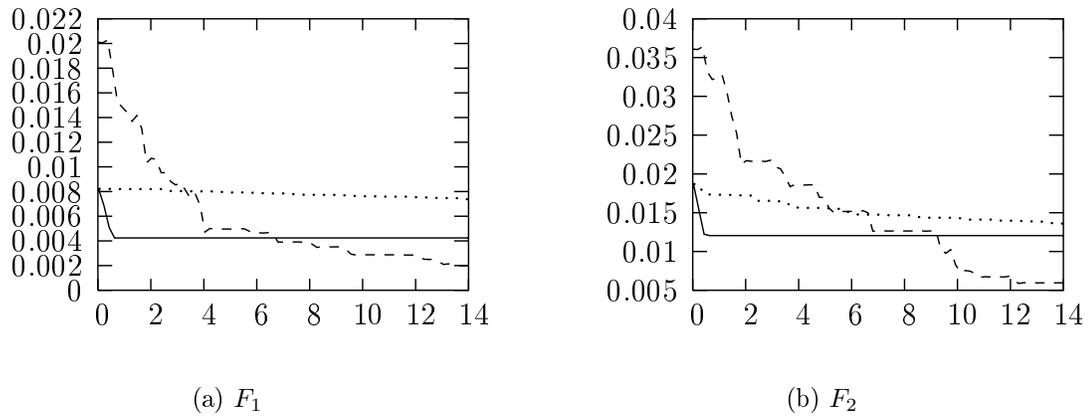


FIG. 3.2 – Évolution de la distorsion aux données Rectangle en fonction du temps en seconde lors de la régression par Levenberg-Marquardt (trait plein), Algorithme Génétique (tirets) et DSM (pointillés). La distorsion est exprimée en fonction de la “distance” utilisée, d’où la différence d’ordre de grandeur. Pour une comparaison des distorsions obtenues il faut se reporter au tableau 3.3 (a) Utilisation de la distance “standard” (b) Utilisation de la distance “euclidienne radiale”

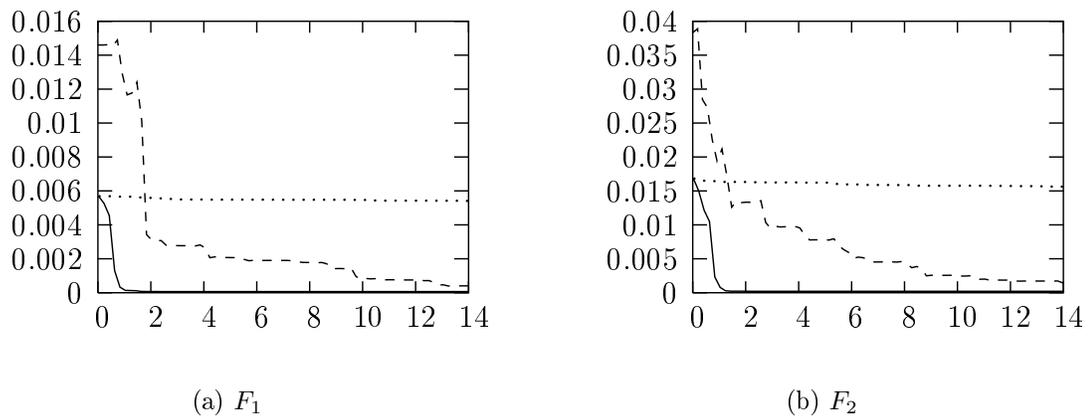


FIG. 3.3 – Même chose que pour la figure 3.2 appliquée aux données Cylindre

La figure 3.5 propose une autre courbe d’évolution de la distorsion sur l’objet Ellipsoïde tronquée auquel nous avons ajouté un peu (fig. 3.5a et 3.5b) puis beaucoup (fig. 3.5c et 3.5d) de bruit. À cause de notre méthode d’estimation des paramètres relativement dépendante du barycentre de l’ensemble de points, le fait d’utiliser une ellipsoïde tronquée donne un point de départ très éloignés. Cela n’empêche pas la convergence qui est d’ailleurs

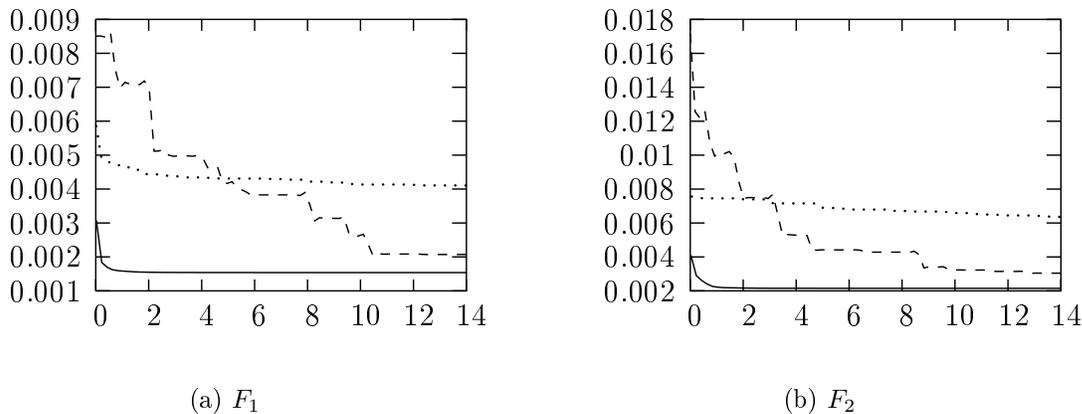


FIG. 3.4 – Même chose que pour la figure 3.2 appliquée aux données Ellipsoïde.

très rapide et de bonne qualité dans le cas de Levenberg-Marquardt. On note aussi que plus il y a de bruit et moins l'approximation est bonne. Cela ne signifie pas que la méthode de régression est moins efficace dans ce cas (on voit d'ailleurs que les courbes ont un profil similaires). Simplement, la meilleure superellipsoïde est moins proche des données à cause de ce bruit (il s'agit de la distorsion ajoutée dans les données en quelque sorte).

La figure 3.6 montre l'évolution de la distorsion pour l'objet Cylindre que nous avons dans un premier temps sous-échantillonné (fig. 3.6a et 3.6b) puis sur-échantillonné (fig. 3.6c et 3.6d). Les courbes montrent une croissance relativement linéaire du temps nécessaire à la régression et surtout de bien moins bon résultats sur le cylindre sous-échantillonné que sur le cylindre original et sur-échantillonné.

Enfin, nous avons cumulé les deux modifications précédentes sur le parallélépipède rectangle pour produire les courbes de la figure 3.7. Les courbes 3.7a ont été obtenues après un sous-échantillonnage et l'ajout d'un bruit léger, tandis que les courbes 3.7b représentent l'évolution de la distorsion après avoir sur-échantillonné puis fortement bruité le parallélépipède.

Ces figures nous montrent plusieurs choses. Premièrement on voit clairement que l'algorithme DSM n'est pas du tout adapté à notre problème. Ses courbes sur cet intervalle de quinze secondes (dans le cas des objets d'un milliers de points) semblent presque horizontales. La régression est donc très lente. L'algorithme génétique donne de meilleurs résultats que le DSM mais l'algorithme de Levenberg-Marquardt a clairement l'avantage. Deuxièmement, quelle que soit la méthode de régression, l'utilisation de l'une ou l'autre approximation de la distance d'un point à la surface ne semble apporter aucun gain d'efficacité en terme de vitesse. La différence de distorsion visible entre les résultats à partir de l'approximation de la distance  $F_1$  ou  $F_2$  est simplement due au fait que cette distorsion

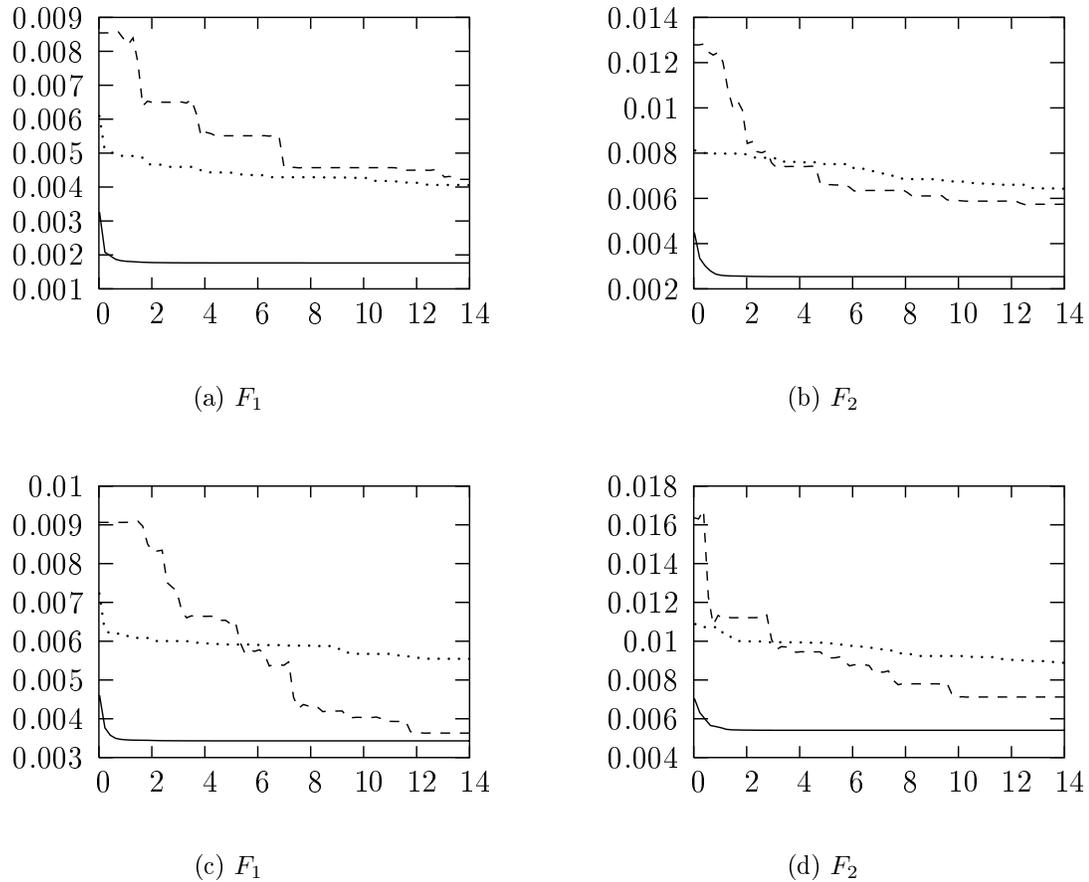


FIG. 3.5 – Même chose que pour la figure 3.2 appliquée aux données Ellipsoïde bruitées dans le cas de (a) et (b) et très bruitées dans le cas de (c) et (d).

est justement exprimée en fonction de  $F_1$  dans le premier cas et de  $F_2$  dans le second et n'est donc pas comparable.

Le tableau 3.3 montre la distorsion réelle, en terme de distance euclidienne, que nous obtenons. On voit clairement que les meilleurs résultats, en gras, sont obtenus grâce à l'algorithme de Levenberg-Marquardt. L'algorithme DSM, quant à lui est très en retrait.

L'approximation de la distance utilisée semble n'avoir que peu d'importance. Cela ne remet pas en cause le fait que la distance euclidienne radiale est plus approprié à notre problème. Car ce n'est pas tant pour l'approximation que nous proposons son utilisation mais surtout lors de la segmentation afin d'avoir une distorsion normalisée par rapport à la taille et la forme de toute superellipsoïde. Les résultats montrent que la méthode est assez sensible aux taux d'échantillonnages, les distorsions sur les données sous-échantillonnées étant dix fois plus élevées que les autres.

D'un point de vue plus général, en ne considérant pas seulement les exemples présentées

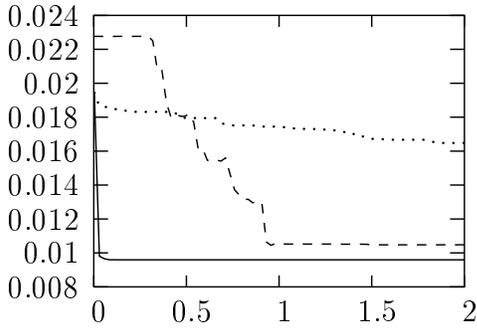
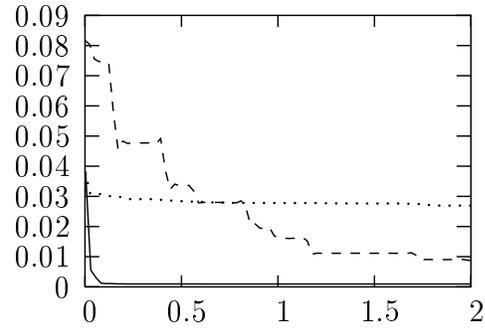
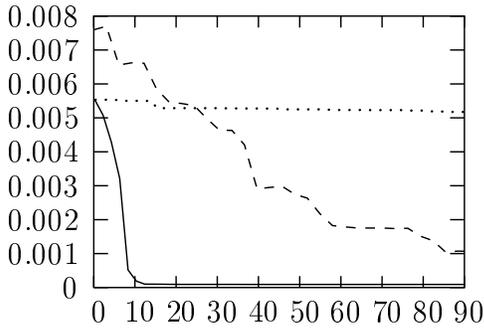
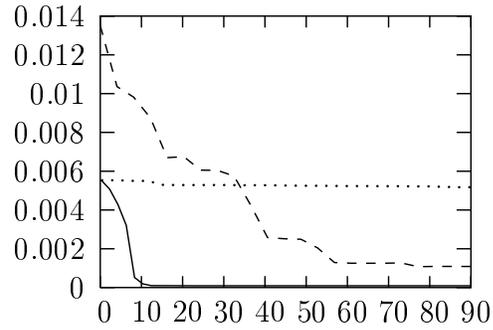
(a)  $F_1$ (b)  $F_2$ (c)  $F_1$ (d)  $F_2$ 

FIG. 3.6 – Même chose que pour la figure 3.2 appliquée aux données Cylindre sous-échantillonnées (143 points) dans le cas de (a) et (b) et sur-échantillonnées (8231 points) dans le cas de (c) et (d).

dans ce chapitre mais l'ensemble des tests effectués durant notre travail, on peut dire que l'algorithme génétique et l'algorithme de Levenberg-Marquardt donnent tous deux de bons résultats. Le premier a parfois l'avantage de donner de meilleurs résultats sur des temps de régressions plus longs que ceux que nous avons présentés ici et le second d'être beaucoup plus efficace dans les premières secondes de la minimisation. De plus, un avantage de Levenberg-Marquardt est de ne nécessiter aucun paramétrage, tandis qu'avant de pouvoir utiliser l'algorithme génétique, il faut fixer la taille de la population, le taux de mutations, le taux de croisement et divers autres paramètres qui sont venus compléter les méthodes récentes. Pour conclure, les alternatives à l'algorithme de Levenberg-Marquardt que nous avons testées n'ont pas su être suffisamment convaincantes et nous garderons donc cette méthode de régression qui avait été utilisée dans la méthode originale de Solina et dans beaucoup des suivantes.

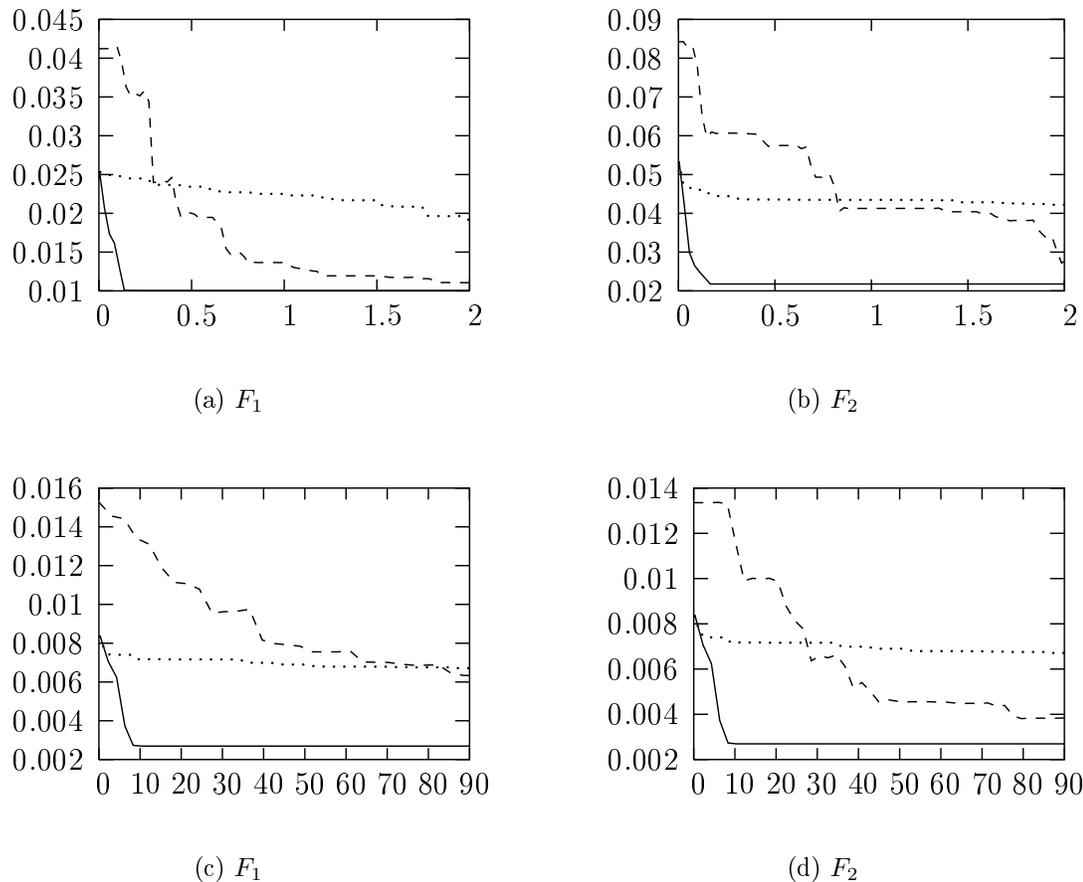


FIG. 3.7 – Même chose que pour la figure 3.2 appliquée aux données Rectangle bruitées et sous-échantillonnées (149 points) dans le cas de (a) et (b) et sur-échantillonnées (8231 points) dans le cas de (c) et (d).

### 3.3 Conclusion

Nous avons présenté dans ce chapitre une méthode pour approximer des données tri-dimensionnelles par une superellipsoïde. Cette approximation se fait au sens des moindres carrés de la distance comme proposé par Solina.

La première étape de cette approximation est une estimation des paramètres du modèle qui permettront d'initialiser une méthode de régression. La méthode originale est d'utiliser l'ellipsoïde d'inertie et est par conséquent complètement dépendante de la régularité des données. Nous avons étudié une autre méthode basée sur le calcul direct des paramètres de l'ellipse approximant des données 2D au sens des moindres carrés. Cette approche serait donc adaptée à notre problème et est de plus indépendante à la régularité de la densité des données mais n'a pas pu être adaptée à la troisième dimension.

Distance utilisée	Levenberg Marquardt		Algorithme Génétique		DSM	
	Standard	Euclidienne Radiale	Standard	Euclidienne Radiale	Standard	Euclidienne Radiale
Rectangle	0.0040	0.0052	0.0030	<b>0.0026</b>	0.0058	0.0064
Rectangle -b	<b>0.0084</b>	0.011	0.0097	0.013	0.014	0.020
Rectangle +B	0.0020	<b>0.0013</b>	<b>0.0013</b>	0.0015	0.0025	0.0026
Cylindre	0.001	<b>0.0009</b>	0.004	0.0009	0.076	0.0058
Cylindre -	0.008	<b>0.007</b>	0.009	<b>0.007</b>	0.013	0.014
Cylindre +	<b>0.0003</b>	<b>0.0003</b>	0.0004	0.0003	0.0019	0.0020
Ellipsoïde	0.0018	<b>0.0016</b>	0.0017	0.0014	0.0038	0.0050
Ellipsoïde b	0.0021	<b>0.0018</b>	0.0038	0.0051	0.0037	0.0048
Ellipsoïde B	0.0036	<b>0.0034</b>	0.0084	0.0151	0.0053	0.0063

TAB. 3.3 – Valeurs de la distorsions des données à une superellipsoïde avec les différentes méthodes après 20 secondes de régression sur les objets d'un milliers de points, 4 secondes pour les objets de 150 points et 150 secondes pour les objets de 8000 points. (légende : - pour sous-échantillonnées, + pour sur-échantillonnées, b pour bruitées, B pour très bruitées)

En ce qui concerne la régression, nous avons mis en place trois algorithmes différents : Levenberg-Marquardt, algorithme génétique et DSM. La méthode de Levenberg-Marquardt qui est la plupart du temps utilisée pour ce problème s'avère, aussi dans notre cas, la plus efficace et donc celle que nous utiliserons pour la suite de cette étude.

En plus de la fonction d'approximation de la distance classiquement utilisée pour ce problème nous avons essayé d'utiliser la distance euclidienne radiale présentée dans le chapitre précédent. Les résultats obtenus avec les deux approches de la distance sont très similaires. Cependant, une fonction de distance normalisée par rapport à la taille des superellipsoïdes, comme la distance euclidienne radiale, sera utile pour comparer les distorsions de superellipsoïdes diverses lors de la segmentation qui constitue la prochaine étape dans l'obtention de notre descripteur d'objet 3D et le sujet du prochain chapitre.

# Chapitre 4

## Segmentation des données tridimensionnelles

Nous désirons obtenir un descripteur d'objet tridimensionnel basé sur la segmentation de ce dernier. Ce descripteur sera un ensemble compact de superellipsoïdes plus ou moins structuré, chacune d'entre-elles modélisant une partie des données et donc de l'objet. Nous avons vu au paragraphe précédent comment approximer un ensemble de points 3D par une superellipsoïde, mais un objet tridimensionnel ne peut en général pas être décrit par une seule superellipsoïde. Il nous faut maintenant une méthode permettant de passer des données à une décomposition en sous-ensembles de points représentables par une unique superellipsoïde. Bien sûr cette décomposition doit produire un nombre minimal de parties pour une distorsion donnée.

Dans ce chapitre, nous allons détailler deux algorithmes de segmentation des données tridimensionnelles. Le premier est une méthode ascendante de type croissance de régions, le second, descendant, de type découpage-fusion. Pour ces deux algorithmes, comme indiqué dans l'introduction de cette première partie, nous menons de front segmentation et approximation de données. Ainsi, à la fin de la segmentation, notre descripteur et donc l'ensemble de superellipsoïdes est complètement défini.

### 4.1 Local à global : croissance de région

Cette première approche de segmentation des données tridimensionnelles a été, à l'origine, implémentée en parallèle à notre méthode originale que nous exposerons dans la partie suivante. Cette méthode est basée sur un algorithme de type croissance de régions et a été proposée par Léonardis [LJS97, LSM94]. Il appliquait cette méthode à des données de type images de profondeur. Un travail d'adaptation pour l'appliquer à nos données non-organisées et irrégulières a dû être effectué. Ce qui au final, en fait une variante assez personnelle.

La méthode se décompose en trois phases principales :

1. **Initialisation des germes** : l'initialisation de la méthode correspond à la création d'un ensemble de germes, c'est-à-dire un ensemble de régions de l'espace contenant un nombre réduit de points tridimensionnels voisins. Chacune de ces régions, ou de ces germes, doit pouvoir être approximée par une superellipsoïde avec une qualité donnée.
2. **Phase de croissance** : durant cette phase, les germes vont tenter de grossir en incorporant des points qui leur sont voisins.
3. **Phase de sélection** : cette étape est chargée d'éliminer les germes qui sont devenus trop redondants, qui décrivent une même partie de l'objet.

Après l'initialisation des germes, des alternances de phase de croissance et de sélection se succéderont jusqu'à ce que l'ensemble de régions soit stable.

Nous allons dans un premier temps détailler ces trois phases, puis commenterons les résultats que nous obtenons.

#### 4.1.1 Initialisation des germes

La méthode d'initialisation est la même que dans l'approche originale de Leonardis. L'espace des données est partitionné suivant une grille tridimensionnelle. Chaque sous-ensemble de cet espace devient alors un germe si et seulement si il contient assez de points et qu'il peut être approximé par une superellipsoïde pour un taux de distorsion donné :  $\tau_i$ .

On obtient, à la fin de cette étape, un ensemble de superellipsoïdes, chacune d'entre-elles approximant des points qui lui sont propres et un ensemble de points tridimensionnels indépendant provenant des régions de l'espace qui n'ont pas donné naissance à un germe. Initialement, tous ces germes sont marqués *en croissance*.

Cette phase est primordiale pour la suite de l'algorithme, si une partie de l'objet tridimensionnel traité ne se voit pas attribuer de germes, elle risque de ne pas être représentée dans le descripteur final. En effet, comme nous allons le voir, les germes sont censés grossir, c'est-à-dire assimiler des points, mais ne peuvent normalement pas traverser les frontières entre les parties naturelles de l'objet. Ainsi, un germe initialisé dans une partie de l'objet a peu de chance d'en modéliser une autre à la fin de l'algorithme (figure 4.1).

#### 4.1.2 Phase de croissance

Maintenant que nous avons initialisé un certain nombre de germes, nous allons essayer d'ajouter des points à chacun de ces germes afin que chaque ensemble tende à regrouper exactement les points d'une partie naturelle de l'objet.

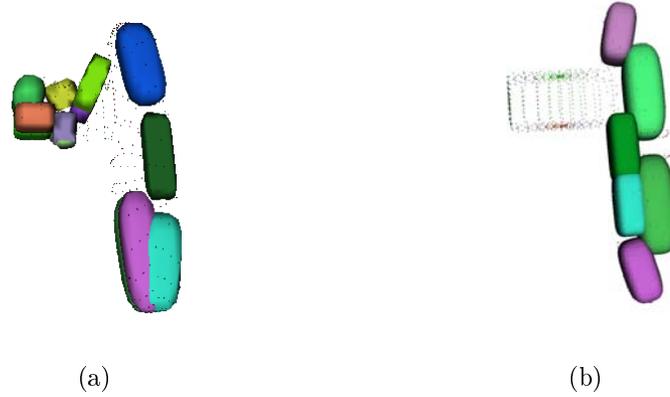


FIG. 4.1 – Exemples d’initialisation des germes. Sur la figure (b), une partie de l’objet ne contient pas de germes. Elle ne sera sans doute pas représentée dans le descripteur final.

Il nous faut d’abord définir le critère nous permettant de choisir si oui ou non un ensemble de points peut être ajouté à une région. Mis à part une distance trop grande de la superellipsoïde aux données, il n’y a pas de raison pour rejeter un tel ajout. Nous utiliserons donc simplement un seuil ( $\tau_g$ ) sur la distorsion  $\Delta$  au-dessus duquel les nouveaux points ne seront pas acceptés. Pour qu’un ensemble de points puisse être ajouté à un germe, il faut et il suffit qu’il existe une superellipsoïde de paramètres  $\mathbf{a}$  tels que, pour l’union des points du germe et de ces nouveaux points :

$$\Delta(\mathbf{a}) < \tau_g$$

Notons que  $\tau_i$  et  $\tau_g$  sont tous deux les taux de distorsion maximum que l’on désire pour la modélisation de l’objet, plus précisément de la distorsion maximale d’une superellipsoïde à ses points associés. Théoriquement, ils devraient donc avoir la même valeur. Cependant, pour être sûr d’avoir des germes dans toutes les parties de l’objet, il peut être intéressant d’utiliser un seuil d’initialisation ( $\tau_i$ ) légèrement supérieur au seuil de croissance ( $\tau_g$ ). Cela a peu de chance d’altérer la qualité du descripteur final, car un germe a très peu de chance d’y figurer identique à ce qu’il était lors de son initialisation.

Le critère d’acceptation étant défini, nous devons maintenant déterminer une méthode de sélection de ces nouveaux points. Les points qui ont le plus de chance d’être dans la continuité du germe sont évidemment des points dans le voisinage de la région. En pratique, au lieu de sélectionner les points les plus proches de la région, ce qui conduirait à déterminer un grand nombre de distances, entre des points à l’intérieur et à l’extérieur de cette région, nous sélectionnons un ensemble candidat des  $n$  points les plus proches de la superellipsoïde modélisant la région.

Dans l’approche originale, Leonardis préconise alors de vérifier si l’union de cet ensemble candidat et de la région traitée vérifie le critère de croissance précédemment défini,

puis en fonction, assimiler ou pas les points et continuer avec une autre région. Nous avons remarqué que dans notre cas, cela rendait la croissance trop difficile. Il est vrai que cette notion de croissance par points voisins est beaucoup plus adaptée aux données de type images de profondeur, relativement régulières et organisées (i.e. on connaît le voisinage de chaque point) qu'aux données que nous traitons, dont nous n'avons aucune connaissance *a priori* de leur régularité ou organisation.

Notre approche est de ne pas rejeter complètement l'ensemble des points candidats. Après un premier rejet de la totalité des points candidats, nous divisons aléatoirement l'ensemble des candidats en deux parties, nous essayons de les assimiler de manière indépendante, puis nous considérons le meilleur de ces sous-ensembles. Si ce dernier est assimilable nous l'assimilons, et nous passons à un autre germe, sinon nous recommençons récursivement ce procédé jusqu'à ce que l'ensemble des candidats ne compte plus qu'un point. Dans ce cas, le rejet est définitif.

Ainsi, au cours d'une phase de croissance nous allons, pour chaque germe marqué *en croissance*, sélectionner ses voisins les plus proches et essayer de les ajouter, tous ou en partie, à ce germe suivant la méthode décrite ci-dessus. Un germe qui n'aura alors pu assimiler aucun point sera marqué *mature*, et sera ainsi écarté de toutes les prochaines phases de croissance (figure 4.2).

Compte tenu du mode opératoire de la phase de croissance, les régions de l'objet ainsi définies ne forment pas une partition des données. Si plusieurs phases de croissance se succèdent, les germes qui ont été initialisés dans la même partie naturelle de l'objet sont associés au même ensemble de points tridimensionnels : l'ensemble des points faisant partie de cette partie naturelle. Si les frontières des parties d'un objet sont floues, certaines régions se chevauchent.

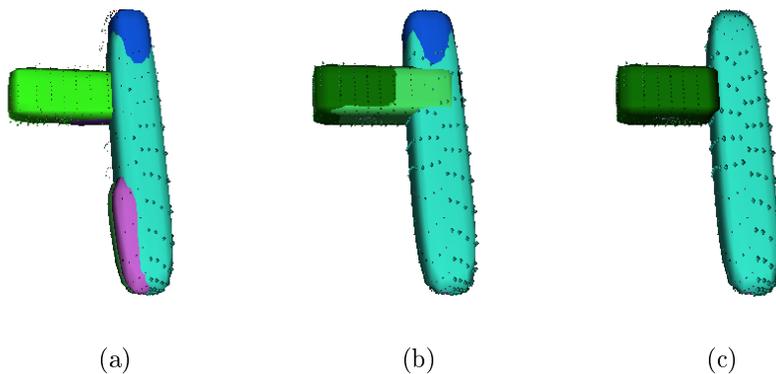


FIG. 4.2 – Exemple de croissance

### 4.1.3 Phase de sélection

Comme nous venons de l'évoquer, certains ensembles de points associés à une superellipsoïde ne sont pas disjoints. En terme de compacité du descripteur, ce n'est pas bon puisque cela signifie que les mêmes points sont représentés plusieurs fois. Cependant, comme nous venons de le dire, les parties naturelles des objets ne sont pas toujours franches, et un certain chevauchement des régions n'implique pas forcément une erreur de description. La phase de sélection va faire un tri entre tous les germes (*en croissance* ou *mature*). Son rôle est d'éliminer la redondance de description. Mais cette phase de sélection doit aussi préserver un ensemble de germes suffisant pour décrire l'ensemble de l'objet.

Pour opérer cette sélection, nous allons procéder d'une manière similaire à la méthode originale proposée par Leonardis. Nous utilisons un vecteur binaire de présence  $Q$  dont la taille correspond au nombre de germes encore présents dans la description (*en croissance* ou *mature*). Chaque vecteur correspond à une configuration différente de la description pour l'itération suivante.

La détermination de la meilleure configuration passe par la maximisation d'une fonction. Si on note  $N$  le nombre de germes *en croissance* et *mature*, il s'agit de déterminer  $Q$  tel que :

$$\max_Q(Q^T.M.Q) \quad (4.1)$$

avec :

- $Q$  le vecteur de configuration de taille  $N$  où  $Q_i = 1$  signifie que le  $i^{eme}$  doit être conservé tandis que  $Q_i = 0$  signifie que le  $i^{eme}$  germe doit être supprimé du descripteur ;
- $M$  une matrice dont la diagonale reflète la taille et la distorsion de chaque germe, et dont les termes hors diagonale caractérisent l'intersection des ensembles de points correspondants aux germes.

Comme dans l'approche originale, l'optimisation et la détermination de  $Q$  se fait grâce à un algorithme *glouton*, mais cependant nous avons défini notre propre matrice de sélection beaucoup plus adaptée aux données 3D irrégulières que celle proposée dans la méthode originale. Nous utilisons la matrice  $M'$  de taille  $N * N$  telle que :

- $M'_{i,i} = (1 - \frac{\chi_i}{T})\frac{n_i}{n}$  où  $n_i$  est le nombre de points contenus dans le sous-ensemble correspondant au  $i^{eme}$  germe,  $n$  le nombre total de points dans l'objet,  $\chi_i$  est la distorsion moyenne du  $i^{eme}$  germe et  $T$  la distorsion maximale que nous acceptons ;
- $M'_{i,j} = -K\frac{\chi_{i,j}}{T}\frac{n_{i,j}}{n_i}$  où  $i \neq j$ ,  $n_{i,j}$  est le nombre de points contenu dans l'intersection des sous-ensembles correspondants aux  $i^{eme}$  et  $j^{eme}$  germes, et  $\chi_{i,j}$  la distorsion moyenne entre la superellipsoïde du  $i^{eme}$  germe et l'ensemble des points présents dans l'intersection des sous-ensembles  $i$  et  $j$ .

Il faut comprendre les termes de la diagonale de la matrice comme un capital de points pour chaque germe qui prend en compte à la fois l'importance de la région en terme de taille ( $\frac{n_i}{n}$ ) modulé par la qualité d'approximation ( $1 - \frac{x_i}{T}$ ). Par exemple, un germe contenant la moitié des points de l'objet et approximant de manière parfaite les données aura un capital de 0.5. Si la distorsion de ce germe vaut la moitié de la distorsion maximum que l'on tolère, ce capital tombera à 0.25, etc.

Les termes hors diagonale sont quant à eux des malus qui vont faire baisser ce capital de départ. Ils sont calculés suivant un principe similaire mais s'intéresse à l'intersection des germes, c'est-à-dire leur redondance. Pour un germe donné, la valeur du malus sera d'autant plus importante que sa redondance est forte (taille de l'intersection importante par rapport à sa propre taille) et la distorsion élevée sur cette zone. La constante scalaire  $K$  sert à donner plus ou moins de poids à ces malus.

Compte tenu de la méthode de sélection (équation 4.1), seuls les germes dont la somme du capital et des malus donne un résultat positif vont être conservés.

L'alternance entre la phase de croissance et la phase de sélection peut avoir une fréquence très variable. On peut imaginer ne pas faire qu'une seule phase de croissance mais plutôt trois ou bien quatre avant chaque sélection. À l'extrême, il est tout à fait possible de ne procéder à une phase de sélection qu'une seule fois, après qu'aucun germe ne peut plus du tout grossir. Alternner les phases de croissance et de sélection a pour but de faire diminuer le nombre de germes, et ainsi de réduire d'autant le nombre d'approximations à faire à chaque tour. Il est donc intéressant de supprimer les germes aussi rapidement que possible pour accélérer la segmentation, mais il faut faire attention de ne pas supprimer ceux qui ont un fort potentiel de croissance.

#### 4.1.4 Résultats

Nous présentons ici les résultats obtenus avec cette méthode.

La figure 4.3 montre l'évolution de la segmentation à travers différentes étapes. L'objet original est un objet synthétique construit à partir de trois superellipsoïdes (deux sphères et un cylindre) puis échantillonné. Nous présentons une série d'étapes parmi les plus significatives. Notamment, à la fin de l'algorithme, la stabilisation du descripteur à trois primitives et le passage de ces primitives du statut de *en croissance* à *mature* (qui est nécessaire pour l'arrêt de l'algorithme) a été allégé. À la fin de cette segmentation, le descripteur contient donc trois superellipsoïdes et est fidèle au modèle utilisé pour obtenir les données.

On peut s'interroger sur le passage des étapes 4.3e à 4.3f où l'on voit que la sélection élimine deux primitives (une rose et une verte) de la première sphère qui paraissent pourtant conformes aux données. Cependant, il faut bien comprendre que lors de la sélection, on considère les sous-ensembles de points modélisés et la qualité de l'approximation.

Or obtenir une sphère, telle que celle éliminée, ne nécessite pas spécialement un grand nombre de points, ainsi il est tout à fait possible que les sous-ensembles de points représentés par les superellipsoïdes rose et verte soient moins intéressants pour la segmentation que d'autres sous-ensembles dont la représentation est moins "remarquable".

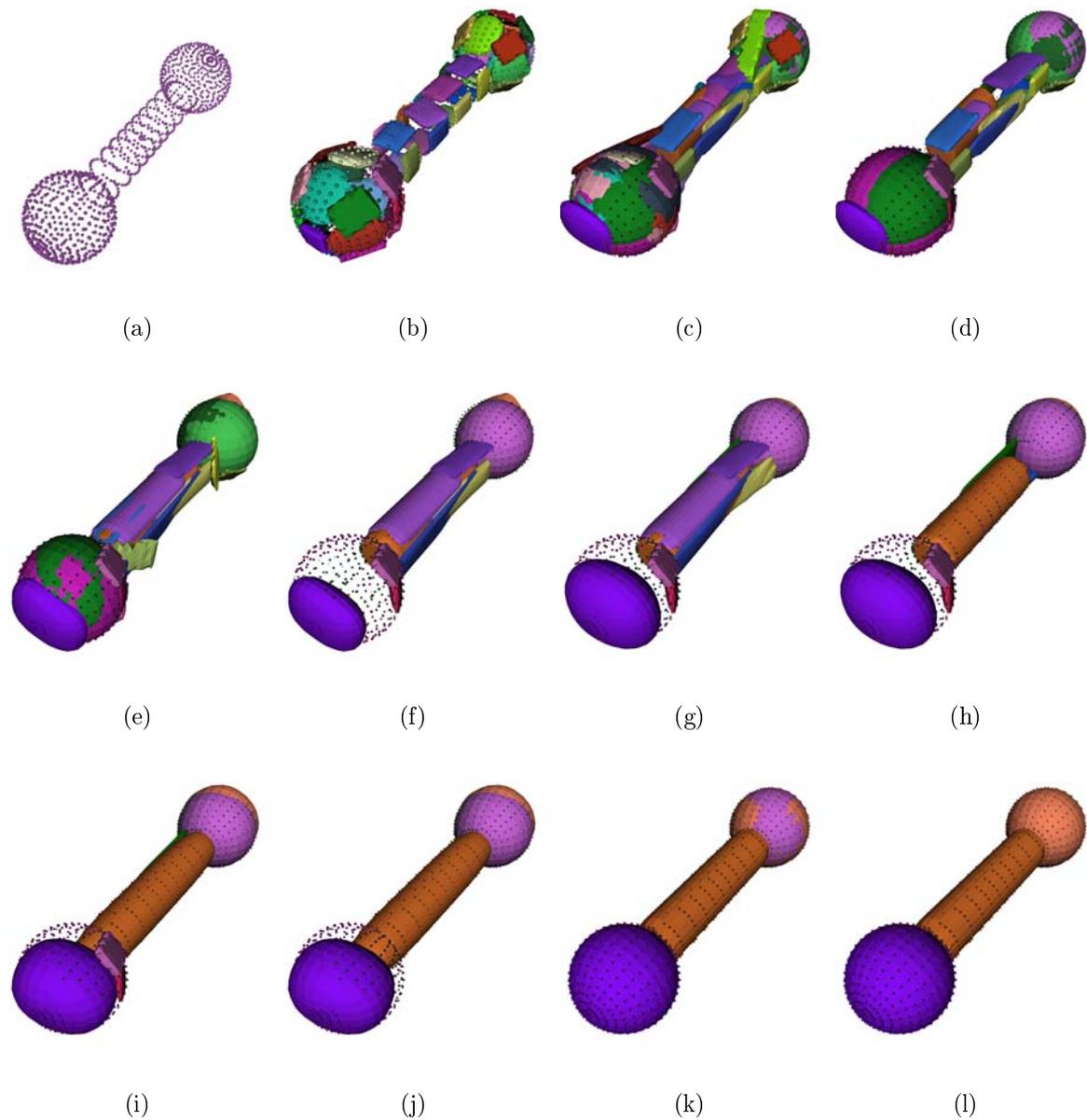


FIG. 4.3 – Étapes principales de la segmentation par la méthode de croissance de régions. (a) l'ensemble de points 3D. (b) Initialisation des germes (c)-(k) phase de croissance et sélection. (l) Le descripteur final (3 superellipsoïdes)

La figure 4.4 montre le résultat obtenu sur un autre objet synthétique. Ici encore

la description est fidèle au modèle utilisé pour obtenir les données tridimensionnelles et comporte trois superellipsoïdes.

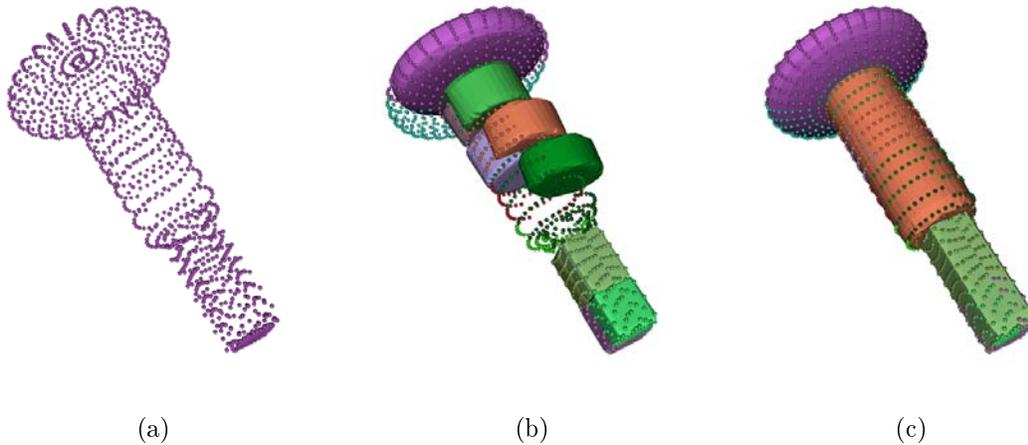


FIG. 4.4 – Résultat de la segmentation par croissance de régions sur l’objet Burin. (a) Données 3D. (b) Les germes. (c) Le descripteur final.

La figure 4.5 montre le résultat obtenu sur un objet synthétique à peine plus complexe, mais où la méthode segmentation ne parvient pas à obtenir le nombre attendu de primitives. En effet, dans le descripteur final 4.5c, le rectangle du dessous est décrit par deux superellipsoïdes une (jaune et une bleue) qui semble redondante. On pourrait alors croire à un problème de la méthode de sélection. En fait, il n’en est rien et il faut rapprocher ce problème de celui évoqué dans l’exemple présenté dans la figure 4.3 : on ne s’appuie pas sur les superellipsoïdes pour effectuer la sélection mais sur les sous-ensembles de points qu’elles modélisent. Ainsi, les deux superquadriques qui semblent être redondantes ne le sont en fait pas car les sous-ensembles de points qu’elles représentent sont relativement disjoints. Ce n’est donc pas la phase de sélection qui est à blâmer mais la phase de croissance. En effet, le principal problème que nous avons rencontré avec cet algorithme se situe au niveau de la recherche de points à ajouter aux régions lors de cette phase de croissance. Cette étape n’est pas adaptée à nos données irrégulières et non-organisées. Dans cet exemple, les régions jaune et bleu sont passées au stade *mature* avant d’avoir réellement fini d’occuper toute la partie de l’objet où elles se trouvaient et ont donc rendu l’algorithme de sélection inefficace.

#### 4.1.5 Conclusion

Comme on peut le voir, nous ne réussissons, avec cette méthode, la segmentation et la description que de quelques objets assez simples. Si les résultats ne sont pas aussi bons qu’escomptés, cela est dû en grande partie au type de nos données.

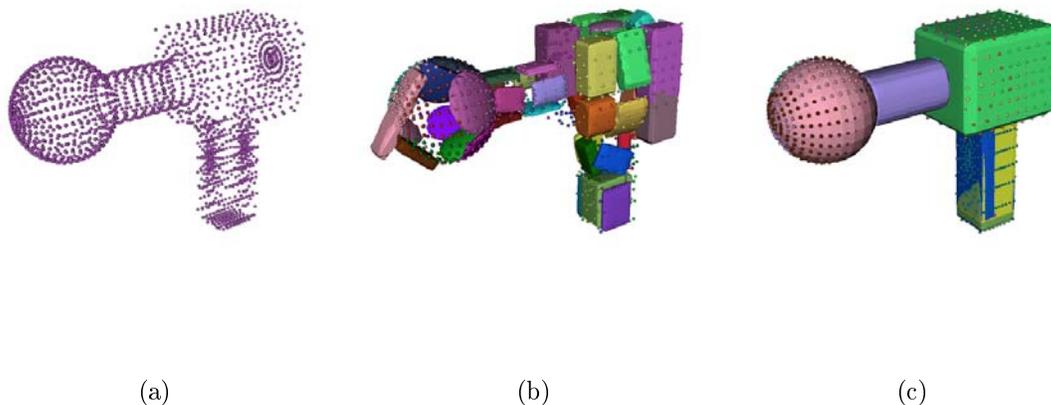


FIG. 4.5 – Résultat de la segmentation par découpage-fusion. (a) Données 3D. (b) Les germes. (c) Le descripteur final.

Il est important de souligner les deux problèmes principaux de cette méthode. Le premier est intrinsèque à la méthode même. Avec un algorithme de ce type, nous ne pourrons jamais être sûrs que chaque partie de l’objet 3D est bien représentée par une superellipsoïde dans le descripteur final, pour la simple raison que la phase d’initialisation n’assure en aucun cas qu’il y ait au moins un germe dans chaque partie de l’objet. Si une partie n’est pas représentée dès le début, elle ne le sera sans doute pas non plus à la fin de la segmentation, puisque la croissance des germes est faite justement dans l’optique de ne pas absorber d’autres parties de l’objet.

Le second problème est lié à notre type de données. Si cet algorithme était utilisable dans le cadre d’images de profondeur où l’organisation des points en grille 2D lui était particulièrement adaptée, ce n’est pas le cas de nos données que nous considérons très générales et donc quasiment toujours irrégulières. En effet, c’est la phase de croissance de régions qui s’est avérée difficile dans notre cas en raison de la difficulté de recherche des points à insérer.

Comme alternative à cet algorithme, et pour être mieux adapté à des données 3D plus générales, nous proposons une méthode originale basée sur une approche *global vers local* de type découpage-fusion.

## 4.2 Global à local : découpage-fusion (split and merge)

Le principe de la méthode découpage-fusion est classique, principalement en analyse d’image 2D. Comme son nom l’indique, il s’agit de la succession de deux étapes :

1. **Phase de découpage (split)** : les données vont être récursivement scindées en plusieurs parties suivant un critère d'homogénéité, jusqu'à ce que ce critère soit respecté pour toutes les régions ;
2. **Phase de fusion (merge)** : afin de réduire l'ensemble des trop nombreuses régions produites précédemment, certaines vont être regroupées sans pour autant que l'homogénéité des sous-parties n'en pâtisse ou que la distorsion du modèle aux données n'augmente significativement.

Dans cette partie, nous allons dans un premier temps détailler les deux phases de cet algorithme appliqué à notre objectif, puis nous présenterons les résultats que nous obtenons.

### 4.2.1 Phase de découpage (split)

Comme nous venons de l'évoquer, le but de cette étape est d'obtenir une partition des données, telle que chacune des sous-parties soit homogène suivant un critère. Mais, cette partition ne doit pas forcément être minimale puisque l'étape suivante va justement s'occuper de regrouper ces parties afin de rendre leur nombre minimal. En fait, on ne sait pas comment localiser directement les frontières des régions naturelles de l'objet<sup>5</sup>, soit on ne peut pas les calculer simplement ou en un temps raisonnable. Nous allons donc, de manière récursive, scinder les données arbitrairement en un nombre donné de sous-ensembles en projetant que cette approche va forcément, à un niveau ou à un autre, isoler dans un de ces sous-ensembles un morceau de région naturelle de l'objet. Par exemple, si vous prenez une photographie d'un de vos amis, que vous la coupez en deux, que vous prenez la ou une des parties contenant au moins une partie de sa tête et recommencez cette opération, vous arriverez tôt ou tard avec un morceau plus ou moins petit ne contenant pour une tolérance donnée, qu'une partie de sa tête et uniquement cela.

Ainsi, à la fin de cette étape, on obtient généralement une configuration remarquable :

1. des régions peu nombreuses et de taille importante au centre des parties naturelles de l'objet ;
2. des régions beaucoup plus nombreuses et de taille réduite dans les zones de frontières.

À la différence de l'algorithme de croissance de régions où il y avait un problème d'initialisation des germes, ici la précision est complètement guidée par les données. Si une partie d'objet est complexe beaucoup de régions seront produites localement pour celle-ci, mais ce ne sera pas forcément le cas dans les parties plus simples.

---

<sup>5</sup>Nous qualifierons de naturelles les régions ou parties de l'objet, celles pouvant être approximées par une seule superellipsoïde

Si on revient à notre problème, nous devons donc déterminer deux choses. La première est une manière de découper les données et la seconde est un critère d'homogénéité permettant de stopper la scission.

## Méthode de scission

Procédons par étapes pour la scission, en discutant d'abord le nombre de régions produites. Dans le cas d'un algorithme de découpage-fusion appliqué à de la segmentation d'image, il est habituel que ce découpage se fasse en quatre, afin de produire une structure de *quadtree*. Si on voulait donc faire quelque chose de similaire en 3D, il faudrait opter pour un découpage en huit et ainsi obtenir une structure d'*octree*. Mais cela ne serait pas forcément judicieux dans notre cas où ces régions vont être approximées par des superellipsoïdes. En effet, en coupant à chaque fois les données en huit sous-parties, les ensembles de points tridimensionnels à approximer seront la plupart du temps des surfaces très "ouvertes" ce qui rendra d'autant plus difficile l'approximation et produira aussi de très nombreuses régions. Pour notre objectif, nous choisirons donc de scinder les données en seulement deux parties, dans le but d'ouvrir les surfaces le moins possible.

Comme nous l'avons déjà évoqué, si nous choisissons d'utiliser ce type d'algorithme c'est parce que nous ne savons pas estimer directement les frontières entre les différentes parties naturelles de l'objet. La méthode de découpage peut rester simple, relativement arbitraire. C'est l'association des étapes découpage et fusion qui se charge de déterminer les frontières complexes des parties de l'objet. Ayant décidé de scinder les données en seulement deux parties, le plus simple est donc d'utiliser un plan de coupe. C'est-à-dire, pour une région donnée  $R$  et un plan  $P$ , les régions  $R_1$  et  $R_2$  résultantes de la scission de  $R$  par  $P$  seront formées de l'ensemble des points de  $R$  sur  $P$  ou à gauche de  $P$  pour la première, et strictement à droite de  $P$  pour la seconde. N'ayant pas d'indication sur comment placer ce plan ou comment l'orienter, nous utiliserons le plan le plus naturel, c'est-à-dire celui qui passe par le centre de gravité de l'ensemble des points tridimensionnels et est orthogonal à son axe principal (Figure 4.6). Notons que ce plan de coupe est "naturel" si les données sont régulièrement réparties, ce qui n'est pas forcément le cas. Mais cela n'a pas beaucoup d'importance, car cela va être compensé par l'étape de fusion.

La méthode du choix du plan de coupe est importante et mériterait peut être plus d'approfondissement. En effet, s'il n'est pas sensé altérer le résultat final de la segmentation, ce plan de coupe a une importance cruciale dans la progression de l'algorithme et du nombre d'étapes nécessaires pour aboutir à la segmentation. Cependant, nos données non-organisées et irrégulières ne nous laissent pas beaucoup d'options. Par exemple, les méthodes qui se basent sur l'analyse des courbures ne peuvent pas être envisagées si on ne veut pas passer, comme dans la méthode de croissance de région, par l'étude du voisinage des points 3D.

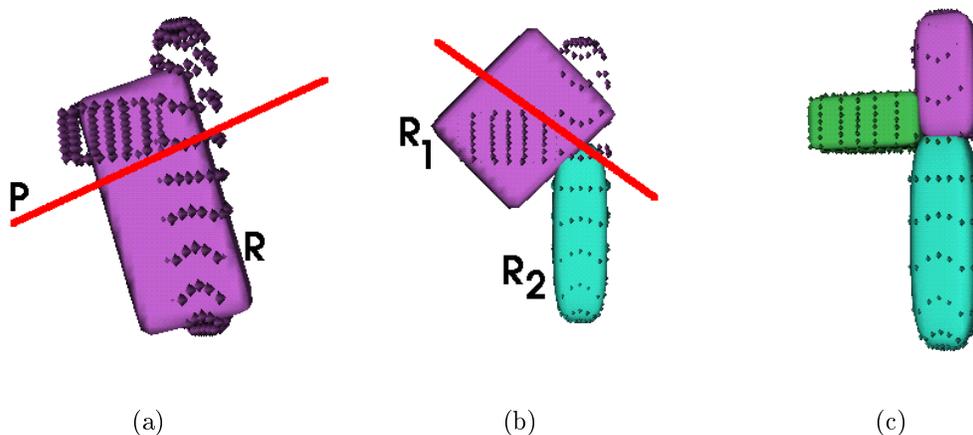


FIG. 4.6 – Exemple de phase de découpage

### Critère d'homogénéité

Afin de stopper cette scission récursive du nuage de points, nous devons établir un critère d'arrêt.

D'abord, il y a le critère de taille : si une région a un nombre insuffisant de points pour être approximée par une superellipsoïde, cela ne sert à rien de descendre plus bas (i.e. de le découper). Par expérience, on peut affirmer qu'un ensemble de seulement dix points n'est pas vraiment approximable par une superquadrique avec notre technique. Donc, dans le cas d'un ensemble inférieur à vingt points, par exemple, il faut stopper la phase de découpage.

Bien entendu, il y a aussi un critère d'homogénéité de la région. Ce critère va arrêter la phase de découpage pour une région avant que celle-ci n'ait atteint la taille minimum requise.

Comme nous ne savons pas ce qui va se passer dans la seconde étape de l'algorithme (i.e. fusion), si telles ou telles régions vont fusionner ou pas, chaque partie que nous obtenons dans cette première étape doit être approximable par une superellipsoïde suivant le niveau de distorsion  $\Delta$  désiré. Cela doit être pris en compte dans le critère d'homogénéité. En fait, on dira qu'une région, c'est-à-dire un ensemble de points tridimensionnels, est homogène s'il est possible de l'approximer par une superellipsoïde avec une tolérance donnée. Si la tolérance d'approximation est  $\tau_s$ , et suivant la notation de l'équation 3.1 de la page 44, une région est homogène si et seulement si :

$$\Delta(\mathbf{a}) \leq \tau_s$$

D'autres formes de critères pourraient être prises en compte, notamment la différence de qualité d'approximation d'une région  $R$  et de celle des régions résultantes de la scission

de  $R : R_1$  et  $R_2$ . Une région serait dite homogène si les approximations des régions  $R_1$  et  $R_2$  ne sont pas meilleures que celle de  $R$ .

## Algorithme de découpage

Ainsi, en prenant pour région initiale la totalité des données tridimensionnelles, si une région donnée est approximable par une superellipsoïde avec une distorsion inférieure à un seuil donné, ou est de taille inférieure à vingt points, la phase de découpage s'arrête, sinon la région est scindée en deux régions par le plan de coupe passant par son centre de gravité et orthogonal à son axe d'inertie, et le procédé est répété récursivement sur les deux régions résultantes (algorithme 4.2.1).

**Ensure:** Scission(REGION  $R$ , REGION  $R_1$ , REGION  $R_2$ )

$P = \text{PlanDeCoupe}(R)$

**for all**  $p \in R$  **do**

**if** (AGauche( $x, P$ )) **then**

$R_1 = R_1 + p$

**else**

$R_2 = R_2 + p$

**end if**

**end for**

**Ensure:** Split(REGION  $R$ )

**if** ( ( $\text{taille}(R) \geq 20$ ) **et** (  $\nexists \mathbf{a}$  tel que  $\Delta(\mathbf{a}) \leq \tau_s$  ) ) **then**

  Scission( $R, R_1, R_2$ );

  Split( $R_1$ );

  Split( $R_2$ );

**end if**

### 4.2.2 Phase de fusion (merge)

L'étape de découpage nous a permis d'obtenir un partitionnement, tel que chacune des parties peut être approximée avec précision par une seule superquadrique. Mais le nombre de régions créé est loin d'être optimal et de correspondre aux parties naturelles de l'objet traité. Cela vient du fait que le plan de coupe ne tient pas compte des frontières naturelles de cet objet. En revanche, toute région ne fait partie que d'une partie naturelle. L'étape de fusion va permettre de regrouper les régions qui appartiennent à la même partie naturelle mais qui ont été séparées lors de la phase de découpage.

C'est donc l'étape de découpage qui va déterminer la précision de la segmentation. En effet, durant l'étape de fusion, aucune nouvelle frontière entre les régions ne va apparaître, au contraire certaines vont disparaître. C'est donc le caractère du critère d'homogénéité de l'étape précédente ( $\tau_s$ ) qui déterminera la précision maximum de la segmentation obtenue

par cette méthode.

Afin de mener à bien ces regroupements, nous avons besoin de deux choses. La première est un critère permettant de déterminer si deux régions méritent d'être regroupées, et la seconde est une stratégie nous indiquant quels couples de régions peuvent être candidats à la fusion (figure 4.7).

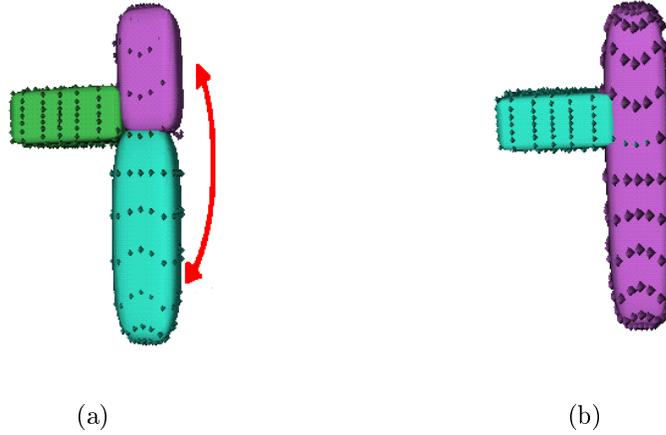


FIG. 4.7 – Exemple de phase de fusion

## Critère de regroupement

Comme pour la phase précédente, et à plus forte raison même, certains des ensembles de points qui vont être créés dans cette étape par regroupement, seront ceux qui formeront la segmentation finale des données tridimensionnelles. Nous devons donc être en capacité de représenter tous ces ensembles par une superellipsoïde pour un taux de distorsion donné. Cela doit faire partie du critère de regroupement. On ne peut regrouper deux régions que si pour l'union des ensembles de points qui les composent, il existe un ensemble  $\mathbf{a}$  de paramètres d'une superellipsoïde, tel que :

$$\Delta(\mathbf{a}) < \tau_m$$

Les seuils de distorsions  $\tau_s$  et  $\tau_m$  utilisés respectivement dans la phase de découpage et celle de fusion sont idéalement les mêmes ( $\tau_s = \tau_m$ ), puisqu'ils représentent tous deux la distorsion maximale désirée pour toute superellipsoïde du descripteur final.

Comme pour le critère d'homogénéité, une prise en compte des itérations précédentes est aussi envisageable. Ainsi le regroupement de deux régions  $R_1$  et  $R_2$  en une seule région  $R$  n'aurait lieu que si la qualité de modélisation de  $R$  est aussi bonne que celle  $R_1$  et  $R_2$ . Notons que ce critère ainsi défini pourrait autoriser moins de regroupement que nécessaire.

Si par exemple,  $R_1$  et  $R_2$  sont modélisées presque parfaitement, il faudrait que  $R$  le soit aussi.

En plus de la qualité d'approximation, d'autres considérations doivent être faites dans le cas du critère de regroupement. Il s'agit en fait d'un cas particulier à prendre en compte et c'est ce que nous allons voir.

Supposons que nous essayons de déterminer si deux tiges de même taille, parallèles entre elles ou étant jointes bout à bout et formant un angle droit doivent être regroupées. Il est tout à fait possible que l'approximation, au sens ou nous l'avons défini, de ces deux objets par une seule superellipsoïde soit de bonne qualité, puisque la surface d'une superellipsoïde de type "plaque" peut être, au même moment, relativement près des deux ensembles de points (figure 4.8). La seule chose que nous pouvons reprocher à ce genre d'approximation est qu'une grande partie de la superellipsoïde ne modélise aucune donnée.

C'est pourquoi dans le critère de regroupement nous devons tenir compte de l'augmentation du volume lors du passage d'une modélisation par deux superellipsoïdes à une seule superellipsoïde. Ainsi, si cette augmentation est supérieure à un certain facteur (1.5 par exemple), les régions ne seront pas regroupées quelle que soit la qualité d'approximation.

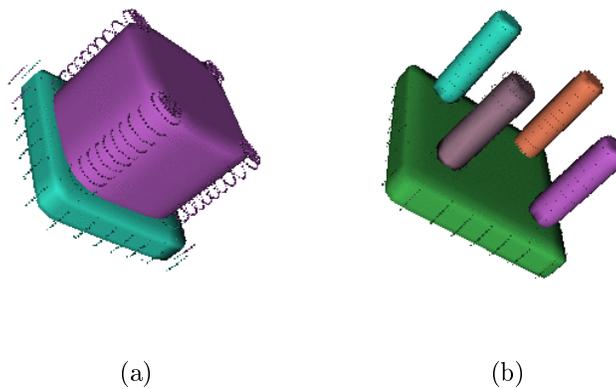


FIG. 4.8 – Exemple de fusion à éviter. Il n'est pas possible d'éviter ce genre de cas uniquement en se basant sur la distorsion, car les points des tiges sont dans les deux cas relativement près de la surface, c'est pourquoi nous utilisons un contrôle de l'augmentation du volume.

## Stratégie de recherche des couples candidats

Il n'est pas envisageable pour cette phase de fusion de tester tous les couples possibles de régions produites par l'étape de découpage. D'une part, parce que ce serait trop long,

et d'autre part, il n'y a pas de raison de faire ce test pour certains couples trop éloignés par exemple, ce qui dans le cas contraire pourrait produire des erreurs. Nous avons donc besoin de définir une stratégie sur les choix des couples à tester.

La stratégie de regroupement est relativement importante puisque, pour garantir la compacité du modèle, il est important que tous les couples faisant partie d'une région soient pris en compte, et d'autre part l'ordre de sélection va privilégier certains regroupements plutôt que d'autres. Par exemple, un couple  $R_1R_2$  est sélectionné. Ce couple satisfait le critère de regroupement, il y a donc fusion en une région  $R_3$ . Il se peut alors que plus aucune région ne puisse être groupée avec  $R_3$ , alors que si  $R_1$  et  $R_2$  n'avaient pas fusionné, un nombre important de regroupements auraient pu être fait par la suite avec  $R_1$  ou  $R_2$ .

Nous n'avons pas trouvé de solution parfaite à ce problème de stratégie de regroupement et nous nous sommes contentés d'une méthode plus ou moins arbitraire qui fournit de bons résultats en pratique.

Nous commençons par sélectionner le sous-ensemble de points qui est le mieux approximé par une superellipsoïde. Ensuite, nous testons la fusion de cette région avec l'ensemble de ses régions voisines et regroupons définitivement le couple de sous-ensembles le mieux modélisé.

Nous ne considérons pas ici uniquement le voisinage direct des ensembles de points, mais les "voisins des voisins" sont aussi pris en compte. Un voisinage que nous pourrions appeler de second ordre, donc. Cette extension du voisinage permet d'empêcher l'arrêt prématuré de la phase de fusion.

Comme nous l'avons déjà fait remarquer, cette stratégie ne résout en rien les problèmes que peut entraîner la fusion. Néanmoins, elle donne des résultats satisfaisants en pratique.

### 4.2.3 Résultats

Nous présentons ici les résultats obtenus grâce à la méthode de découpage-fusion.

La figure 4.9 détaille la méthode de segmentation en douze étapes comme on l'avait fait pour la méthode de croissance de régions sur le même objet (figure 4.3). Ici non plus, les étapes présentées ne sont pas exhaustives mais nous avons sélectionné les plus significatives suivant cette vue. La phase de découpage s'étale de la figure 4.9b à la figure 4.9h. Ensuite, il s'agit donc des étapes de fusion. À l'instar du résultat de la méthode précédente, le descripteur final est constitué de trois superellipsoïdes comme le modèle synthétique original.

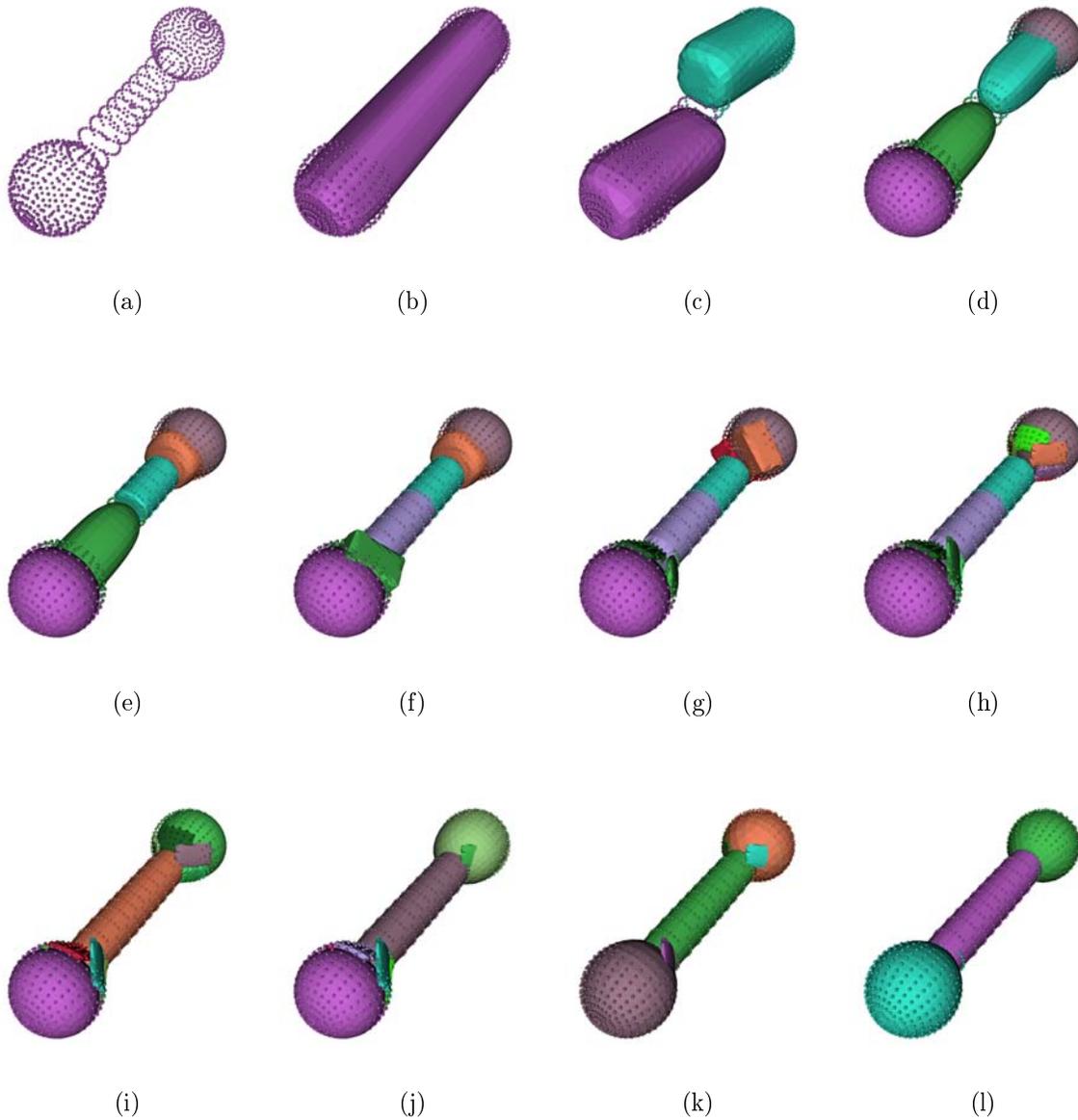


FIG. 4.9 – Étapes principales de la segmentation par la méthode de découpage-fusion. (a) l'ensemble de points 3D. (b)-(h) phase de découpage. (h)-(l) phase de fusion. (l) descripteur final (3 superellipsoïdes)

Les figures 4.10 et 4.11 montrent les résultats de cette nouvelle méthode sur les objets synthétiques que nous avons déjà testés avec la méthode de croissance de régions (figure 4.4 et 4.5 de la page 68). Dans ces deux cas, la segmentation et les descripteurs finaux sont tout à fait conformes à ce qu'on peut en attendre. Pour chacun d'entre-eux, l'assemblage de superellipsoïdes est similaire à celui de l'objet d'où l'ensemble de points provient. Ce n'était pas le cas du second exemple décrit avec la méthode de croissance de régions.

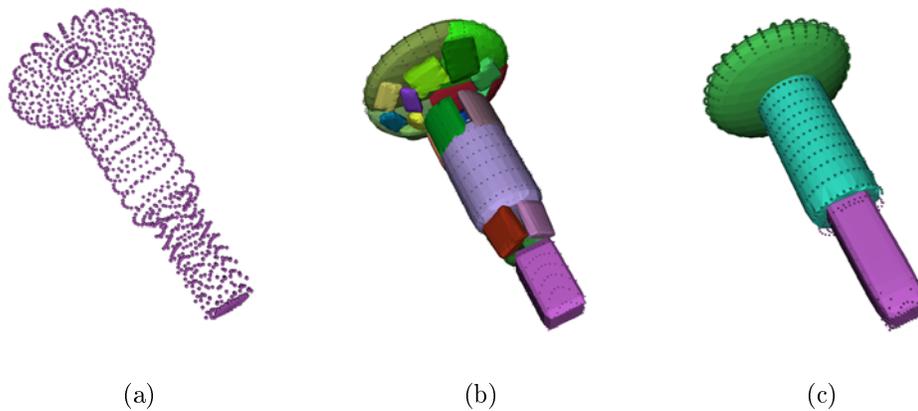


FIG. 4.10 – (a) Données 3D Burin. (b) A la fin du split. (c) descripteur final.

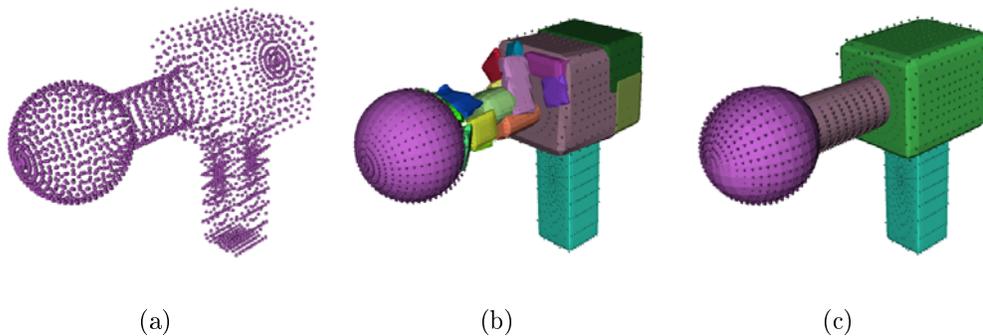


FIG. 4.11 – (a) Données 3D Pistolet. (b) A la fin du split. (c) Le descripteur final.

Cette méthode de segmentation s'avère plus adaptée à notre type de données et nous a donc permis de décrire des objets plus complexes. Les figures 4.12 et 4.13 montrent les résultats obtenus sur deux objets de type chaise. La première provient toujours d'un objet synthétique construit à partir de superellipsoïdes. Le résultat est tout à fait satisfaisant, et la description se fait par un assemblage de superellipsoïdes similaire à celui ayant servi pour obtenir les données (figure 4.12c). La seconde chaise (figure 4.13) est issue d'un objet maillé que nous avons échantillonné. Comme on peut le voir, le résultat est globalement satisfaisant. Un problème persiste au niveau du dossier. Les barreaux qui le constituent ont fusionné pour donner deux plaques. Le problème de cette fusion aurait normalement du être pris en charge par le contrôle du volume que nous imposons lors des étapes de fusions. Il s'avère que dans ce cas les barreaux sont à la fois fins et rapprochés ce qui les rend difficilement différenciables d'une simple plaque et le contrôle du volume nécessaire serait beaucoup trop strict pour que la segmentation aboutisse convenablement dans des cas plus généraux.

Cette méthode donne donc des résultats tout à fait satisfaisants à la fois en terme

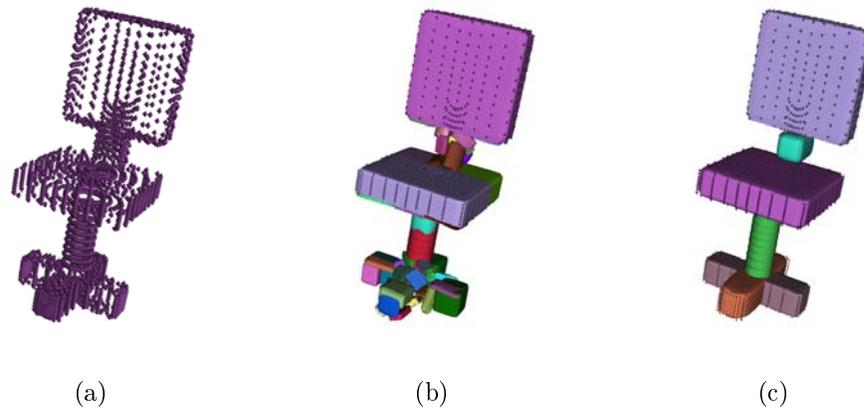


FIG. 4.12 – (a) Données 3D Chaise. (b) À la fin du split. (c) Le descripteur final.

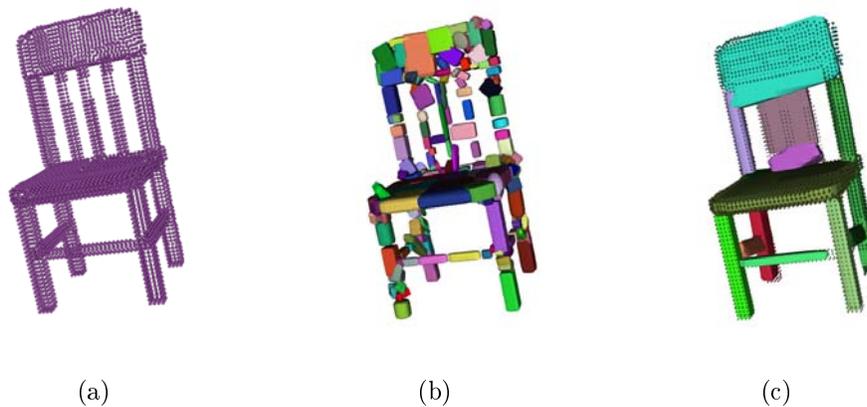


FIG. 4.13 – (a) Données 3D Chaise2. (b) À la fin du split. (c) Le descripteur final.

de segmentation et de représentation pour la visualisation lorsqu'il s'agit de décrire des objets dont les parties sont très proches de superellipsoïdes (c'est le cas de tous objets précédemment cités). Les figures 4.14 et 4.15 montrent l'application de notre méthode sur des objets dont les parties principales sont plus complexes qu'une simple superellipsoïde. Comme on peut le voir, la segmentation est correcte et chaque partie principale de l'objet est représentée par une superellipsoïde et une seule. En revanche, en ce qui concerne la visualisation, on remarque une certaine limite de description des superellipsoïdes. En effet, la représentation est assez grossière. On ne peut pas accuser pour cela une mauvaise méthode d'approximation, les sous-parties ne sont tout simplement pas modélisables avec une superellipsoïde de manière précise. Il faut tout de même garder à l'esprit que nous avons essayé d'obtenir exactement une superellipsoïde par partie principale, ce qui est est peu. Tout le problème est bien entendu de concilier compacité et information sémantique du descripteur d'une part et précision de la représentation d'autre part. Dans ce travail, nous avons surtout privilégié la segmentation sur la représentation.

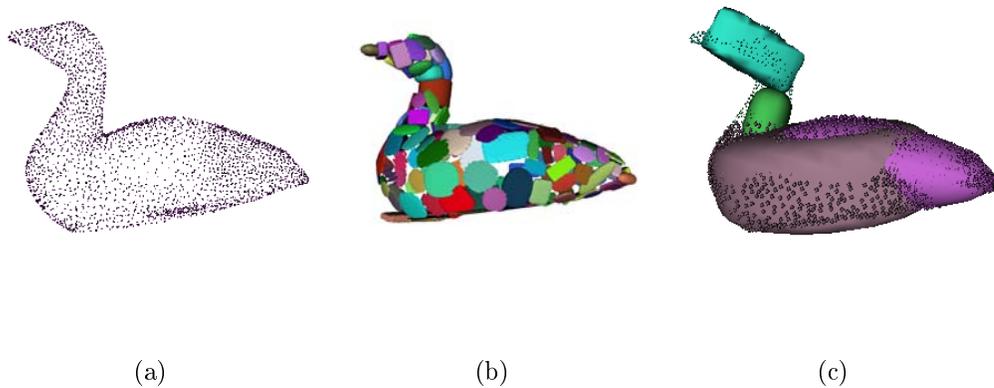


FIG. 4.14 – (a) Données 3D Canard. (b) À la fin du split. (c) Le descripteur final.

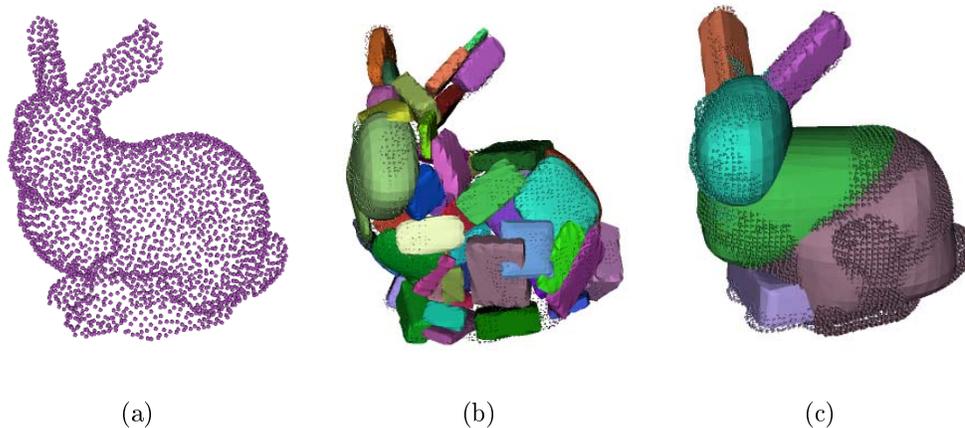


FIG. 4.15 – (a) Données 3D Bunny . (b) À la fin du split. (c) Le descripteur final.

#### 4.2.4 Structure du descripteur

À la différence de la méthode par croissance de régions, le descripteur final n'est pas simplement un ensemble non-organisé de primitives. En effet, durant tout l'algorithme nous devons conserver les relations d'adjacences entre les superellipsoïdes afin de pouvoir, durant la seconde phase, procéder à la fusion de régions voisines le cas échéant. Ces relations topologiques sont toujours présentes dans le descripteur final. Ainsi, le résultat de cette méthode de segmentation est en fait un graphe dont les sommets sont des superellipsoïdes et les arcs indiquent une relation d'adjacence entre celles-ci (fig. 4.17).

Cette structuration est bien sûr un avantage puisqu'elle charge le descripteur d'une information sémantique beaucoup plus importante qu'un ensemble non-organisé de superellipsoïdes. Il serait sans doute possible de déterminer ce graphe pour un descripteur issu de la méthode *region growing*, mais cela serait plus difficile surtout que dans ce cas-là, les

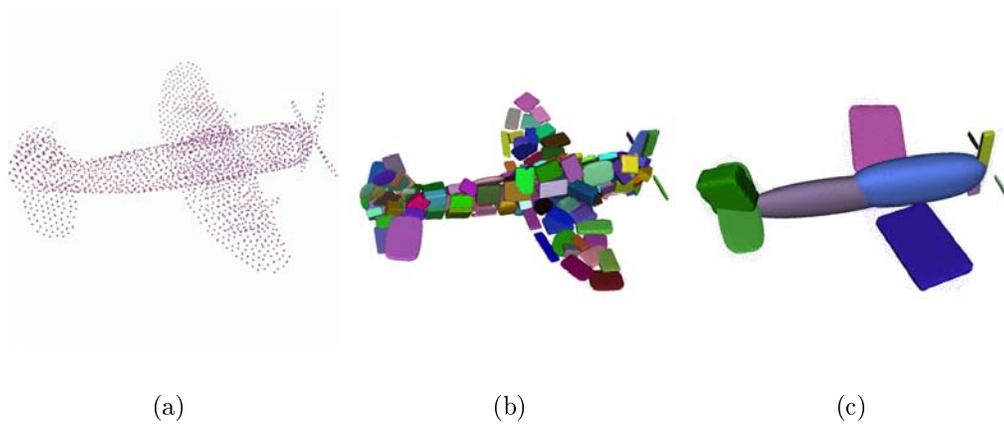


FIG. 4.16 – (a) Donnée 3D Avion. (b) À la fin du split. (c) Le descripteur final.

régions peuvent se chevaucher.

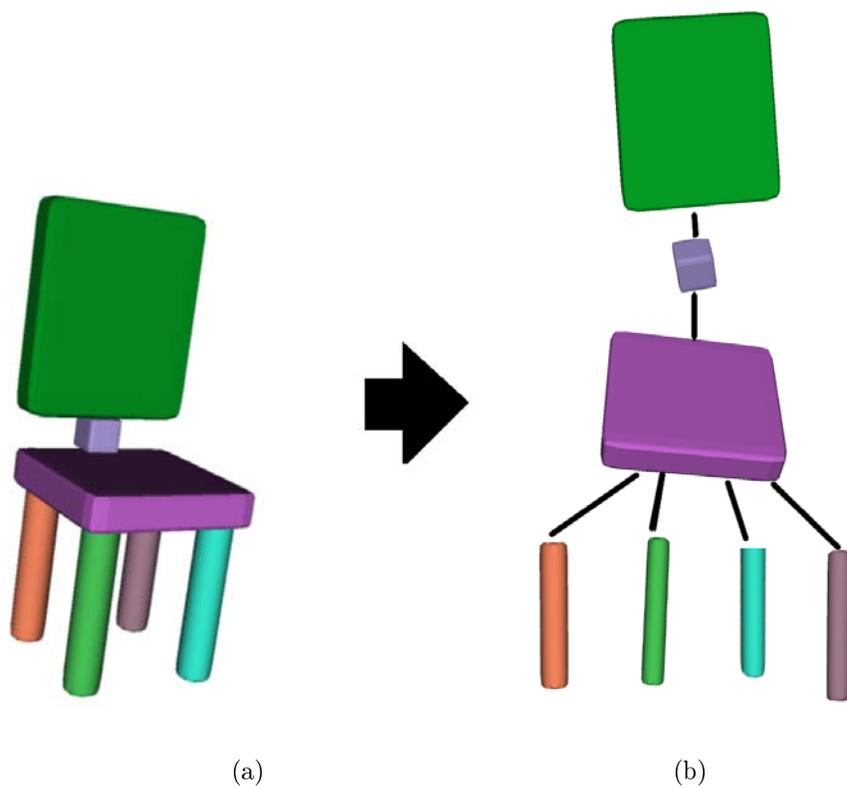


FIG. 4.17 – Le descripteur est structuré par un graphe d'adjacence.

### 4.2.5 Commentaires

Au vu des résultats, cette méthode de segmentation est mieux adaptée aux données 3D générales que nous envisageons, que la méthode de croissance de régions précédemment exposée. Dans cette approche découpage-fusion nous ne faisons référence à aucune connaissance *a priori* sur l'organisation ou la régularité de l'ensemble de points (tel que des relations de voisinage par exemple).

De plus, un avantage très important de cette approche est le fait qu'elle permette de garder les relations topologiques entre les différentes régions, et donc entre les superellipsoïdes.

## 4.3 Conclusion

Dans ce chapitre, nous avons présenté deux méthodes de segmentation. Si la première, de type croissance de régions, n'a pas abouti, la seconde, de type découpage-fusion, donne de bons résultats au niveau de la segmentation des données tridimensionnelles. Le descripteur que nous obtenons finalement est un ensemble de superellipsoïdes structuré par un graphe d'adjacence.

Néanmoins, si les résultats sont satisfaisants d'un point de vue de la segmentation, ils permettent aussi de mettre en évidence les limites des superellipsoïdes en ce qui concerne leurs capacités de description. En effet, tant que les objets reconstruits sont des objets de synthèse modélisés avec des superellipsoïdes, la reconstruction est fidèle. En revanche, en ce qui concerne les objets réels, les différentes parties de l'objet obtenues après décomposition ne peuvent pas être modélisées avec précision par une seule superellipsoïde. Il faut tout de même noter que nous nous sommes, dans cette partie et dans les résultats, appliqués à montrer l'efficacité de notre méthode de segmentation et que cela n'est pas forcément compatible avec une approximation précise d'un objet à l'aide de superellipsoïdes.

# Conclusion

Dans cette première partie, nous avons présenté une méthode originale de segmentation et de reconstruction de données tridimensionnelles, échantillon de points de la surface d'un objet. Nous nous sommes appliqués à considérer des données tridimensionnelles très générales : un ensemble de points 3D non organisé.

La première étape de cette modélisation a été l'approximation d'un nuage de points par une seule superellipsoïde.

Dans un premier temps, nous avons été amenés à déterminer une méthode de mesure de la distance d'un point à sa surface. Nous avons proposé d'utiliser la distance euclidienne radiale qui non seulement a un comportement plus proche de la distance euclidienne mais a aussi l'avantage d'être indépendante par rapport à la taille de la superellipsoïde considérée (contrairement à la mesure la plus communément utilisée). Cette normalisation de la distance est importante car nous avons besoin de comparer des mesures sur des superellipsoïdes de tailles diverses durant la segmentation. L'approximation proprement dite est réalisée au sens des moindres carrés suivant la méthode de Solina [SB90]. Après une étude de différentes approches de régression, nous avons décidé d'utiliser, comme prévu par la méthode originale, l'algorithme de Levenberg-Marquardt.

La segmentation est la deuxième et dernière étape nous ayant permis d'aboutir à notre descripteur. Le descripteur et ses superellipsoïdes font partie intégrante de cette segmentation et sont obtenus directement par celle-ci.

Nous avons d'abord tenté d'adapter une méthode de croissance de régions initialement prévue pour les images de profondeur proposée par Leonardis [LJS97]. Cette méthode ne s'est pas avérée efficace dans notre cas où les données ne sont pas régulières comme peuvent l'être ce type de données et de ce fait la croissance des germes était difficile.

Nous avons alors proposé une méthode originale basée sur un algorithme de découpage-fusion. Les données sont scindées récursivement jusqu'à ce que l'erreur d'approximation par une superellipsoïde de chacune des sous-parties soit en-dessous d'un certain seuil. La coupe se fait suivant un plan, choisi entre autres pour la simplicité de son calcul, passant par le barycentre de l'ensemble de points et orthogonal à son axe d'inertie. Ensuite, les régions vont être regroupées jusqu'à ce que plus aucune union ne puisse être faite. Celle-ci ne fait donc pas appel à la recherche du voisinage de points et produit de bien meilleurs

résultats.

Finalement, le descripteur que nous obtenons est constitué d'un ensemble de superellipsoïdes structuré par un graphe d'adjacence.

Cette première partie détaille une chaîne complète allant de la segmentation de points non organisés jusqu'à l'obtention d'un descripteur 3D compact.

Au début de cette étude, nous visions un ensemble assez large de possibilités d'exploitation pour notre descripteur, nous pensions à des applications de type visualisation basse résolution, visualisation progressive ou multi-résolutions pour les réseaux bas débits, compression et comparaison d'objets pour une application à l'indexation. Malgré la capacité de description très importante des superellipsoïdes, relativement à leur faible nombre de paramètres, la pratique a montré qu'il était difficile de concilier à la fois un faible nombre de primitives et une distortion peu élevée ou du moins une représentation suffisamment précise de la surface pour des applications de visualisation en moyenne et haute résolution.

Si, en l'état, le descripteur est donc limité pour les applications de type visualisation, il est en revanche porteur d'information sémantique forte. En plus, ce descripteur est très compact. Cela rend donc crédible son utilisation dans une application de type indexation d'objets 3D. Ainsi, nous allons voir dans la seconde partie de ce mémoire une méthode de comparaison d'objets à travers ce descripteur et donc rendre son exploitation possible pour l'indexation.

## Deuxième partie

# Utilisation du descripteur d'objets pour l'indexation et la comparaison d'objets 3D



# Introduction

L'augmentation du volume des données tridimensionnelles et l'engouement pour leur utilisation dans tous les domaines font de l'indexation d'objets tridimensionnels une problématique réellement d'actualité. Il s'agit, à l'instar des mots clefs associés à un article ou une page web, d'adjoindre un descripteur à chaque modèle tridimensionnel d'une base de données. Bien entendu, celui-ci doit rester compact pour ne pas augmenter la taille déjà importante des données mais doit aussi pouvoir être suffisamment discriminant pour pouvoir distinguer des objets 3D entre-eux car c'est lui qui sera utilisé pour comparer les objets afin de répondre aux requêtes des utilisateurs.

Dans la partie précédente de ce mémoire, nous avons proposé une méthode de description d'objets 3D définis par un ensemble de points échantillonnés sur sa surface. Notre approche de description est basée sur une segmentation des données d'où résulte un ensemble de superellipsoïdes structuré à l'aide d'un graphe d'adjacence. Cette structure est porteuse d'une information sémantique importante permettant de caractériser un objet et de le comparer à d'autres. De plus, ce descripteur est particulièrement compact et c'est pourquoi nous proposons dans cette partie de l'utiliser dans une application de type indexation. Ayant montré dans la première partie comment construire ce descripteur, il ne nous manque pour cette application, qu'une méthode permettant de comparer les descripteurs entre-eux, et par là, les objets auxquels ils sont associés. C'est ce que nous allons exposer dans cette partie.

Les deux parties du mémoire peuvent être abordées de manière relativement indépendantes. Ainsi, nous considérerons notre descripteur sans supposition sur la méthode ayant permis son obtention. Il pourrait résulter de la méthode que nous avons présentée dans la première partie, d'une tout autre méthode automatique, ou bien même, comme ce dernier est tout à fait intuitif, assemblé manuellement par l'auteur d'une requête à une base de données.

Cette seconde partie du mémoire s'articule autour de trois chapitres.

Nous présentons dans le premier chapitre un aperçu des méthodes existantes en indexation et comparaison d'objets 3D. Bien sûr, notre démarche n'est pas habituelle puisque

nous avons déjà le descripteur qui nous servira à l'indexation. Ce tour d'horizon n'aboutira donc pas sur le choix d'une méthode. L'ambition de chapitre est plutôt de replacer notre cas dans le contexte de l'indexation 3D et des approches très différentes dont celle-ci fait l'objet.

Le second chapitre est consacré aux méthodes associées à la comparaison de graphes. Nous présentons dans un premier temps les notions traditionnelles de la théorie des graphes telles que la famille des isomorphismes. Nous mettons alors en évidence que ces méthodes ont des critères souvent trop stricts pour une application comme la nôtre. Une méthode moins classique est étudiée : la distance d'édition. Celle-ci propose de calculer la similarité entre deux graphes en cumulant le coût de différentes opérations modifiant les graphes comparés et permettant ainsi de passer de l'un à l'autre.

Dans le troisième chapitre, la distance d'édition est utilisée pour comparer nos descripteurs. Nous commençons par définir les opérations de modification des graphes nécessaires puis nous en fixons le coût. Comme nous le verrons, les opérations sont choisies en tenant compte de notre méthode de segmentation et des différences de description non significatives qu'elle pourrait produire sur des objets assez proches. Bien sûr, la comparaison n'en est pas moins généraliste, et peut être utilisée sur tout objet caractérisé par notre descripteur quelle que soit la méthode dont ce dernier résulte. À la fin de ce chapitre, des résultats de comparaison et d'indexation d'objets 3D sont présentés.

# Chapitre 5

## État de l’art sur l’indexation et la comparaison d’objets 3D

Dans ce chapitre, nous faisons un tour d’horizon des différentes méthodes d’indexation et de comparaison d’objets tridimensionnels. Dans la plupart des cas, les objets dont il va être question ici sont représentés sous forme de maillages. Puisque ce format est de loin le plus utilisé pour la modélisation de surfaces, et donc le plus courant dans les bases de données d’objets 3D, il est bien légitime que les travaux portant sur l’indexation lui accordent une place si importante.

Le domaine de l’indexation 3D peut être abordé par des approches très variées. Nous allons dans ce chapitre en exposer trois principales qui, comme nous allons le voir, ont chacune leurs spécificités :

- l’approche statistique ;
- l’approche multi-vues ;
- l’approche structurale.

Bien sûr, toutes ces approches doivent respecter les contraintes d’invariance relativement aux transformations géométriques, et également être robustes par rapport au niveau de modélisation de l’objet 3D (niveau de facétisation par exemple, puisqu’il s’agit le plus souvent de maillages), aux petites modifications et aux bruits divers.

Nous décrirons successivement les particularités de ces trois différentes approches.

### 5.1 Approches statistiques

L’approche statistique de l’indexation 3D consiste à identifier les objets grâce à une ou plusieurs distributions de descripteurs de formes. Trouver la distance entre deux objets 3D se traduit alors par comparer les distributions qui leur correspondent (i.e. comparaison

d'histogrammes).

Suivant le descripteur de forme utilisé, on peut distinguer deux sous-catégories à cette première approche.

### 5.1.1 Approches statistiques locales

L'approche statistique locale repose sur l'étude des caractéristiques locales de la forme de l'objet à décrire. Typiquement, la courbure calculée en tout point de la surface de l'objet sera utilisée comme descripteur de forme et c'est la distribution de ces courbures qui servira à caractériser l'objet.

Le descripteur par spectre de forme 3D (SF3D) inclus dans la norme MPEG-7 entre dans cette catégorie et vise à décrire les objets 3D représentés sous forme de maillages 3D au format VRML.

Koenderink [Koe90] a proposé un index de forme permettant de caractériser localement la géométrie d'une surface (Figure 5.1). Si on considère un point  $p$  d'une surface  $C^2$ -différentiable et  $k_p^1$  et  $k_p^2$  les courbures principales en  $p$ , alors l'index est défini par :

$$I_p = \frac{1}{2} - \frac{1}{\pi} \arctan \left( \frac{k_p^1 + k_p^2}{k_p^1 - k_p^2} \right)$$

Notons que  $I_p \in [0, 1]$ , qu'il n'est pas défini pour les surfaces planes et est invariant aux translations, rotations et aux homothéties.

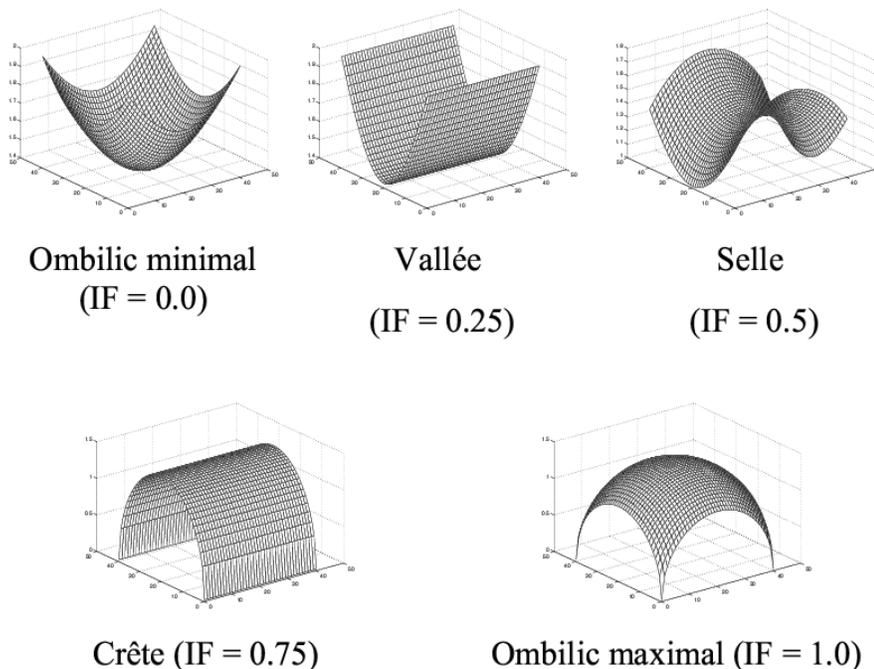


FIG. 5.1 – Indice de Koenderink des différentes formes élémentaires de surface

Le SF3D n'est autre qu'un vecteur de taille  $N$  où chaque composante  $k$  cumule l'aire relative à la surface totale de l'objet de toutes les facettes dont  $I_p \in [\frac{k-1}{N}, \frac{k}{N}]$ , c'est-à-dire la distribution de l'index de forme de Koenderink pour l'objet 3D maillé considéré.

Par exemple, dans [TZ02], Zaharia et Prêteux utilisent le SF3D pour indexer des objets 3D maillés. Ils décrivent notamment comment obtenir une estimation fiable et précise des courbures nécessaires au descripteur en approximant localement la surface par une quadrique. Ils montrent aussi que le SF3D n'est pas robuste aux représentations topologiques multiples (niveau de facétisation notamment), et que son utilisation doit par conséquent être précédé de traitements de filtrage lourds. Ils proposent alors une autre approche, par transformée cette fois-ci et dérivée de Hough 3D : DH3DO.

### 5.1.2 Approches statistiques globales

À la différence des approches statistiques locales, d'autres approches vont s'intéresser à la mesure de propriétés géométriques globales de l'objet considéré.

Il s'agit alors de calculer des moments à différents ordres comme par exemple dans les travaux de Firooz et Hall [AH80] ou ceux de Chong-Huan et Hon-Son [LD89]; ou bien, de la même manière que dans les approches statistiques locales, des distributions de mesure comme la distance entre deux points de la surface de l'objet, la surface des triangles formés par trois points de l'objet, etc.

Pour cette dernière approche, on peut se référer à Osada qui dans [OFCD01] teste plusieurs mesures choisies pour leur simplicité de calcul :

- la distance entre un point fixe et un point de la surface aléatoirement choisis ;
- la distance entre deux points de la surface aléatoirement choisis ;
- l'angle entre trois points pris aléatoirement sur la surface de l'objet ;
- l'aire du triangle formé par trois points pris aléatoirement sur la surface de l'objet ;
- le volume du tétraèdre formé par quatre points pris aléatoirement sur la surface de l'objet ;

Bien entendu, certains travaux ne se cantonnent pas à une seule approche et vont combiner plusieurs descripteurs. C'est par exemple le cas de Paquet [EM97] au sein du projet Nefertiti, qui va utiliser à la fois des descripteurs de formes mais aussi de couleur, de texture et de matériau, ou bien Vandeborre [VCD02] qui va tirer profit de la combinaison d'une distribution des index de courbure de Koenderink, d'une distribution de distances inspirée d'Osada et d'une distribution de volumes élémentaires (des tétraèdres reliant le centre de gravité de l'objet à une facette triangulaire).

## 5.2 Approches multi-vues

L'approche dite "multi-vues" est généralement qualifiée d'indexation 2D/3D. En effet, l'objectif ici est de rechercher les modèles tridimensionnels à partir d'une vue 2D d'un objet. L'indexation des modèles 3D se passe en deux étapes :

1. extraction d'un certain nombre de vues (i.e. projections) caractéristiques du modèle à indexer ;
2. description de chaque vue grâce à un descripteur de forme 2D.

Notons que, pour cette approche, la requête étant en deux dimensions, cette dernière pourra très bien être une photographie ou issue d'une vidéo.

À cet effet, la norme MPEG-7 intègre le schéma de description Multiview DS, ainsi que deux descripteurs de forme 2D :

- le descripteur ART (Angular Radial Transform)
- le descripteur CSS (Curvature Scale Space) caractérise les formes 2D grâce à une analyse multi-échelle des courbures de son contour.

Outre les descripteurs de forme utilisés pour la description, le choix du nombre de vues est essentiel.

Par exemple, dans [CS98], Chen et Stockman utilisent huit vues résultantes de la projection suivant les normales aux faces de l'octoèdre inscrit dans la sphère englobante de l'objet. Dans [SM02], Mahmoudi et Daoudi considèrent un nombre de vues fixé à sept. Trois de ces vues sont le résultat de projections suivant les trois axes de l'ellipsoïde d'inertie de l'objet (calculée en considérant l'objet comme homogène) et les quatre autres vues sont prises à des positions intermédiaires aux trois premières. Les descripteurs de forme utilisés pour ces projections sont de type CSS qu'ils organisent en M-Tree [CPZ97] afin d'accélérer la recherche et optimiser la gestion de la mémoire.

Dans [MA00], Mokhtarian et Abbasi utilisent eux aussi le CSS pour l'indexation des vues de leurs objets tridimensionnels, mais considèrent un nombre de vues par objet variant entre cinq et vingt-cinq. Les vues sont sélectionnées en fonction de la pertinence de leur description CSS : les vues trop similaires au sens du CSS sont éliminées.

Notons que dans [TZ04], Zaharia propose une comparaison des approches 3D et 2D/3D pour l'indexation de maillages dans le cadre d'applications de requêtes de similarité.

## 5.3 Approches structurales

Les approches structurales s'appuient sur une description guidée par l'organisation spatiale de primitives ou de caractéristiques de l'objet tridimensionnel. Cette approche

peut paraître plus naturelle et intuitive que les méthodes statistiques. Comme nous allons le voir, et compte tenu du format de notre descripteur, c’est cette approche qui va nous permettre d’utiliser celui-ci dans une application de type indexation d’objets 3D. De plus, cette méthode n’est pas limitée à la recherche de modèles 3D entre-eux mais peut être utilisée dans le cadre de la recherche d’un objet 3D dans une image 2D.

La sous-famille de l’approche structurale la plus connue et qui nous intéresse plus particulièrement, préconise une segmentation de l’objet en sous-parties volumiques élémentaires. Chaque sous-partie est alors représentée par une primitive.

Une première classe de travaux utilise des “primitives qualitatives” appelés geons. Les geons (géométric ions) ont été proposés par Biederman [Bie85] lors de son étude de la vision humaine et de la méthode de celle-ci pour identifier des objets dans des images. Cette théorie est connue sous le nom de *recognition-by-component (RBC)*. Ces géons sont pour lui au nombre de 36 et sont, avec un agencement correct, à même de décrire n’importe quel objet. Ils forment une sorte d’alphabet de description. Chacun d’entre-eux est une combinaison unique de quatre caractéristiques qualitatives de forme :

- la forme de la section est droite ou courbe.
- la section possède une symétrie axiale, centrale ou est asymétrique
- la taille de la section le long de l’axe est constante, croissante ou croissante puis décroissante.
- l’axe est courbe ou droit.

L’identification des parties par les geons se fait après l’étude de propriétés telles que la colinéarité, le parallélisme ou la symétrie.

D’autres auteurs vont créer leur propres geons. C’est par exemple le cas de Wu et Levine [WL94] qui utilisent seulement sept geons qu’ils qualifient de paramétriques.

La seconde classe utilise pour primitives volumiques, des modèles à paramètres quantitatifs comme par exemples les cylindres généralisés dans les travaux de Zerroug et Nevatia [ZN94], ou bien encore les superquadriques [PD90, GB93]. Pour faire le lien entre ces deux écoles, on peut noter que dans [RJ92], Raja montre comment obtenir un geon à partir d’une superquadrique.

Pour refléter la structure des modèles décrits, ces volumes élémentaires seront généralement liés par un graphe d’adjacence. Le problème de la comparaison de modèles devient alors un problème de mise en correspondance de graphes. De nombreux algorithmes, utilisant des méthodes très variées, le permettent. Néanmoins, les notions très strictes de la théorie des graphes ne sont pas toujours très adaptées à ce contexte. Nous aborderons ces questions dans le chapitre suivant.

Malheureusement, la plupart des travaux cités plus haut proposent des moyens pour

obtenir le descripteur d'objets 3D mais ne font qu'évoquer des approches pour la comparaison.

Ces approches peuvent sembler intuitives et attirantes, il faut tout de même noter que les méthodes inhérentes de segmentation de données 3D et d'appariements de graphes sont souvent très lourdes.

La décomposition par volumes élémentaires, même s'il elle est la plus présente, n'est pas la seule tendance de l'approche structurale.

La description des objets par leur squelette est une autre possibilité qui entre dans cette catégorie. Si les travaux sur l'indexation d'objets 3D utilisant le squelette sont rares, cette approche est beaucoup plus commune pour les formes 2D. On peut par exemple, citer les travaux de Klein et Sebastian [KTSK00, SKK01]. Ils considèrent le squelette comme un graphe et plus précisément un arbre. Ils mesurent la distance entre les objets 2D en calculant la distance d'édition entre les arbres correspondants. Ces articles ont pour intérêt de décrire particulièrement cette méthode de mise en correspondance. Nous reviendrons en détail sur la distance d'édition dans le chapitre suivant.

## 5.4 Conclusion

Ce tour d'horizon nous a montré que l'indexation d'objets tridimensionnels pouvait être abordée par des approches vraiment très différentes. Celle que nous adopterons, est tout naturellement l'approche structurale, liée directement à la méthode de description par un ensemble de superquadriques présentée dans la première partie. Celle-ci, bien que souvent plus lourde que les méthodes statistiques, a l'avantage d'être beaucoup plus intuitive et d'être proche du schéma de perception humain. Ainsi, il est facile pour un utilisateur de construire manuellement un descripteur comme le nôtre pour un objet réel ou imaginaire afin de formuler une requête.

Dans le cas d'une approche structurale, la comparaison des descripteurs s'effectue en général grâce à une comparaison des graphes sous-jacents. Nous avons appris dans la première partie de ce manuscrit à construire notre descripteur, la prochaine étape est donc la mise en correspondance de graphes. Nous aborderons ce problème dès le prochain chapitre.

# Chapitre 6

## Comparaisons de graphes

Nous avons vu que la comparaison d'objets décrits suivant une approche structurale était souvent synonyme de mise en correspondance des graphes liés aux descripteurs de ces d'objets. Ce qui correspond tout à fait exactement à notre problématique. Ce chapitre est l'occasion de déterminer la méthode qui nous permettra cette mise en correspondance et donc comparaison de graphes.

Nous commençons par rappeler quelques définitions sur la théorie des graphes, puis discutons sur différents outils permettant de comparer deux graphes afin de déterminer leur similitude, soit qualitativement, soit quantitativement. Nous aborderons les notions classiques que sont celles de la famille des isomorphismes. Ensuite, nous présenterons l'approche de comparaison par la distance d'édition qu nous avons déjà rapidement évoquée dans l'état de l'art.

### 6.1 Généralités

**Définition 6.1.1** *Un graphe  $G$  est un 4-uplet  $(V, E, \alpha, \beta)$  tel que :*

- $V$  est un ensemble de sommets ;
- $E$  est un ensemble de couples de  $V * V$  appelés arcs ;
- $\alpha : V \rightarrow L_V$  fait correspondre à chaque sommet de  $V$  un label  $L_V$  ;
- $\beta : E \rightarrow L_E$  fait correspondre à chaque arc de  $E$  un label de  $L_E$ .

Un graphe est dit *complet* si pour tout  $v, v' \in V$ , il existe un arc  $(v, v') \in E$ . Dans un tel graphe, chaque sommet est donc relié à tous les autres, et ainsi, si le nombre de sommet de  $G$  est égal à  $n$  alors  $|E| = \sum_{i=0}^{n-1} i = \frac{n(n-1)}{2}$ .

Si les arcs  $(v, v')$  et  $(v', v)$  doivent être distingués, le graphe sera qualifié d'*orienté*, et de *non-orienté* sinon (figure 6.1 ).

La labelisation par  $\alpha$  et  $\beta$  n'est pas nécessaire à la définition d'un graphe. Si ces fonctions ne sont pas définies, le graphe sera dit *non-attribué*. Néanmoins, les graphes

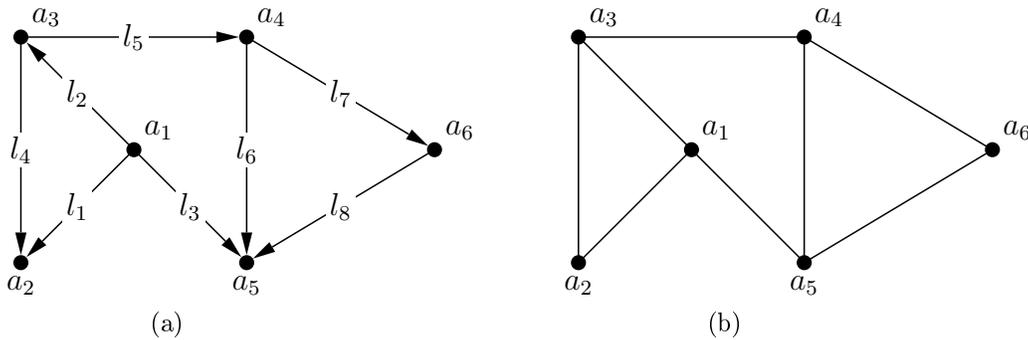


FIG. 6.1 – (a) Un graphe orienté dont les arcs sont attribués. (b) Un graphe non-orienté dont les arcs ne sont pas attribués

non-attribués sont rares dans le domaine qui nous intéresse, car plus qu’une simple structure, les graphes représentent le plus souvent une scène (resp. un objet), les sommets correspondant à des objets (resp. des parties d’objet) et les arcs des relations entre ceux-ci comme par exemple un voisinage ou une relation temporelle.

Dans le cas de notre descripteur, le graphe définit un objet tridimensionnel. Chaque sommet correspond à une partie de cet objet modélisé par une superellipsoïde. Ainsi,  $L_V$  est l’ensemble des vecteurs de onze paramètres définissant une superellipsoïde. Les arcs du graphe du descripteur indiquent une relation d’adjacence entre les deux sommets liés mais ne sont pas attribués. Comme la relation indiquée par les arcs est commutative (si  $s_1$  est adjacent à  $s_2$ , alors  $s_2$  est adjacent à  $s_1$ ), le graphe de notre descripteur est non-orienté.

Dans la suite de ce rapport, nous ne distinguerons plus le descripteur et le graphe du descripteur. Nous nous permettrons donc d’utiliser le vocabulaire des graphes pour qualifier directement le descripteur (nous pourrons parler, par exemple, des arcs du descripteur).

**Définition 6.1.2** On appelle isomorphisme de graphe de  $G = (V, E, \alpha, \beta)$  vers  $G' = (V', E', \alpha', \beta')$  la fonction bijective  $f$  qui à chaque sommet de  $G$  fait correspondre un sommet de  $G'$  tout en préservant la structure de  $G$  (i.e. les arcs) et tous les attributs. C’est-à-dire  $f : V \rightarrow V'$  telle que :

- $\forall v \in V$ ,  $\alpha(v) = \alpha'(f(v))$ ;
- $\forall e = (v_1, v_2) \in E$  il existe  $e' = (f(v_1), f(v_2)) \in E'$  et  $\beta(e) = \beta'(e')$
- $\forall e' = (v'_1, v'_2) \in E'$  il existe  $e = (f^{-1}(v'_1), f^{-1}(v'_2)) \in E$  et  $\beta'(e') = \beta(e)$

S’il existe un isomorphisme de graphe  $f$  de  $G$  vers  $G'$  alors on dira que  $G$  est isomorphe à  $G'$ , et comme  $f$  est bijective, si  $G$  est isomorphe à  $G'$  alors  $G'$  est isomorphe à  $G$  (Figure 6.2)

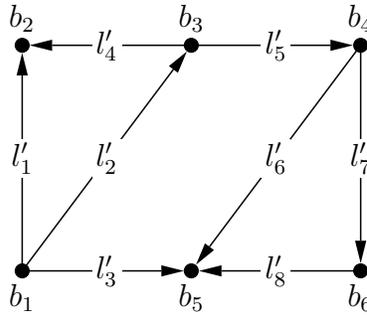


FIG. 6.2 – Ce graphe est isomorphe au graphe de la figure 6.1a s’il existe une bijection  $f$  telle que  $\forall i \in [1, 6] f(a_i) = b_i$  et  $\forall j \in [1, 8] f(l_j) = l'_j$

Essayer de comparer deux de nos descripteurs d’objets en déterminant les isomorphismes possibles entre ceux-ci serait très peu efficace. En effet, même si le nombre de sommets et la structure des deux descripteurs étaient identiques, les superellipsoïdes représentant les parties des objets décrits auraient vraiment peu de chance d’être exactement semblables, et cela même si ces objets sont très proches. Cette égalité est normalement nécessaire si on s’en tient à la définition d’un isomorphisme. De plus, le résultat d’une comparaison par isomorphisme est en l’état uniquement qualitatif, ce qui limite beaucoup les applications d’indexation et notamment rend impossible tout classement par degré de similitude (puisque cette notion de degré n’existe pas).

Une première idée à envisager pourrait être de considérer une interprétation tolérante de l’égalité portant sur les sommets :

$$\forall v \in V, \alpha(v) = \alpha'(f(v))$$

en la remplaçant par :

$$\forall v \in V, d(\alpha(v), \alpha'(f(v))) < \epsilon$$

avec  $d(s_1, s_2)$  une mesure de similarité entre les superellipsoïdes  $s_1$  et  $s_2$ .

Cela aurait pour avantage d’une part d’avoir au moins quelques comparaisons positives, mais aussi de pouvoir obtenir une quantification de la similarité entre les deux descripteurs combinant les différentes mesures de similarité de chaque superellipsoïde correspondante dans l’un et l’autre descripteur.

Mais même ainsi, l’isomorphisme est une notion trop restrictive et stricte pour nous être vraiment utile. Pour que deux objets soient évalués comme semblables ou même proches, il faudrait que les descripteurs aient le même nombre de sommets et exactement la même structure (i.e. configuration d’arcs).

**Définition 6.1.3** *Un graphe  $S = (V', E', \alpha', \beta')$  est un sous-graphe d’un graphe  $G = (V, E, \alpha, \beta)$  si et seulement si :*

- $E' \subseteq E$  ;
- $V' \subseteq V$  ;
- Pour tout  $v \in V'$  ,  $\alpha'(v) = \alpha(v)$  ;
- Pour tout  $e \in E'$  ,  $\beta'(e) = \beta(e)$ .

Dans notre cas, un sous-graphe du graphe d'un descripteur représente une partie de l'objet décrit. Mais il faut tout de même faire attention au fait que la totalité des relations (i.e. les arcs) entre les segments d'objets de ce sous-graphe n'est pas forcément préservée. Cela n'a pas vraiment de sens puisqu'une partie d'objet garde la même structure interne séparée ou non de l'objet auquel elle appartient. Pour être exact, si on veut vraiment définir une partie d'objet à partir de notre descripteur, il faut se référer à un sous-graphe maximal en terme de nombre d'arcs pour un ensemble de sommets donnés. C'est-à-dire un sous-graphe de  $G$  tel qu'il n'existe pas d'autre sous-graphe à  $G$  avec les mêmes sommets et contenant plus d'arcs (Figure 6.3).

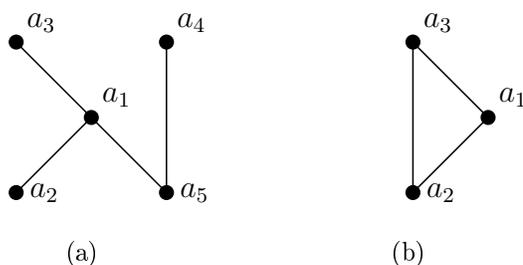


FIG. 6.3 – Sous-graphes du graphe de la figure 6.1b. (a) Non maximal (b) Maximal en terme de nombre d'arcs.

**Définition 6.1.4** *Un isomorphisme de sous-graphe  $f$  de  $G = (V, E, \alpha, \beta)$  vers  $G' = (V', E', \alpha', \beta')$  est une fonction telle qu'elle définit un isomorphisme de graphe de  $G$  vers  $S$  avec  $S$  un sous-graphe de  $G'$ .*

La notion d'isomorphisme de sous-graphe est déjà plus intéressante pour notre application. En utilisant la version tolérante de l'égalité des labels des sommets et des arcs, elle pourrait par exemple permettre de comparer un modèle  $S$  très simple en terme de nombre de sommets et d'arcs avec le graphe d'un objet réel  $G$ . S'il existe un isomorphisme de sous-graphe de  $S$  vers  $G$ , on pourra conclure que  $G$  appartient à la classe de  $S$ .

Certes, ce n'est pas vraiment ce que nous recherchons. Nous n'avons pas de modèle et voulons comparer directement des descripteurs d'objets 3D réels. En revanche, on peut s'intéresser au *sous-graphe commun*  $S$  de  $G$  et  $G'$ , c'est-à-dire au graphe  $S$  tel qu'il existe

un isomorphisme de sous-graphe de  $S$  vers  $G$  et de  $S$  vers  $G'$ . Ainsi, le *sous-graphe commun maximal*  $S$  de  $G$  et  $G'$  est un graphe tel que  $G$  et  $G'$  ne possèdent pas de sous-graphe commun plus grand ni en terme de nombre de sommets, ni en terme de nombre d'arcs. Cette dernière notion est un cas particulier d'isomorphisme et d'isomorphisme de sous-graphe. En effet, supposons que  $S$  soit le *sous-graphe commun maximal* à la fois de  $G$  et de  $G'$  alors si  $G = S$  et  $G' = S$  alors  $G$  et  $G'$  sont isomorphes et d'autre part, si  $G = S$  et  $\forall G'$  alors il existe un isomorphisme de sous-graphe entre  $G$  et  $G'$ .

Le sous-graphe commun maximal est pour l'instant la notion qui semble la plus appropriée à notre problème. Toujours en utilisant une version tolérante de l'isomorphisme, déterminer le sous-graphe commun maximal de deux descripteurs permettrait de mettre en évidence des structures sous-jacentes plus ou moins grandes, que ce soit en terme de nombre de noeuds ou bien même de volume occupé par les superellipsoïdes correspondantes (on choisira la plus adaptée suivant les applications), et ainsi classer les descripteurs et donc les objets 3D.

Dans ce cas de figure, le modèle de classe, que l'on évoquait dans un paragraphe précédent, serait en fait un sous-graphe de chacun des descripteurs. Le fait de pouvoir déterminer si deux descripteurs possèdent une importante structure commune, et donc un modèle commun, indiquerait que les objets décrits sont assez proches et qu'ils appartiennent à la même classe (figure 6.4) et ce sans avoir à prédéfinir de classes.

H. Blunke a défini une mesure de similarité entre les graphes basée sur le sous-graphe commun maximal [MB98]. La définition en est très simple :

$$d(G, G') = 1 - \frac{|scm(G, G')|}{\max(|G|, |G'|)} \quad (6.1)$$

avec  $G$  et  $G'$  deux graphes non vides,  $scm(G, G')$  le sous-graphe commun maximal à  $G$  et  $G'$  et  $|G|$  le nombre de sommets de  $G$ .

Il démontre que cette mesure définit une distance. Cette méthode propose de comparer deux graphes de manière très générale, il utilise donc logiquement le rapport entre le nombre de sommets du sous-graphe commun maximal et des graphes à comparer. Bien entendu rien n'empêche de l'adapter pour l'utiliser dans des applications particulières. Dans notre cas par exemple, où les sommets représentent des superellipsoïdes, il serait plus adapté, comme évoqué plus haut, de prendre en compte les volumes de ces superellipsoïdes et ainsi utiliser le rapport des volumes du sous-graphe commun maximal et des graphes à comparer. Ainsi, nous proposons une adaptation de la distance de Bunke (eq. 6.1) à notre problème :

$$d(G, G') = 1 - \frac{Volume(scm(G, G'))}{\max(Volume(G), Volume(G'))} \quad (6.2)$$

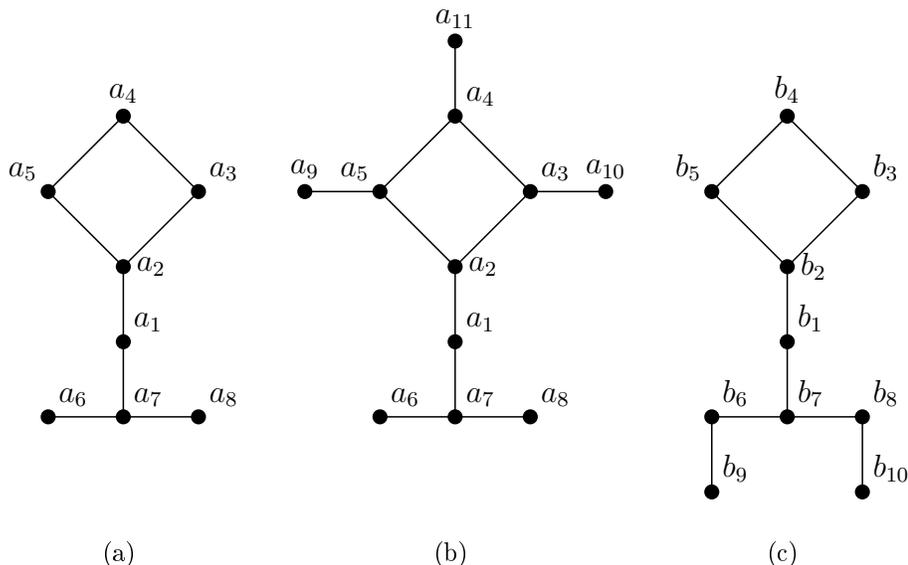


FIG. 6.4 – Si on admet que les superellipsoïdes  $a_i$  et  $b_i$  sont relativement similaires, on peut dire que (b) et (c) sont des objets de classe (a)

Si cette approche peut paraître séduisante au premier abord, elle possède au moins une limitation.

En effet, nous voulons comparer des descripteurs d'objets qui ont été construits par notre méthode de segmentation par modélisation (entre autres, car les descripteurs peuvent avoir été obtenus avec n'importe quelle technique). Il faut donc que la méthode de comparaison soit au moins robuste aux imperfections les plus couramment produites par incertitudes liées à notre méthode. Or, quelle peut être l'interprétation pratique d'une approche par isomorphisme de sous-graphe ? C'est ce que nous allons voir dans la suite.

Dans un tel algorithme, nous allons déterminer le plus grand sous-graphe commun aux deux graphes que nous allons essayer de comparer. Ainsi, certains sommets (i.e. superellipsoïdes représentant une partie de l'objet) vont être supprimés, ou plus exactement, mis de côté avant la comparaison deux à deux des sommets restants. Dans le cas de comparaisons d'objets similaires, ces sommets évincés modéliseraient les détails qui diffèrent sur ces objets ou peuvent être dus à des problèmes apparus lors de la segmentation. En effet, il peut arriver que de petits groupes de points le plus souvent à la charnière entre deux parties plus importantes, ne puissent être intégrés (i.e fusionnés) à aucune des régions voisines, à cause d'une tolérance aux erreurs trop stricte par exemple. Ces petites régions charnières restent donc dans le descripteur final sans pour autant représenter une partie bien définie de l'objet. Pour cela, on peut dire que l'approche du plus grand sous-graphe commun est adaptée car tout à fait à même de passer outre ces imperfections (Figure

6.5).

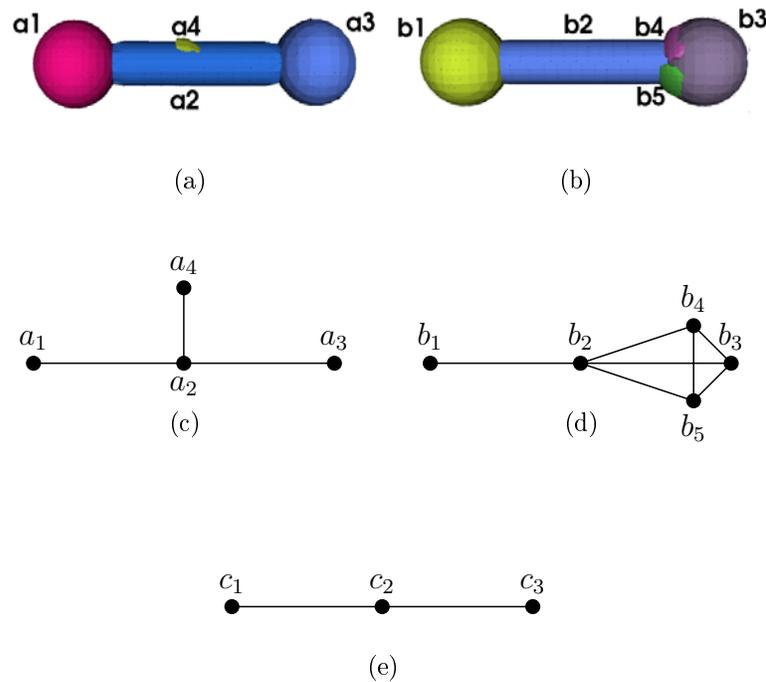


FIG. 6.5 – (a) et (b) sont deux descripteurs obtenus pour deux objets très proches. De petites superquadriques sont présentes dans ces descripteurs. Elles peuvent être le résultat d'erreurs de modélisation ou bien de réelles aspérités sur la surface de l'objet. (c) et (d) sont les graphes associés respectivement au descripteur (a) et (b). Supposons que notre définition de l'égalité entre superquadriques soit assez large pour que l'on puisse écrire que :  $a_1 = c_1 = b_1$ ,  $a_2 = c_2 = b_2$  et  $a_3 = c_3 = b_3$ , alors le graphe (e) est un sous-graphe commun maximal pour (c) et (d). Dans cet exemple, la notion de sous-graphe commun maximal remplit très bien son rôle d'éliminateur de bruit. Le volume occupé par l'objet associé au descripteur d'un graphe comme (e), étant très proche du volume des objets (a) et (b), on pourrait conclure que ces objets sont similaires (le volume de leurs différences, les bruits, étant faibles en comparaison).

En revanche, la méthode sera beaucoup moins robuste aux différences de descriptions provoquées par la "non fusion" de deux régions lors de la segmentation. Il peut arriver, parce que les objets sont légèrement différents ou que certains résultats d'approximation diffèrent à cause d'échantillonnages différents d'un même objet, qu'une même partie d'objet soit représentée dans un cas par une superellipsoïde et dans un autre par deux (i.e. dans un cas la fusion entre deux sous-parties a pu avoir lieu et dans un autre cas, pour la même partie de l'objet, elle ne l'a pas été). Si on essaie de comparer un objet où la fusion a pu avoir lieu avec un objet où la fusion n'a pas eu lieu, la méthode du plus grand sous-graphe commun va devoir éliminer l'une des deux sous-parties non fusionnées afin de

faire correspondre les structures des deux descripteurs (Figure 6.6). Ainsi, puisque la comparaison des superellipsoïdes (i.e des sommets) se fait deux à deux, nous comparerons une partie de l'objet avec une fraction de cette même partie dans l'autre objet. L'algorithme considérera donc comme différentes les parties concernées de ces objets.

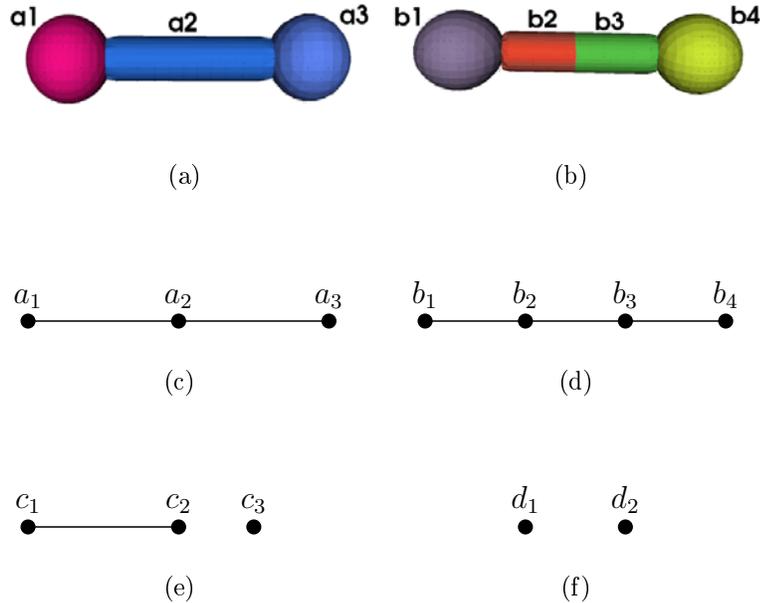


FIG. 6.6 – (a) et (b) sont des descripteurs différents, bien que tout deux valides, pour un même objet. Dans le descripteur (b), les superquadriques de la barre centrale n'ont pas fusionné. (c) et (d) sont les graphes associés respectivement aux descripteurs (a) et (b). Si un sous-graphe commun maximal de trois sommets existe pour les graphes (c) et (d), il est du type de (e), mais cela est impossible car on ne peut mettre en correspondance  $c_2$  avec deux sommets similaires dans (c) et (d). Le sommet  $a_2$  et le couple  $b_2, b_3$ , bien que très proches sont considérés comme différents et c'est (f) le sous-graphe commun maximal pour (c) et (d) avec  $a_1 = d_1 = b_1$  et  $a_3 = d_2 = b_4$ . Le volume commun (i.e. le volume de l'objet décrit par le graphe (f)) et par là, la quantité de données sur laquelle repose la comparaison est relativement faible par rapport aux objets originaux.

La théorie des graphes dispose donc de nombreuses notions permettant la comparaison. Le plus grand des sous-graphes communs en est une particulièrement intéressante pour notre application, mais comme toute la famille des isomorphismes, celle-ci est trop rigide pour être robuste aux petites différences de modélisation qui sont difficilement évitables dans le domaine de la description d'objets tridimensionnels. D'autres approches ont justement été développées pour tenir compte de ces erreurs et différences de modélisation. La distance d'édition en est une, et comme nous allons le voir, elle reste très proche et

peut même, dans certains cas, devenir avantageusement équivalente aux notions que nous venons d'aborder.

## 6.2 La distance d'édition

La distance d'édition est, comme son nom le laisse présager, un outil de mesure de similarité entre deux entités se basant sur le coût des modifications à leur appliquer pour pouvoir passer de l'une à l'autre. Il s'agit donc de définir des opérations d'éditions (i.e. de modifications) sur ces entités et le coût de ces modifications. Ensuite, il faut déterminer la séquence d'opérations ayant le coût le plus faible permettant de transformer la première entité en la seconde.

### 6.2.1 Comparaison de chaînes de caractères

La notion de distance d'édition a été développée en premier lieu pour la comparaison de chaînes de caractères. Sa mesure entre deux chaînes peut, par exemple, permettre de proposer les mots les plus proches au sens de la distance d'édition, et donc les plus probables, dans le cas de la correction orthographique d'un mot absent d'un dictionnaire par exemple.

Comme nous l'avons déjà évoqué, il s'agit de définir, dans un premier temps, les opérations d'éditions. Dans le cas de comparaisons de chaînes, les opérations les plus communes sont :

- le remplacement d'un caractère par un autre (la relabélisation) ;
- l'insertion d'un caractère ;
- la suppression d'un caractère.

En ce qui concerne les coûts de ces modifications, elles dépendront des applications. En aucun cas, ce coût n'a besoin d'être constant pour une opération. Par exemple, si l'application visée est la correction des fautes de frappe, le coût de la relabélisation pourra être fonction de la distance euclidienne entre les deux caractères sur le clavier. Ainsi, *zon* sera considéré comme plus proche de *son* que de *mon*, la touche *z* étant plus proche du *s* que du *m*.

Bien sûr, il existe une infinité de séquences d'opérations permettant de passer d'une chaîne à l'autre. Dans le dernier exemple, une opération de relabélisation suffisait mais on aurait pu imaginer une séquence insertion puis suppression pour passer d'un mot à l'autre (*zon* → *szon* → *son*). Si les coûts des opérations d'insertion et de suppression avaient été définis en toute logique, cette séquence aurait eu sans doute un coût supérieur à l'unique relabélisation.

Afin de calculer la distance d'édition, il nous faut un algorithme permettant de déterminer la séquence de coût minimum parmi toutes les séquences possibles permettant de passer d'une chaîne à l'autre.

La méthode de calcul de la distance d'édition rentre dans le cadre de la programmation dynamique. Ce type de procédé permet la résolution de problème nécessitant une séquence de décisions dépendantes les unes des autres (ici la décision se fait à chaque étape sur le choix entre les différentes opérations de base : insertion, suppression, relabélisation). Les problèmes qui vont être résolus par la programmation dynamique satisfont le principe d'optimalité de Bellman (1955), à savoir que la solution d'un problème de taille  $n$  s'exprime en fonction de la meilleure solution de taille  $n - 1$ . Il faut donc résoudre le problème de taille  $n - 1$  pour passer à la résolution du problème de taille  $n$ .

Ainsi l'algorithme peut se définir tout naturellement de manière récursive :

$$\begin{aligned}
 \text{Distance}(S_1 + c_1, S_2 + c_2) &= \min \{ \text{Distance}(S_1, S_2) + \text{Relabélisation}(c_1, c_2), \\
 &\quad \text{Distance}(S_1, S_2 + c_2) + \text{Suppression}(c_1), \\
 &\quad \text{Distance}(S_1 + c_1, S_2) + \text{Suppression}(c_2) \} \\
 \text{Distance}(S_1 + c_1, \epsilon) &= \infty \\
 \text{Distance}(\epsilon, S_2 + c_2) &= \infty
 \end{aligned}$$

avec  $S_i$  une chaîne de caractères quelconque (qui peut être vide),  $c_i$  un caractère quelconque, et  $\epsilon$  la chaîne vide et  $+$  la concaténation.

On peut tout de suite remarquer que l'insertion n'est pas présente dans l'algorithme. En effet, dans cette définition l'insertion n'est que la réciproque de la suppression quand la modification se fait "dans l'autre sens". C'est-à-dire que l'insertion d'un caractère dans la première chaîne est équivalent à la suppression de ce même caractère (i.e. le caractère à la même place) dans la seconde chaîne. Néanmoins, certaines applications peuvent nécessiter la définition d'une fonction d'insertion distincte de la suppression, et rien ne nous empêche de l'implémenter.

## 6.2.2 Adaptations aux graphes

Si cette notion de distance d'édition a, dans un premier temps, été définie pour la comparaison de chaînes de caractères, elle a été adaptée au domaine des graphes où elle se révèle être un outil de mesure de similarité très puissant pour résoudre le problème des graphes bruités.

Dans le cas des graphes, les opérations de modifications vont donc porter sur sa structure et les attributs des sommets et des arcs. En gardant le même type d'opérations que pour les chaînes, ce qui est généralement possible, on va s'autoriser à :

- changer l'attribut d'un sommet en un autre (relabélisation);
- changer l'attribut d'un arc en un autre (relabélisation);
- insérer un sommet (cela suppose insérer aussi des arcs);
- supprimer un sommet (cela suppose distribuer ses arcs à ses voisins afin de préserver l'intégrité du graphe).

Nous pourrions alors définir un algorithme de distance d'édition s'appliquant aux graphes. Une autre méthode pour calculer cette distance entre deux graphes est de les transformer en chaînes de caractères puis de traiter le problème comme un problème de distance d'édition entre chaînes.

Le passage d'un graphe en chaîne se fait suivant un parcours de ce graphe. Un graphe quelconque ne peut être équivalent à une seule chaîne de caractères, mais chaque parcours possible de ce graphe correspond à une chaîne unique et si une chaîne correspond au parcours d'un graphe, le graphe défini est unique.

Pour générer cette chaîne correspondante, il faut imaginer chaque sommet comme une paire de parenthèses ayant pour indice le numéro de ce sommet. Nous allons alors parcourir tous les sommets du graphe. À chaque fois que nous arriverons sur un nouveau sommet, nous ouvrirons sa parenthèse, puis visiterons tous ses voisins non encore visités selon le même principe (i.e de manière récursive). Quand tous ses voisins auront été visités, nous fermerons sa parenthèse.

Comme on aura pu le remarquer, cette stratégie pour calculer la distance d'édition entre deux graphes quelconques n'est pas complète. En effet, le parcours utilisé pour générer la chaîne est en fait celui d'un arbre couvrant du graphe. Bien sûr, plusieurs arbres couvrants existent pour un même graphe. Or, si on souhaite déterminer la distance entre un graphe  $A$  et un graphe  $B$  que nous transformons respectivement en une chaîne  $S_A$  et  $S_B$ , les objets  $A$  et  $B$  auront beau être identiques, nous n'arriverons pas à affirmer la similarité si ce ne sont pas les "mêmes" arbres couvrants qui ont été choisis pour produire  $S_A$  et  $S_B$ . Ainsi, d'une part si nous ne pouvons être certains que les sommets de départ des parcours (i.e. les racines des arbres couvrants) sont ceux qui correspondent l'un avec l'autre dans  $A$  et  $B$ , nous devons comparer un parcours de  $A$  avec autant de parcours de  $B$  partant d'une origine différente qu'il y a de sommets dans  $B$ . Et d'autre part, plusieurs chaînes peuvent résulter d'un même arbre si les fils de chaque noeud n'ont pas un ordre de visite prédéterminé. Et alors, si ces voisins ne sont pas ordonnés, nous devons faire autant de parcours de  $B$  qu'il y en a de différents (c'est-à-dire en prenant en compte tous les ordonnancements possibles, les permutations, des voisins).

Si on considère, par exemple, la figure 6.7 qui comme on peut le remarquer ne présente pas l'ensemble exhaustif des parcours de 6.7(a) possibles mais seulement trois exemples (b), (c) et (d), les chaînes correspondantes sont :

- Pour 6.7b :  $(_1(7(8)8(6)6)7(2(3(4(5)5)4)3)2)_1$

- Pour 6.7c :  $(_1(2(5(4(3)3)4)5)_2(7(6)6(8)8)7)_1$
- Pour 6.7d :  $(_3(4(5)5)4(2(1(7(6)6(8)8)7)1)2)_3$

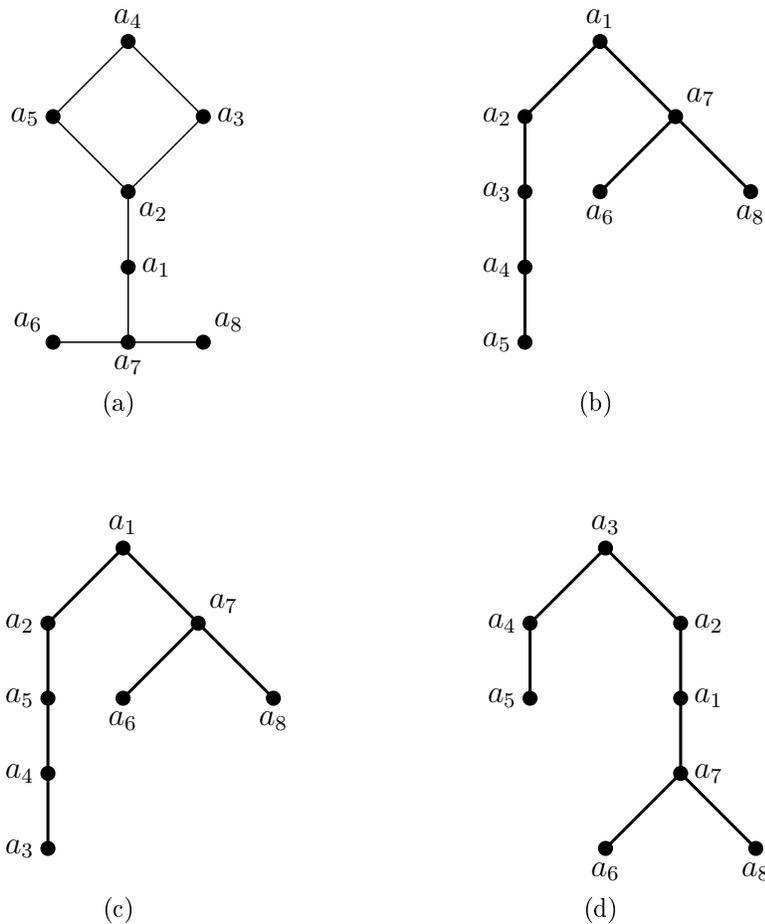


FIG. 6.7 – (b), (c) et (d) sont trois arbres couvrants de (a). Ils correspondent tous les trois à un parcours différent. (b) et (c) ont  $a_1$  pour sommet de départ mais (b) donne la priorité à son fils gauche tandis que (c) la donne à son fils droit. Le parcours (d) a pour origine  $a_3$  et donne la priorité au fils gauche.

À chaque sommet correspondent deux caractères intimement liés : une paire de parenthèses. Le problème de distance d'édition que nous aurons à résoudre sera donc particulier puisque les opérations ne seront pas appliquées sur un seul caractère comme habituellement mais sur deux (la paire de parenthèses). Quand nous relabéliserons, nous modifierons les deux parenthèses de la même manière. Quand nous supprimerons, nous supprimerons les deux parenthèses en une seule opération, etc. Cela est très important pour préserver l'intégrité du graphe (i.e. pour que la chaîne continue de correspondre à un graphe, ou plus exactement à un arbre).

Après adaptation, l'algorithme devient donc (nous noterons '(' les parenthèses de la première chaîne et '[' celles de la seconde) :

$$\begin{aligned}
& \text{Distance}(A_1({}_i A_2)_i, B_1[{}_j B_2]_j) = \\
& \min \{ \text{Distance}(A_1, B_1) + \text{Distance}(A_2, B_2) + \text{coût Relabélisation}(({}_i)_i, [{}_j]_j), \\
& \quad \text{Distance}(A_1 A_2, B_1 [{}_j B_2]_j) + \text{coût Suppression}(({}_i)_i), \\
& \quad \text{Distance}(A_1({}_i A_2)_i, B_1 B_2) + \text{coût Suppression}([{}_j]_j) \} \\
& \text{Distance}(A, \epsilon) = \infty \\
& \text{Distance}(\epsilon, B) = \infty
\end{aligned}$$

Nous pouvons nous interroger sur le fonctionnement de la suppression et sa capacité à préserver une structure de graphe tout en la modifiant. D'abord remarquons que si une chaîne est de type  $A = ({}_i B)_i$  alors le graphe correspondant est connexe. C'est un arbre et le sommet  $i$  est la racine de cette arbre. En effet, la chaîne  $A$  signifie que l'on entre par le sommet  $i$  (parenthèse ouvrante d'indice  $i$ ), que l'on parcourt récursivement tous les voisins de  $i$  (ce parcours est représenté par la chaîne  $B$ ), puis que l'on sort par le sommet  $i$  (parenthèse fermante d'indice  $i$ ). La chaîne  $B$  définit l'ensemble des sous-arbres fils du sommet  $i$ . C'est-à-dire que  $B$  est de la forme  $({}_{j_1} B_1)_{j_1} ({}_{j_2} B_2)_{j_2} \dots ({}_{j_n} B_n)_{j_n}$  si le sommet  $i$  possède  $n$  fils. Si on observe les modifications appliquées aux chaînes par l'algorithme, on peut remarquer que toutes les chaînes traitées sont du même type que  $B$ . Ainsi, lors d'une suppression, on élimine la dernière parenthèse de la chaîne et sa correspondante. On aura par exemple :

$$({}_{j_1} B_1)_{j_1} ({}_{j_2} B_2)_{j_2} \dots ({}_{j_n} B_n)_{j_n} \longrightarrow ({}_{j_1} B_1)_{j_1} ({}_{j_2} B_2)_{j_2} \dots B_n$$

or,  $B_n$  est aussi de la forme de  $B$  :

$$B_n = ({}_{k_1} C_1)_{k_1} ({}_{k_2} C_2)_{k_2} \dots ({}_{k_m} C_m)_{k_m}$$

et donc, après une suppression,  $B$  devient :

$$({}_{j_1} B_1)_{j_1} ({}_{j_2} B_2)_{j_2} \dots ({}_{j_{n-1}} B_{n-1})_{j_{n-1}} ({}_{k_1} C_1)_{k_1} ({}_{k_2} C_2)_{k_2} \dots ({}_{k_m} C_m)_{k_m}$$

Une suppression élimine donc la racine d'un arbre ou d'un sous-arbre. Ses sous-arbres fils (i.e.  $({}_{k_1} C_1)_{k_1} ({}_{k_2} C_2)_{k_2} \dots ({}_{k_m} C_m)_{k_m}$ ), s'ils existent, deviennent les fils de son père (figure 6.8).

## 6.3 Conclusion

La distance d'édition est donc un outil intéressant pour résoudre le problème de comparaison d'objets décrit par des graphes. Elle a d'ailleurs déjà été utilisée dans ce but [SKK01]. De plus, la distance d'édition est loin d'être incompatible avec les notions

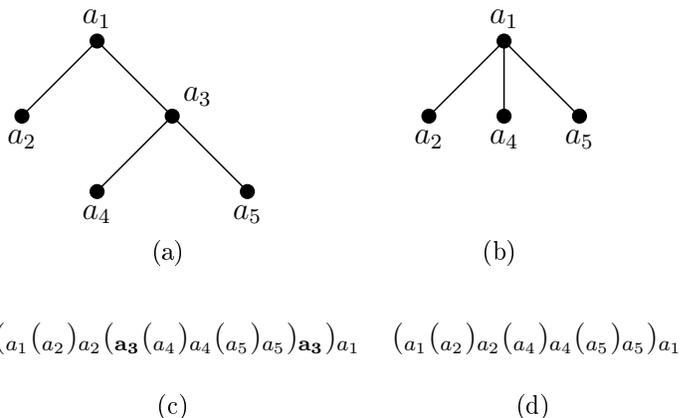


FIG. 6.8 – (a) Un arbre. (b) Le même arbre après une opération de suppression de  $a_3$ . (c) (d) sont les chaînes correspondantes respectivement à (a) et (b).

traditionnelles de comparaison de la théorie des graphes que nous avons abordées dans le paragraphe 6.1. En effet, H. Bunke a montré dans [MB98] que si les coûts de suppression et d'insertion étaient toujours inférieurs aux coûts de relabélisation alors déterminer la distance d'édition était équivalent à déterminer le sous-graphe commun maximal ! Les cas d'isomorphismes et d'isomorphismes de sous-graphes étant des cas particuliers de sous-graphe commun maximal, ils peuvent être déterminés grâce à l'algorithme de distance d'édition. Il faut donc voir la distance d'édition comme un moyen de mettre en évidence, qualifier et quantifier les similitudes entre des graphes. Elle permet de calculer simplement ce fameux sous-graphe commun maximal. C'est de cette manière que H. Bunke propose de le faire pour déterminer sa distance (cf équation 6.1), alors que cette distance avait justement été mise au point pour éviter d'avoir à passer par la distance d'édition et la mise en place des fonctions de coûts. Car en effet, la définition du coût des opérations d'édition est cruciale car elle détermine à elle seule le résultat de la méthode de distance d'édition. La définition de fonctions de coûts adaptées à notre problème sera l'objet de la première partie du chapitre suivant.

# Chapitre 7

## Comparaison d'objets 3D par la distance d'édition entre graphes de superellipsoïdes

Dans ce chapitre, nous allons appliquer cette notion de distance d'édition, que nous avons précédemment examinée afin de comparer nos descripteurs d'objets tridimensionnels. Dans un premier temps, il est donc nécessaire de définir précisément les opérations d'édition qui nous permettront de modifier les descripteurs et de passer de l'un à l'autre. Nous proposerons un coût pour chacune des opérations que nous aurons retenues. Ensuite, nous présenterons les résultats obtenus dans le cas de comparaisons d'objets deux à deux puis dans une application de type indexation.

### 7.1 Choix et coût des fonctions d'édition

Comme nous l'avons vu dans le chapitre précédent, les opérations d'édition les plus communément utilisées sont : la relabélisation, la suppression et l'insertion. Sont-elles pertinentes dans le cadre de notre application ? Nous pouvons distinguer deux différents groupes d'opérations :

1. les opérations de suppression et d'insertion modifient la structure même du graphe en ajoutant ou supprimant sommets et arêtes ;
2. l'opération de relabélisation s'occupe, à proprement parler, de la comparaison des graphes en comparant deux à deux chacune de leurs primitives.

Ces groupes peuvent aussi nous permettre de comprendre un peu mieux le fonctionnement de l'algorithme, qui ayant été défini de manière très synthétique, peut paraître un peu obscur. La relabélisation nécessite que l'on ait pu mettre en correspondance les

graphes ou les parties de graphes sur lesquelles l'algorithme travaille<sup>6</sup>, la raison des opérations de modification de structure est de transformer dans un premier temps les graphes afin de présenter à la relabélisation des structures superposables, isomorphes.

La relabélisation paraît donc indispensable. Rappelons qu'il s'agit de la substitution d'une primitive de la première structure par celle correspondante dans la seconde structure ; c'est-à-dire du remplacement d'un caractère par un autre dans le cas de distance d'édition appliquée aux chaînes de caractères et du remplacement d'une superellipsoïde par une autre dans le cas de nos descripteurs. C'est aussi, et surtout, le calcul du coût d'une telle modification. Cette relabélisation est la base même de la comparaison, elle serait définie aussi bien dans un algorithme de type isomorphisme, et sans doute présente sous une forme ou sous une autre dans la grande majorité des méthodes de comparaison de structures. C'est la partie de comparaison au niveau local (i.e des primitives) d'une méthode de mesure de similarités de structures composées. Nous définirons donc une opération de relabélisation.

Comme nous l'avons vu dans le paragraphe 6.2.1, les opérations d'insertion et suppression sont très liées et peuvent même être définies comme une seule et même opération qui change de nom suivant que l'on veuille passer d'une entité  $A$  à une entité  $B$  ou bien d'une entité  $B$  à une entité  $A$ . Dans notre cas, comme pour la plupart des applications, si nous optons pour le couple suppression-insertion comme opérations de modification de structures, celles-ci seront confondues puisque que nous considérons qu'il n'y a aucune différence de dégradations entre ajouter une primitive à un objet  $B$  pour le rapprocher de  $A$ , que de supprimer cette même primitive (car c'est bien de la même primitive qu'il s'agit) de  $B$  pour le rapprocher de  $A$ .

Mais est-ce que ce couple suppression-insertion est adaptée à notre application ? Puisque nous avons rejeté, au moment où nous l'examinions, la notion de sous-graphe commun maximal, nous devons en toute logique répondre négativement à cette question. En effet, nous avons vu que la distance d'édition pouvait être un moyen de calculer le sous-graphe commun maximal de deux graphes, et cela est en partie dû au fait que le couple suppression-insertion est tout à fait compatible avec cette notion. Pour trouver un sous-graphe commun à  $G$  et  $G'$ , il suffit de supprimer des arêtes et des sommets à  $G$  et  $G'$ . C'est justement cette méthode de modification structurelle du graphe que nous trouvons inadaptée pour la résolution de notre problème, ou du moins peu à même d'être robuste aux différences de description les plus courantes dans notre méthode de segmentation/modélisation (revoir les figures 6.5 et 6.6 du chapitre précédent). Ainsi, pour résoudre à la fois le problème des détails persistants et celui des "non-fusions", nous proposons d'utiliser une opération de fusion de superellipsoïdes (figure 7.1). Celle-ci aura bien entendu

---

<sup>6</sup>La méthode étant récursive, pour déterminer la distance entre deux graphes, nous allons être amenés à travailler sur des graphes de rang inférieur, c'est-à-dire ayant moins de sommets.

une opération complémentaire de scission. Cette fusion de superellipsoïdes, et donc de sommets de graphe, se traduit en terme de chaîne de caractères par la suppression de l'avant-dernière parenthèse et de sa correspondante, et par le remplacement de la dernière parenthèse et sa correspondante par une nouvelle paire représentant les deux primitives fusionnées. Ainsi, lors d'une fusion, les fils (i.e. les parties voisines de l'objet) des deux sommets fusionnés deviennent les fils du nouveau sommet.

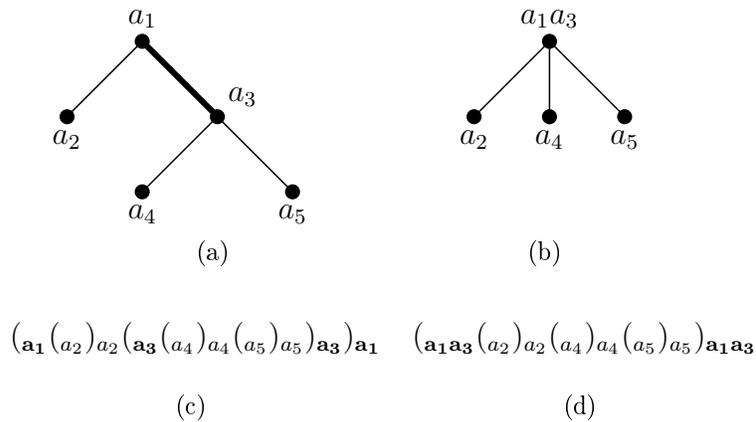


FIG. 7.1 – (a) un arbre. (b) l'arbre (a) après une opération de fusion de  $a_1$  et  $a_3$ . (c) (d) sont les chaînes correspondantes à (a) et (b) respectivement.

Nous allons maintenant examiner en détail les opérations de relabélisation et de fusion, en termes pratiques, appliquées à des superellipsoïdes et plus particulièrement leurs coûts.

### 7.1.1 Relabélisation ou mesure de similarité entre superellipsoïdes

Si nous voulons estimer la distance d'édition entre deux descripteurs d'objets  $A$  et  $B$ , la relabélisation est la substitution d'une superellipsoïde de  $A$  par celle que l'on envisage comme correspondante à la même partie d'objet dans  $B$  à un moment donné de l'exécution de l'algorithme. La difficulté ne consiste pas en l'action triviale de remplacer une superellipsoïde par une autre mais bien d'estimer le coût d'une telle substitution. Il s'agit implicitement de comparer deux superellipsoïdes et de quantifier leurs différences.

Ainsi, la relabélisation n'est rien d'autre qu'une mesure de similarité entre superquadrriques, et c'est bien là où se trouve toute la difficulté !

Dans notre cas, nous considérons la position des superellipsoïdes entre-elles dans le descripteur comme étant gérée par les arcs du graphe. Ainsi, nous comparons les sommets (ie. les superellipsoïdes) dans un repère canonique. Nous ne tenons donc pas compte des paramètres de translation et de rotation. De plus, nous alignons les superellipsoïdes suivant leurs axes en fonction de leur taille. Nous mettons en correspondance les axes

principaux et les axes les plus petits. Cette remarque est très importante pour le reste de cette partie. Toute la méthode en dépend et ne serait pas correcte dans le cas contraire.

En 1997, L.H. Chen [CLL97] a proposé une méthode de comparaison entre deux superellipsoïdes basée sur la différence de volume. Les deux superellipsoïdes sont exprimées dans le repère canonique. Elles se chevauchent en partie. L'approche proposée par Chen est de quantifier la similarité de deux superellipsoïdes en mesurant le volume des parties qui "débordent" (Figure 7.2). Cette approche est tout à fait intuitive et, en quelque sorte, le sens propre de mesure de similarité (ou plutôt ici de mesure de différences). Pour déterminer la distance proposée, il faut donc calculer la somme des volumes des deux superellipsoïdes et lui soustraire deux fois le volume de leur intersection.

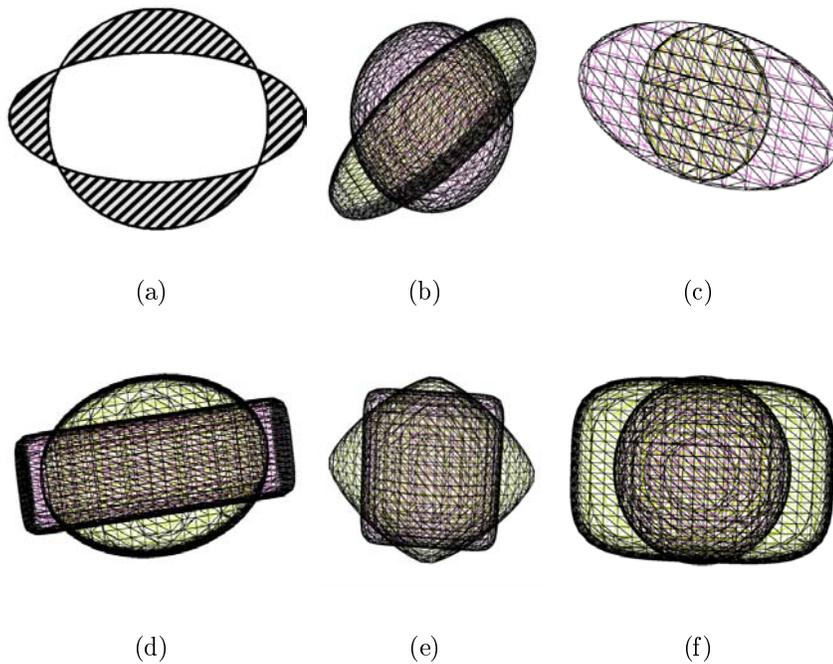


FIG. 7.2 – (a) Le volume (ici il s'agit de surface puisque la figure est en 2D) à calculer pour déterminer la distance proposée par Chen correspond à la partie hachurée sur la figure. (b)-(f) Divers exemples de mise en correspondance de superellipsoïdes. Elles sont exprimées dans leur repère canonique et alignées suivant leurs axes d'inertie.

Chen montre dans son article [CLL97] que cette mesure correspond à une métrique puisque vérifiant :

$$\begin{aligned}
 d(s_1, s_1) &= 0 \\
 d(s_1, s_2) &> 0 && \text{si } s_1 \neq s_2 \\
 d(s_1, s_2) &= d(s_2, s_1) \\
 d(s_1, s_2) &\leq d(s_1, s_3) + d(s_3, s_2)
 \end{aligned}$$

avec  $s_1, s_2$  et  $s_3$  des superellipsoïdes et  $d$  la mesure de similarité.

Néanmoins, cette mesure de similarité a un défaut majeur. Si le calcul du volume d'une superquadrique est réalisable en un temps réaliste, ce n'est pas le cas de celui de l'intersection de deux superellipsoïdes. Ainsi, la méthode de calcul que propose Chen pour mesurer cette distance est de type Monte-Carlo. C'est-à-dire que pour déterminer la distance entre deux superellipsoïdes  $s_1$  et  $s_2$ , après les avoir recentrées, il propose de calculer pour un nombre  $N$  de points  $x_i$  de coordonnées aléatoires choisies entre les coordonnées minimales et maximales des deux surfaces, les valeurs des fonctions potentielles  $f_{s_1}(x_i)$  et  $f_{s_2}(x_i)$ . Le nombre  $H$  de points dont les potentiels sont de signes opposés appartiennent à la partie hachurée de la figure puisque à l'intérieur pour l'un (signe négatif) et à l'extérieur pour l'autre (signe positif) (figure 7.2). Ainsi,  $\frac{H}{N}$  indique la distance entre  $s_1$  et  $s_2$ .

Ce calcul nécessite donc deux estimations de la fonction de potentiel d'une superellipsoïde par point calculé, et est par conséquent assez coûteux.

Une méthode pour calculer le volume d'un objet complexe de manière numérique peut être la décomposition de celui-ci en volumes élémentaires dont on sait calculer le volume. On peut, par exemple, découper le volume en cubes, mais ici comme nous disposons de la formule paramétrique des modèles, une décomposition en tétraèdres ayant tous pour sommet commun le centre de la superellipsoïde, est plus judicieuse. Vérifions si cette approche nous permet de rendre le calcul moins coûteux.

## Calcul du volume par une décomposition en tétraèdres

Si on considère deux superellipsoïdes  $Se$  et  $Se'$  dans le repère canonique de centre  $O$ , que l'on choisit trois points proches  $A$ ,  $B$  et  $C$  sur la surface de  $S$  et que l'on nomme  $A'$ ,  $B'$  et  $C'$  l'intersection avec  $Se'$  des demi-droites débutant de  $O$  et passant respectivement par  $A$ ,  $B$  et  $C$ , le volume que l'on recherche est localement la valeur absolue de la différence entre le volume de  $OABC$  et  $OA'B'C'$  (Figure 7.3a). En découpant, le volume de  $Se$  en tétraèdres on peut ainsi déterminer le volume recherché (Figure 7.3b).

La méthode fait donc deux approximations :

1. on suppose la surface comprise entre  $A$ ,  $B$  et  $C$  localement plane ( $ABC$  est un triangle plan) ;
2. ce volume est localement faux pour le tétraèdre contenant l'intersection des surfaces  $Se$  et  $Se'$  si celle-ci n'est pas confondue avec un point de l'échantillonnage.

Bien entendu, plus l'échantillonnage en tétraèdres sera fin (i.e. plus il y aura de tétraèdres), et plus l'erreur produite sera minime.

Grâce à la formule paramétrique, nous pouvons facilement obtenir les points  $A$ ,  $B$  et  $C$  sur  $Se$ , ils correspondent à  $S(\eta, \mu)$ ,  $S(\eta, \mu + \epsilon)$  et  $S(\eta + \epsilon, \mu)$ . Nous ne pourrions pas le faire aussi simplement avec un modèle implicite pur puisque c'est justement dans l'échantillonnage de sa surface que ce modèle est inefficace. Il ne nous reste alors plus

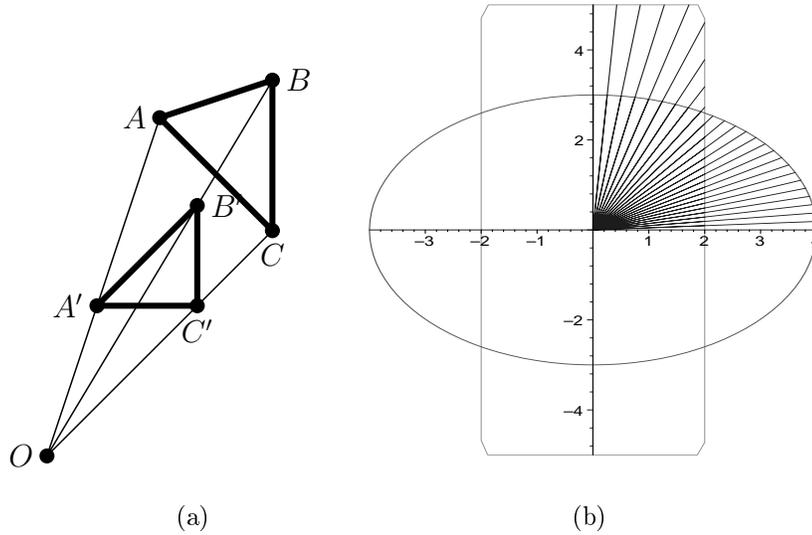


FIG. 7.3 – (a) Le volume que l'on recherche est localement la valeur absolue de la différence entre le volume de  $OABC$  et  $OA'B'C'$  (b) On découpe le volume en tétraèdres en s'aidant des formules paramétriques pour calculer le volume total recherché.

qu'à déterminer les points d'intersections  $A'$ ,  $B'$  et  $C'$  sur  $Se'$ , l'idéal serait de les obtenir de la même manière, en utilisant la forme paramétrique. Comme le montre la figure 7.4a sur deux superellipses, il ne s'agit pas d'utiliser simplement les mêmes paramètres  $\eta$  et  $\mu$  dans les formules paramétriques de  $Se$  et  $Se'$  car dans ce cas là,  $A'$  sur  $Se'$  ne correspond pas à l'intersection de  $[OA]$  avec  $Se'$ . De plus, les coefficients de courbures rendent difficilement manipulable cette forme paramétrique standard et obtenir  $\eta'$  et  $\mu'$  auxquels correspondraient le point d'intersection entre  $Se'$  et la demi-droite partant de  $O$  et passant par un point  $S(\eta, \mu)$  de  $Se$  s'avère délicat.

Nous pouvons remédier à ce problème en utilisant la formule "paramétrique régulière" (proposé par Bardinet) que nous avons déjà abordée dans le chapitre consacré aux superquadriques. Nous en rappelons la formule :

$$S'(\eta, \mu) = \begin{bmatrix} a_1 \rho(\eta, \mu) \cos(\eta) \cos(\mu) \\ a_2 \rho(\eta, \mu) \cos(\eta) \sin(\mu) \\ a_3 \rho(\eta, \mu) \sin(\eta) \end{bmatrix}, \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \mu \leq \pi \end{array}$$

avec :

$$\rho(\eta, \mu) = \left( \left( |\cos(\mu) \cos(\eta)|^{\frac{2}{\epsilon_2}} + |\sin(\mu) \cos(\eta)|^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + |\sin(\eta)|^{\frac{2}{\epsilon_1}} \right)^{-\frac{\epsilon_1}{\epsilon_2}}$$

Ainsi, comme  $\rho(\eta, \mu)$  est un scalaire, un point  $S'(\eta, \mu)$  défini par cette paramétrisation

est aligné avec le centre du repère et le point  $E(\eta, \mu)$  défini par la formule standard d'une ellipsoïde de même taille (Figure 7.4b) :

$$OS'(\eta, \mu) = \rho(\eta, \mu) * \begin{bmatrix} a_1 \cos(\eta) \cos(\mu) \\ a_2 \cos(\eta) \sin(\mu) \\ a_3 \sin(\eta) \end{bmatrix} = \rho(\eta, \mu) * OE(\eta, \mu)$$

En considérant cela, on peut déduire que  $O$ ,  $S'_{Se}(\eta, \mu)$  et  $S'_{Se'}(\eta', \mu')$  sont alignés si et seulement si  $O$ ,  $E_{Se}(\eta, \mu)$  et  $E_{Se'}(\eta', \mu')$  le sont. Or, cela signifie que les pentes de  $(OE_{Se}(\eta, \mu))$  et  $(OE_{Se'}(\eta', \mu'))$  sont égales, cela peut se traduire par l'égalité des pentes de la projection de ces droites dans le plan des courbes latitudinale et longitudinale. C'est-à-dire que :

$$\frac{E_{SeY}(\eta, \mu)}{E_{SeX}(\eta, \mu)} = \frac{E_{SeY}(\eta', \mu')}{E_{SeX}(\eta', \mu')}$$

et :

$$\frac{E_{SeZ}(\eta, \mu)}{\sqrt{E_{SeX}^2(\eta, \mu) + E_{SeY}^2(\eta, \mu)}} = \frac{E_{Se'Z}(\eta', \mu')}{\sqrt{E_{Se'X}^2(\eta', \mu') + E_{Se'Y}^2(\eta', \mu')}}$$

En identifiant  $E$  avec la formule d'une ellipsoïde :

$$E(\eta, \mu) = \begin{bmatrix} a_1 \cos(\eta) \cos(\mu) \\ a_2 \cos(\eta) \sin(\mu) \\ a_3 \sin(\eta) \end{bmatrix}$$

On peut alors en déduire que :

$$\eta' = \arctan\left(\frac{a'_1 * a_2 \tan(\eta)}{a_1 * a'_2}\right) \quad (7.1)$$

et :

$$\mu' = \arctan\left(\frac{\sqrt{(a'_1 \cos^2(\eta') + a'_2 \sin^2(\eta')) * a_3} \tan(\mu)}{\sqrt{(a_1 \cos^2(\eta) + a_2 \sin^2(\eta)) * a'_3}}\right) \quad (7.2)$$

## Comparaison des approches

La figure 7.5 montre l'évolution de la valeur de la mesure de similarité en fonction du temps alloué à son calcul. Ce temps alloué augmente en fonction du nombre de points considéré dans la méthode de Monte-Carlo et de la finesse d'échantillonnage pour la méthode de décomposition en tétraèdres.

On peut voir sur ces figures que la méthode des tétraèdres est plus efficace que la méthode de Monte-Carlo. Elle converge beaucoup plus rapidement et progresse de manière

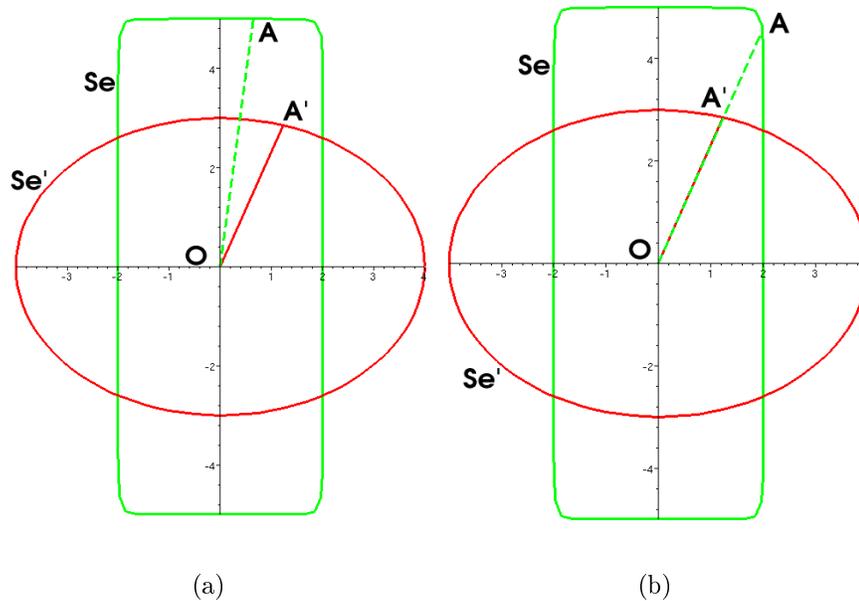


FIG. 7.4 – (a) Si on considère la paramétrisation standard, les points de même paramètre  $A$  et  $A'$  sur deux superellipses concentriques différentes respectivement  $S$  et  $S'$  ne sont pas alignés avec leur centre  $O$  (b) En considérant la paramétrisation régulière, on peut néanmoins calculer le paramètre de  $A'$  en fonction du paramètre de  $A$  tel que  $O$ ,  $A$  et  $A'$  soient alignés (cf équation 7.1 et 7.2)

plus régulière, ce qui est tout à fait logique compte tenu de l'aspect stochastique de Monte-Carlo. Cette diminution du temps de calcul grâce à la méthode des tétraèdres est même assez importante si on considère que le calcul s'arrête lorsque la valeur est relativement stable. On doit attendre en général presque 0.1 seconde dans le cas de la méthode de Monte-Carlo (ce qui correspond à 10000 points), alors que la méthode de la décomposition en tétraèdres donne de bons résultats à partir de 0.02 secondes ce qui correspond environ à 2500 tétraèdres. Notons tout de même que les résultats fournis par Monte-Carlo avant ces 0.1 secondes sont très proches de la valeur finale et que cette attente n'est donc peut-être pas nécessaire dans notre cas. Même si le découpage en tétraèdres a toujours l'avantage, les résultats des deux méthodes restent tout compte fait, si on considère la précision nécessaire à notre application, assez comparables.

Pour améliorer les choses de manière radicale, il faudrait abandonner l'idée d'un calcul itératif. De plus, pour beaucoup d'applications, la connaissance précise du volume recherché n'est pas nécessaire pour discriminer des objets. Rappelons-nous que les méthodes basées sur les geons n'utilisent qu'un faible nombre de primitives, et que c'est leur arrangement dans l'espace qui permet de déterminer la similitude entre les objets.

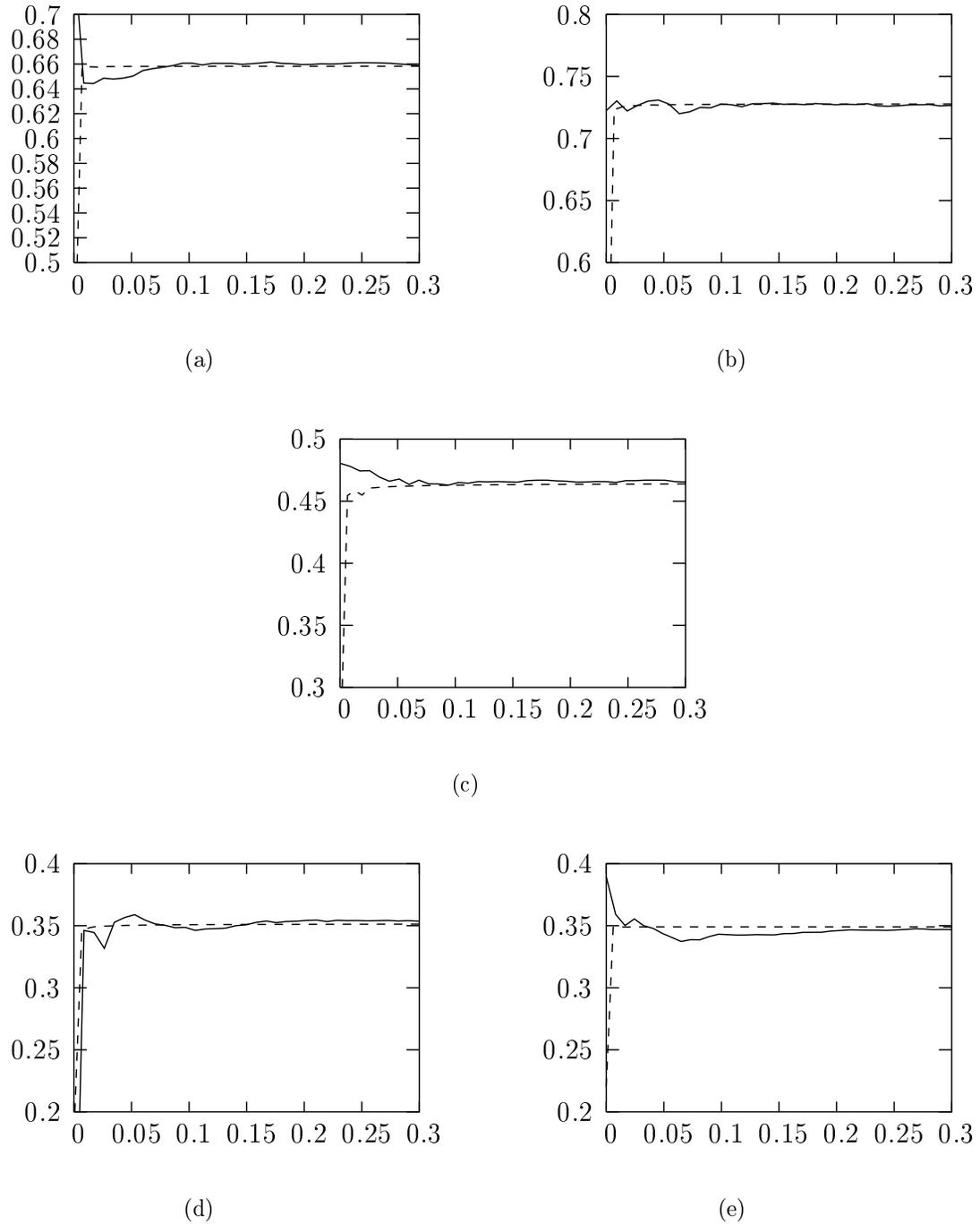


FIG. 7.5 – évolution de la valeur de la mesure de similarité en fonction du temps de calcul pour la méthode de Monte-Carlo (trait plein) et la décomposition en tétraèdres (pointillés) (a) Mesure de similarité entre un cube  $C$  tel que  $a_1 = a_2 = a_3 = 1$  et un rectangle  $R$  tel que  $a_1 = a_2 = 1$  et  $a_3 = 3$ . (b) entre le rectangle  $R$  et un cylindre tel que  $a_1 = a_2 = a_3 = 1$ . (c) entre le rectangle  $R$  et une ellipsoïde  $E$  tel que  $a_1 = a_2 = 1$  et  $a_3 = 3$ . (d) entre  $E$  et  $C$ . (e) entre  $E$  et une sphère de rayon 1.

Ainsi, toujours sur le principe de la mesure de similarité par le calcul du volume des parties non communes de superellipsoïdes superposées, nous pouvons proposer une méthode non itérative beaucoup plus rapide mais plus approximative de ce calcul.

## Calcul approximatif non-itératif du volume

Si on revient à la définition de la mesure, la distance est la somme du volume des deux superquadriques à laquelle on soustrait deux fois le volume de l'intersection. Or, si on s'intéresse aux intersections et à leurs formes, on remarque que celles-ci ne sont pas tellement irrégulières et pourraient même être modélisées par une superellipsoïde. Ainsi, si on utilise une superellipsoïde pour représenter l'intersection, il nous faut simplement savoir comment déterminer les paramètres de celle-ci et pouvoir calculer de manière rapide le volume d'une superellipsoïde quelconque.

Supposons que nous désirions mesurer la similarité de deux superellipsoïdes  $A$  et  $B$  et que nous modélisons leur intersection par une troisième  $C$ . Écartons tout de suite le cas particulier, plus que courant en pratique, où  $B$  est complètement incluse dans  $A$  (ou réciproquement), l'intersection  $C$  est alors exactement  $B$  et la distance :

$$d(A, B) = \text{volume}(A) + \text{volume}(B) - 2 * \text{volume}(B) = \text{volume}(A) - \text{volume}(B)$$

Si ce n'est pas le cas, comme  $A$  et  $B$  sont exprimées dans le repère canonique, les dimensions de  $C$  sont très simples à définir, il suffit de prendre pour chaque axe de  $C$ , le plus petit des axes correspondants dans  $A$  et  $B$ . En effet,  $A$ ,  $B$  et  $C$  étant concentriques, la taille des axes pour aller du centre à la surface de l'intersection est forcément la même que la distance pour aller du centre à la surface la plus à l'intérieur suivant un axe, et donc celle qui est la plus petite pour cet axe dans les deux superellipsoïdes.

Les coefficients de courbure sont moins évidents à déterminer. En pratique, même si les coefficients peuvent varier entre 0,01 (carré) et 2 (losange), les descripteurs sont en fait composés essentiellement de superellipsoïdes comprises entre le parallélépipède rectangle et l'ellipsoïde en passant par le cylindre, c'est-à-dire que les deux coefficients  $\epsilon_1$  et  $\epsilon_2$  sont généralement compris entre 0.01 et 1.0 Nous allons donc nous concentrer sur ces cas. Si on simplifie l'étude en se plaçant dans le cas 2D des superellipses (figure 7.6), on peut remarquer que si  $A$  et  $B$  ont des coefficients de courbure compris entre 0.01 et 1.0 alors  $C$ , qui modélise leur intersection, aura aussi des coefficients de courbure compris entre 0,01 et 1, si  $A$  et  $B$  ont des coefficients de courbure égaux à 0.01 alors  $C$  aussi (l'intersection de deux rectangles est un rectangle). Il faut en fait déterminer les coefficients de courbure en fonction la manière plus ou moins abrupte où l'intersection a lieu. Celle-ci dépendant à la fois des coefficients de courbures mais aussi de la taille (et du rapport des axes) des superellipsoïdes  $A$  et  $B$ . Ainsi, nous proposons d'utiliser pour chaque coefficient

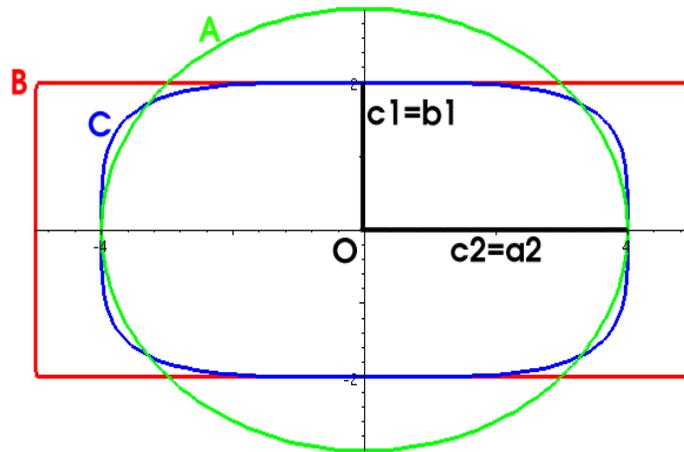


FIG. 7.6 – Modélisation de l'intersection de deux superellipses  $A$  et  $B$  par une superellipse  $C$ . Les paramètres de taille de  $C$  ( $c_1$  et  $c_2$ ) sont facilement calculables il suffit de prendre les plus petits des paramètres correspondants pour  $A$  et  $B$ . Il reste alors à déterminer le coefficient de courbure de  $C$ .

de courbure l'équation suivante (Figure 7.7) :

$$\epsilon_C = \frac{\epsilon_A + \epsilon_B}{2} * \frac{\min(A_2, B_2)}{\max(A_2, B_2)} \quad (7.3)$$

Ce fait est visible particulièrement sur les figures 7.7b et 7.7c. Le second facteur module cette moyenne en fonction de la taille des axes secondaires. En effet, quand la superellipse moins longue est beaucoup plus large que la superellipse la plus longue alors l'intersection aura une courbure forte. Ce second fait peut être observé dans la figure 7.7e. Dans cet exemple, bien que les deux superellipses sont en fait des ellipses, leur intersection est proche du rectangle.

Notons que ces remarques et cette équation est valable seulement parce que nous nous sommes placés dans le cas de superellipsoïdes superposées alignées par rapport à leurs axes principaux d'inertie.

Dans le cas 3D des superellipsoïdes, on utilisera simplement cette formule pour déterminer la courbure latitudinale de  $C$  (resp. longitudinale) en fonction des courbures latitudinales (resp. longitudinales) de  $A$  et  $B$ .

Il ne nous reste plus qu'à déterminer le volume d'une superellipsoïde pour pouvoir estimer la mesure de similarité. Dans son ouvrage sur les superquadriques [JLS00], Solina montre que ce volume est :

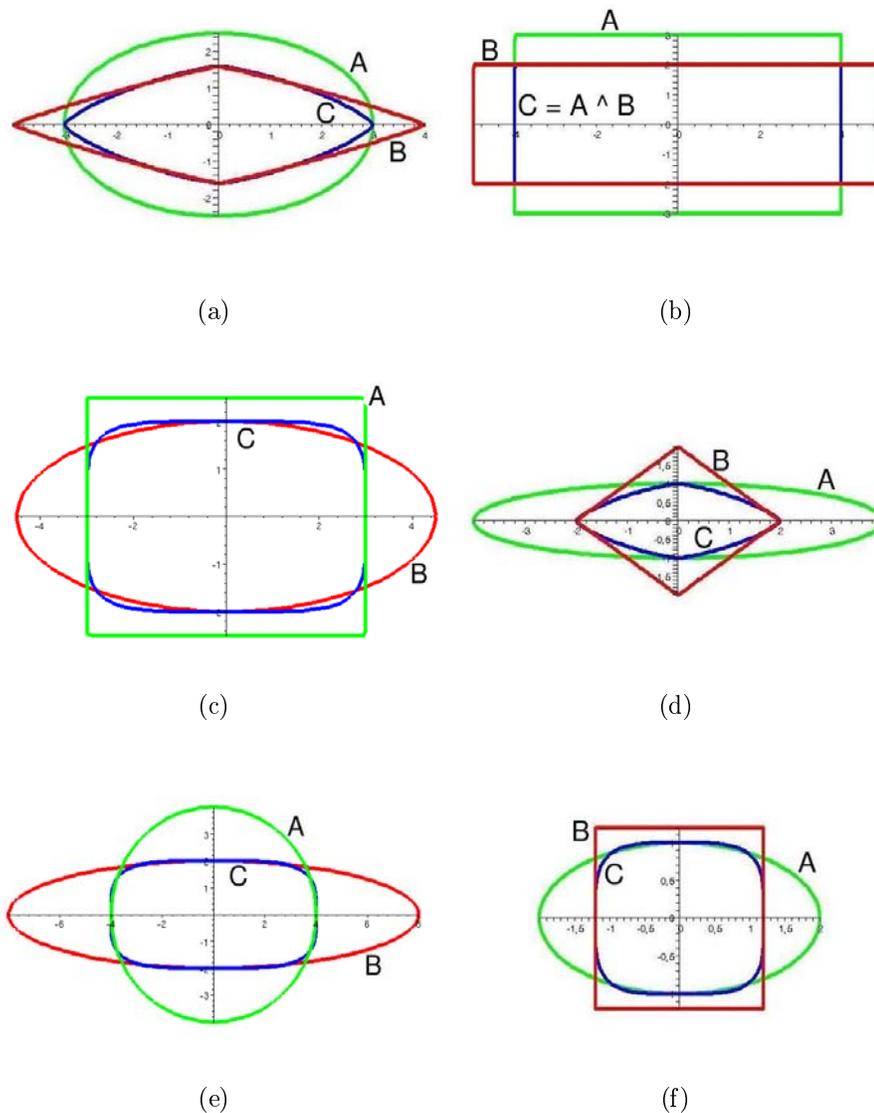


FIG. 7.7 – Exemples de modélisation d'intersection par une superellipsoïde appliquée à des superellipses de formes diverses.

$$V = 2a_1a_2a_3\epsilon_1\epsilon_2 B\left(\frac{\epsilon_1}{2} + 1, \epsilon_1\right) B\left(\frac{\epsilon_2}{2}, \frac{\epsilon_2}{2}\right) \quad (7.4)$$

avec :

$$B(x, y) = 2 \int_0^{\pi/2} \sin^{2x-1} \phi \cos^{2y-1} \phi d\phi$$

Dans un souci de rapidité, nous ne pouvons nous permettre de calculer l'intégrale de la fonction  $B(x, y)$  pour chaque mesure. Cette fonction étant appelée seulement avec  $\epsilon_1$  et  $\epsilon_2$ , et eux-mêmes variant entre 0.01 et 2, on aura donc  $x \in [0.005, 2]$  et  $y \in [0.005, 2]$ . Pour

accélérer le processus, on va échantillonner cette fonction sur ces intervalles (suivant un pas de 0.2, donc cent valeurs, semble plus que suffisant), et n'utiliser que des interpolations linéaires entre les valeurs de cette table pour le calcul du volume. Le calcul de notre mesure de similarité est alors extrêmement rapide car non seulement elle n'est plus itérative, mais en plus, elle est composée uniquement de simples additions, multiplications et divisions.

## Variations de la mesure de similarité

Les courbes de la figure 7.8 montrent les variations de la mesure similarité lorsque l'on compare une superellipsoïde de référence avec une autre dont on fait varier un ou des paramètres. La superellipsoïde de référence que nous avons choisie est un cube de côté 1 (c'est-à-dire une superellipsoïde telle que  $a_1 = a_2 = a_3 = 0.5$  et  $\epsilon_1 = \epsilon_2 = 0.01$ ).

La courbe 7.8a a été construite en comparant le cube avec un parallélépipède rectangle de même section mais dont la longueur varie. Elle confirme que la "distance" entre ces superellipsoïdes croît de manière linéaire par rapport à la taille du parallélépipède. On a bien sûr un minimum de distance à 0 en 0.5 puisque les deux superellipsoïdes sont identiques.

La courbe 7.8b montre les variations de la mesure de similarité en fonction de la courbure d'une superellipsoïde dont la taille (les paramètres  $a_1$ ,  $a_2$  et  $a_3$ ) reste égale à celle du cube de référence.

Pour la dernière courbe (figure 7.8c), le cube est comparé à un ensemble de superellipsoïdes dont la longueur et la courbure varient simultanément suivant des procédés similaires aux deux premières expériences. Si les deux figures précédentes étaient facilement interprétables et compréhensibles, celle-ci l'est un peu moins. À partir d'un certain degré de courbure, la distance entre le cube et la superellipsoïde va diminuer alors que la longueur de cette dernière augmente.

Bien sûr d'autres approches peuvent être employées pour quantifier la similitude entre deux superellipsoïdes. On peut notamment penser à une comparaison directe des paramètres en considérant, par exemple, le rapport de taille des axes et les coefficients de courbures. L'avantage de ces méthodes est, bien sûr, d'être vraiment peu coûteuse en temps de calcul. Mais, leur inconvénient est de nécessiter la définition et le calibrage, toujours très laborieux, d'un grand nombre de coefficients afin de pouvoir corrélérer les mesures de différence de tous les paramètres. On peut aussi penser à une simplification des primitives à la manière d'une approche *geons*. On remplacerait alors chaque superellipsoïde par la primitive la plus proche d'un catalogue prédéfini.

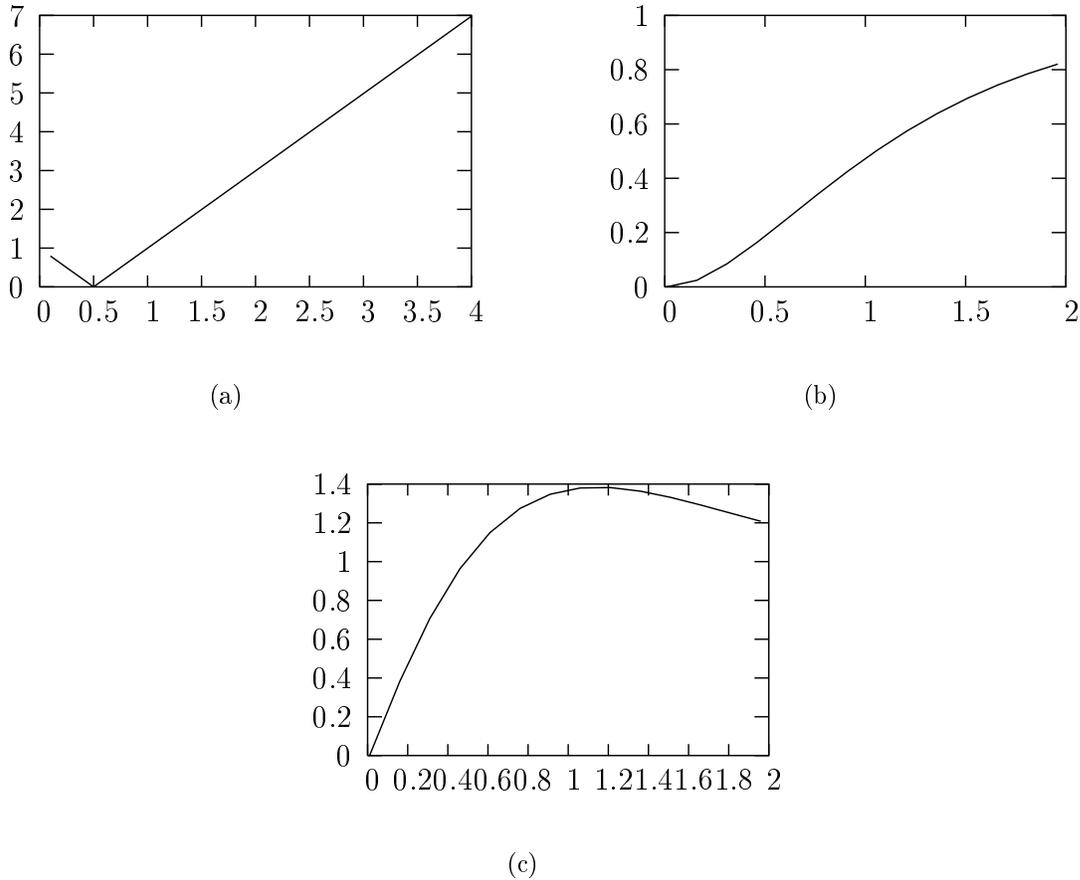


FIG. 7.8 – Variation de la mesure de similarité entre un cube de coté 1 ( $a_1 = a_2 = a_3 = 0.5$  et  $\epsilon_1 = \epsilon_2 = 0.01$ ) (a) Avec un parallélépipède rectangle tel que  $a'_1 = a'_2 = 0.5$  et  $\epsilon'_1 = \epsilon'_2 = 0.01$  en fonction de  $a'_3 \in [0.2, 6]$  (b) Avec une superellipsoïde telle que  $a'_1 = a'_2 = a'_3 = 0.5$  en fonction de  $t \in ]0, 2]$  et avec  $\epsilon'_1 = \epsilon'_2 = t$  (c) Avec une superellipsoïde telle que  $a'_1 = a'_2 = 0.5$  en fonction de  $t \in ]0, 2]$  et avec  $\epsilon'_1 = \epsilon'_2 = t$  et  $a'_3 = \frac{4}{3}t + 0.5$  (on a donc  $a'_3 \in ]0.5, 3.1666]$ )

### 7.1.2 Fusion de superellipsoïdes

La fusion de deux superellipsoïdes est la seule opération qui nous permette de modifier la structure du descripteur. Cette fusion ne peut avoir lieu que sur des superellipsoïdes adjacentes c'est-à-dire pour lesquelles il existe un arc entre les sommets correspondants dans le graphe associé au descripteur. En terme de chaîne de caractères, la fusion ne peut se produire que sur les sommets représentés par les deux parenthèses en bout de chaîne si elles correspondent à deux sommets différents (c'est-à-dire si la chaîne est du type  $A_0(iA_1(jA_2)_j)_i$  et non du type  $A_0(j)_j$ ).

Tout le problème est alors d'estimer les paramètres d'une nouvelle superellipsoïde  $k$ ,

fusion des deux sommets traités  $i$  et  $j$ , de définir un coût de passage de  $ij$  à  $k$  et de modifier la chaîne :

$$A_0(iA_1(jA_2)_j)_i \longrightarrow A_0(kA_1A_2)_k$$

En ce qui concerne la fusion proprement dite, c'est-à-dire l'obtention d'une seule superellipsoïde à partir de deux, la méthode que nous avons retenue est basée sur l'ellipsoïde d'inertie. Si on considère le système d'objets  $S$  formés par les deux superellipsoïdes  $A$  et  $B$  que nous voulons fusionner en une superellipsoïde  $C$ , nous proposons d'utiliser le centre de gravité de  $S$  comme centre de  $C$ , et les axes principaux d'inertie de  $S$  comme axes de  $C$ .

Pour calculer l'ellipsoïde  $C$  équivalente à la fusion de  $A$  et  $B$ , il nous faut estimer :

- les paramètres de tailles  $a_1$ ,  $a_2$  et  $a_3$  ;
- les paramètres de translation  $t_x$ ,  $t_y$  et  $t_z$  ;
- les paramètres de rotation  $r_x$ ,  $r_y$  et  $r_z$ .

Pour calculer ces paramètres, nous ne pouvons pas nous baser sur les axes et le centre de  $A$  et  $B$ . Nous voulons dire par là que s'il paraît intuitivement simple et logique d'utiliser les axes de  $A$  et  $B$  pour obtenir les axes de  $C$ , la réalité est, en fait, tout autre et cette voie est plutôt résistante. Ainsi, nous allons repasser par des points situés sur la surface de  $A$  et  $B$ . Nous n'utiliserons pas les points réels (i.e. les données qui nous ont permis d'obtenir le descripteur) de la surface de l'objet car nous désirons que la comparaison n'utilise que les descripteurs, et nous n'allons pas non plus ré-échantillonner de manière précise les deux superellipsoïdes. Comme dans le cas de la mesure de similarité, la méthode doit être rapide.

Ainsi, nous utiliserons six points seulement pour chaque superellipsoïde. Ces points sont ceux se trouvant à l'extrémité de chaque axe du repère intrinsèque de la superellipsoïde considérée. Ceux-ci sont très simples à calculer, nul besoin de la formule paramétrique. Pour une superellipsoïde de taille  $a_1$   $a_2$   $a_3$ , ils correspondent à  $(a_1, 0, 0)$ ,  $(-a_1, 0, 0)$ ,  $(0, a_2, 0)$ ,  $(0, -a_2, 0)$ ,  $(0, 0, a_3)$  et  $(0, 0, -a_3)$  (quelle que soit la valeur des coefficients de courbure  $\epsilon_1$  et  $\epsilon_2$ ) auxquels il faut appliquer la matrice de transformation (rotation et translation) que subit normalement la superquadrique, puisqu'ici nous nous intéressons à la combinaison de deux superellipsoïdes et de leur position l'une par rapport à l'autre. Pour pallier ce sous-échantillonnage, nous pondérons ces points par le volume (obtenu comme expliqué dans le paragraphe précédent) de la superellipsoïde à laquelle il appartient.

Soit  $A_i$  et  $B_i$  avec  $i \in [1, 6]$ , les six points échantillonnés respectivement sur  $A$  et  $B$ . Nous définissons le centre de  $C$  (et donc les paramètres de translations  $t_x$ ,  $t_y$  et  $t_z$  de  $C$ ) par :

$$C_o = \frac{\sum_{i=0}^6 \text{volume}(A) * A_i + \sum_{i=0}^6 \text{volume}(B) * B_i}{6 * (\text{volume}(A) + \text{volume}(B))}$$

On calcule ensuite la matrice des moments centraux d'inertie d'ordre 2 du système des douze points :

$$M = \frac{(\text{volume}(A) \sum_{i=0}^6 (A_i - C_o)(A_i - C_o)^T + \text{volume}(B) \sum_{i=0}^6 (B_i - C_o)(B_i - C_o)^T)}{6 * (\text{volume}(A) + \text{volume}(B))}$$

Puis on détermine les paramètres d'échelle  $(a_1, a_2, a_3)$  et de rotation  $(\rho, \theta, \phi)$  de  $C$  grâce aux valeurs et aux vecteurs propres de  $M$  comme nous l'avons déjà fait lors de l'estimation des paramètres d'une ellipsoïde par un nuage de points 3D pour initialiser l'approximation (cf paragraphe 3.1.1).

Nous avons alors tous les paramètres de l'ellipsoïde  $(a_1, a_2, a_3, t_x, t_y, t_z, \rho, \theta, \phi)$ , il ne nous reste plus qu'à estimer les paramètres de courbures  $\epsilon_1$  et  $\epsilon_2$ , spécifiques aux superquadriques. Cependant, nous n'avons pas vraiment de matière pour les estimer. Rappelons-nous que c'était déjà le cas lors de l'initialisation de l'approximation où nous avons arbitrairement fixé ces paramètres à 1.0 (cf paragraphe 3.1.1).

Si le système  $S$  est formé par une superquadrique cubique énorme et un tout petit cylindre, le résultat de la fusion devra être logiquement un cube, le cylindre étant considéré comme négligeable (peut-être un détail de l'objet ou une erreur de la segmentation). Mais, si les deux objets sont de tailles similaires, que l'un est un cube et l'autre une sphère, quelle forme donner à la superellipsoïde fusion ?

Comme paramètres de courbures  $\epsilon_1$  et  $\epsilon_2$  de la superellipsoïde  $C$  résultante, nous utilisons la moyenne des paramètres de courbures de  $A$  et  $B$ , ici aussi, pondéré par leur volume.

$$\epsilon_{1C} = \frac{\text{volume}(A) * \epsilon_{1A} + \text{volume}(B) * \epsilon_{1B}}{\text{volume}(A) + \text{volume}(B)}$$

et

$$\epsilon_{2C} = \frac{\text{volume}(A) * \epsilon_{2A} + \text{volume}(B) * \epsilon_{2B}}{\text{volume}(A) + \text{volume}(B)}$$

En ce qui concerne le coût de cette fusion, l'idéal est de garder l'idée de différence de volume et de mesurer cette différence entre le volume occupé par l'ensemble formé des deux superellipsoïdes initiales et le volume de celle qui résulte de leur fusion. Outre le fait d'être assez représentative de la dégradation du descripteur, cette approche a pour avantage de permettre une comparaison aisée entre les opérations d'édition puisque leur coût serait

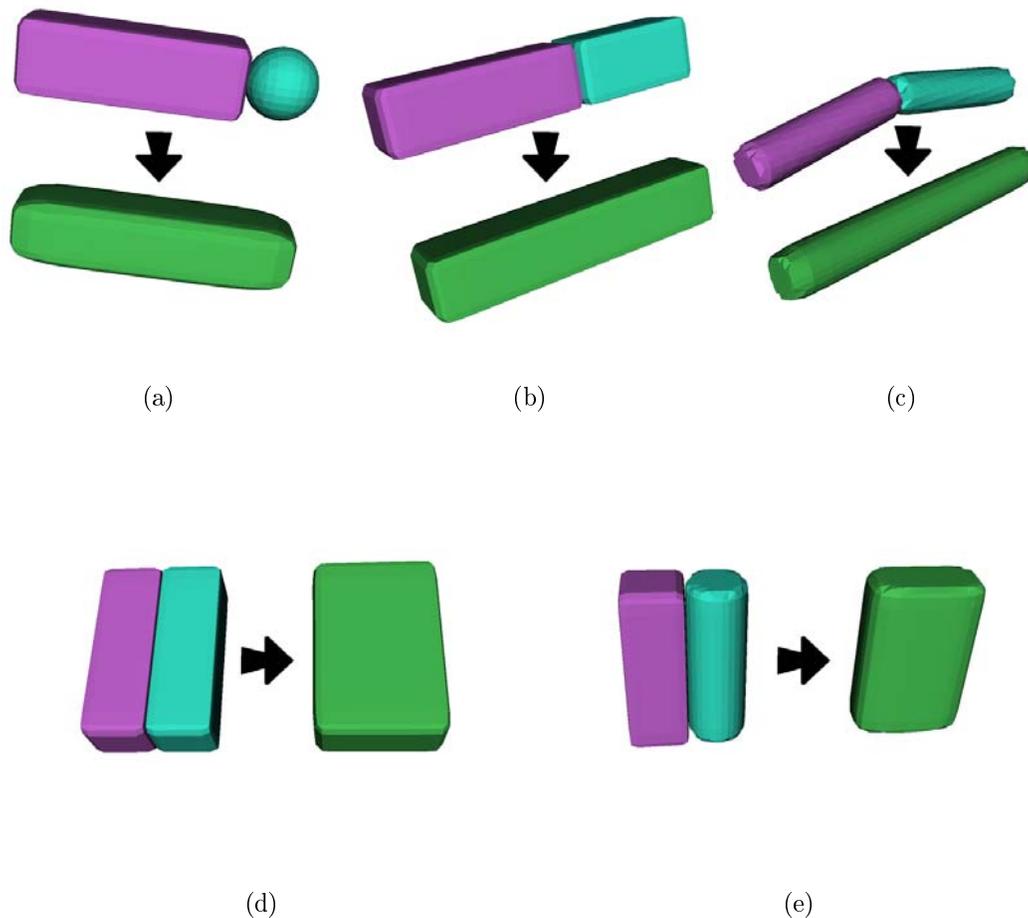


FIG. 7.9 – Exemple de fusion de superellipsoïdes sur des cas simples.

basé sur le même principe. Bien sûr, cela ne signifie pas que la même augmentation de volume dans le cas d'une relabélisation et d'une fusion entraîne le même coût et il peut être judicieux de pondérer ces coûts pour privilégier l'une ou l'autre des opérations d'édition.

Le calcul de cette différence de volume n'est cependant pas simple. En effet, le volume occupé par l'ensemble des deux superellipsoïdes est difficilement estimable à cause de la possibilité de chevauchement (comme on peut le voir sur la figure 7.10, dont les exemples sont tirés de descripteurs réels, le volume de l'intersection, s'il existe, reste tout de même assez faible). Une possibilité est d'utiliser une méthode similaire à celle de Chen, de type Monte-Carlo, mais comme pour la relabélisation, cela est coûteux, d'autant plus qu'il faut ici calculer la valeur du potentiel de trois superellipsoïdes ce qui augmente de 50% le temps de calcul. De plus, l'espace de recherche de l'algorithme est aussi beaucoup plus vaste puisque qu'aucune simplification pour des raisons de symétrie ne peut ici être envisagé et donc le volume englobant les trois superellipsoïdes concernées doit être considéré

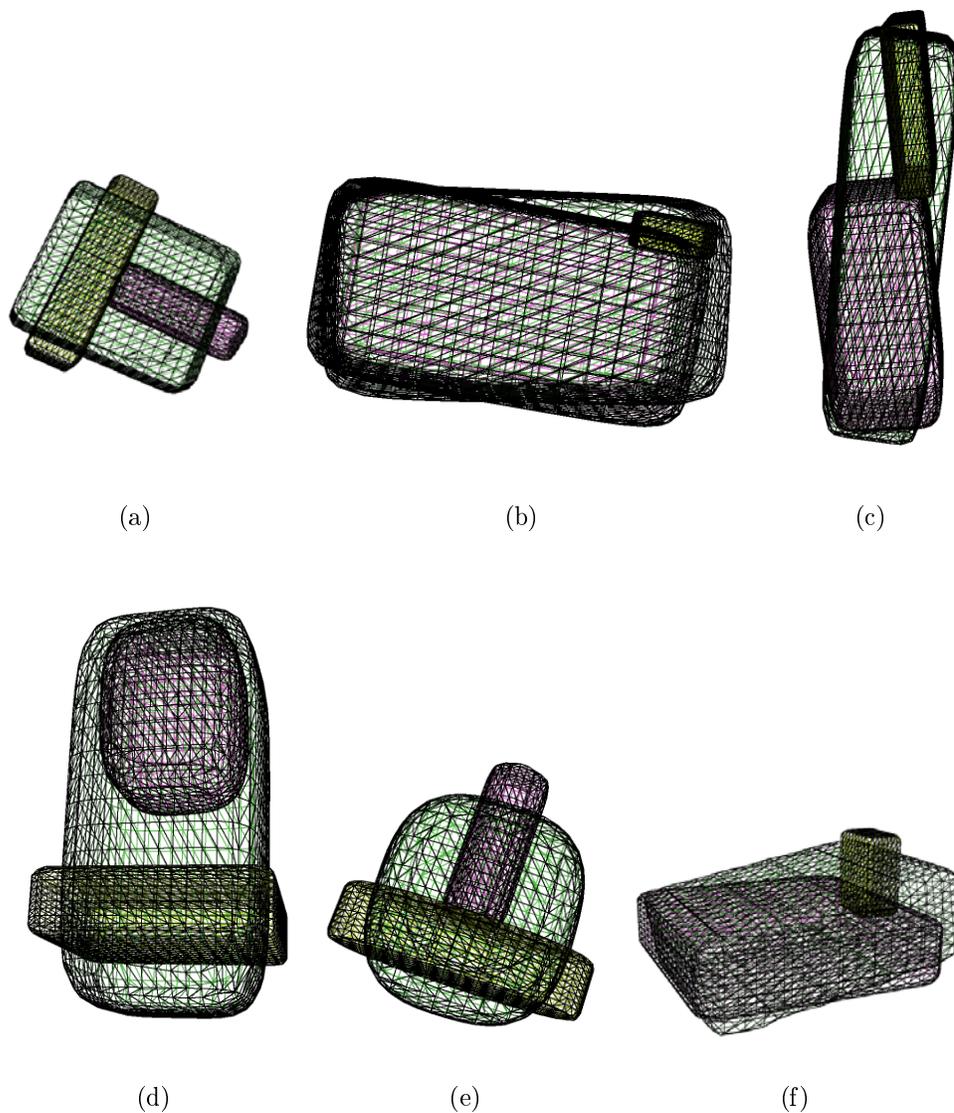


FIG. 7.10 – Exemples de fusion de superellipsoïdes moins triviaux que la figure 7.9. Ces cas réels sont tirés de la comparaison du modèle chaise. Les superquadriques vertes représentent le résultat des fusions des superquadriques jaunes et des superquadriques roses.

dans sa totalité. Notons aussi que les méthodes développées dans le cadre de la relabélisation (méthode des tétraèdres et méthode de l'approximation de l'intersection par une superellipsoïde) ne sont pas applicables ici, car elles nécessitent que les superquadriques soient concentriques.

Toujours en considérant la fusion de deux superellipsoïdes  $A$  et  $B$  en une superellipsoïde  $C$ , et pour gagner en temps de calcul on peut réutiliser la formule du volume définie

précédemment (cf équation 7.1.1) et se limiter à calculer :

$$\text{CoûtFusion}(A, B, C) = |\text{Volume}(A) + \text{Volume}(B) - \text{Volume}(C)|$$

Bien sûr cela suppose que l'intersection entre  $A$  et  $B$  soit nulle, ou du moins négligeable. Cela n'est pas si faux dans le cas où le descripteur a été obtenu grâce à notre méthode de segmentation de type *découpage-fusion*.

Il faut noter que si nous ne fixons pas de limite, cette méthode permet de fusionner tout couple adjacent de superellipsoïdes, tendant à transformer tout descripteur d'objet en une seule superellipsoïde sans grand intérêt. Il peut donc être intéressant d'interdire certaines fusions. Ces dernières sont celles dont on sait que les parties en présence ne sont pas liées à des légères différences de description, mais bel et bien des superellipsoïdes correspondant à des parties réelles et bien distinctes de l'objet.

Ces interdictions ne doivent être mises en place uniquement dans une volonté de garder un certain sens, une intégrité au descripteur. Mais cela peut conduire à négliger certaines possibilités en fixant certaines parties de l'objet de manière assez arbitraire. Notons aussi que la méthode aura pour conséquence d'estimer certaines distances entre descripteurs comme infinies ou non quantifiables, ce qui bien entendu rendra impossible leur classement. C'est pourquoi il faudrait alors peut-être prendre en considération non pas seulement l'"impossibilité" de fusion, mais aussi la taille de la partie d'objet concerné. En effet, cela serait une erreur d'abandonner la comparaison de deux objets à cause de la fusion de deux superquadriques négligeables devant le reste de l'objet.

On peut aussi penser que ces interdictions accéléreraient la comparaison de descripteurs. En effet, si lors de la comparaison de deux arbres couvrants, une fusion retourne un coût infini, on peut arrêter la comparaison, retourner une distance infinie pour cet arbre couvrant puis passer au suivant. Cependant, ce type de "court-circuit" existe déjà dans notre implémentation de l'algorithme. Si au cours de l'examen d'un arbre couvrant la distance devient plus grande que la distance minimum obtenue avec les précédents, l'examen est bien entendu abandonné et on passe au suivant. Le gain n'est donc pas exceptionnel.

Interdire certaines fusions est donc un choix qui peut être fait pour empêcher une perte de sens du descripteur et donc de la comparaison, mais ce choix comporte certaines contraintes. Aussi, en définissant en conséquence le rapport des coûts entre la fusion et la relabélisation, l'interdiction doit pouvoir être évitée.

## 7.2 Test de validations

Dans cette partie, nous présentons les tests de validations de la méthode de comparaisons de descripteurs dans une application de type indexation.

La base de test est constitué de 27 objets divisés en trois classes :

- la classe **Avions** contient 12 objets dérivant de 3 modèles originaux.
- la classe **Chaise1** contient 7 objets de chaise à un pied soit dérivants de l'objet présenté dans la figure 4.12 de la page 79 soit construits synthétiquement.
- la classe **Chaise2** contient 8 objets représentant des chaises à quatre pieds construits synthétiquement.

Dans chacune de ces classes, on peut retrouver les mêmes modèles sous forme simple, bruitée, ou altérée par des fusions forcées ou retenues. L'ensemble des objets de cette base de test sont présentés dans les figures 7.11, 7.12 et 7.13.

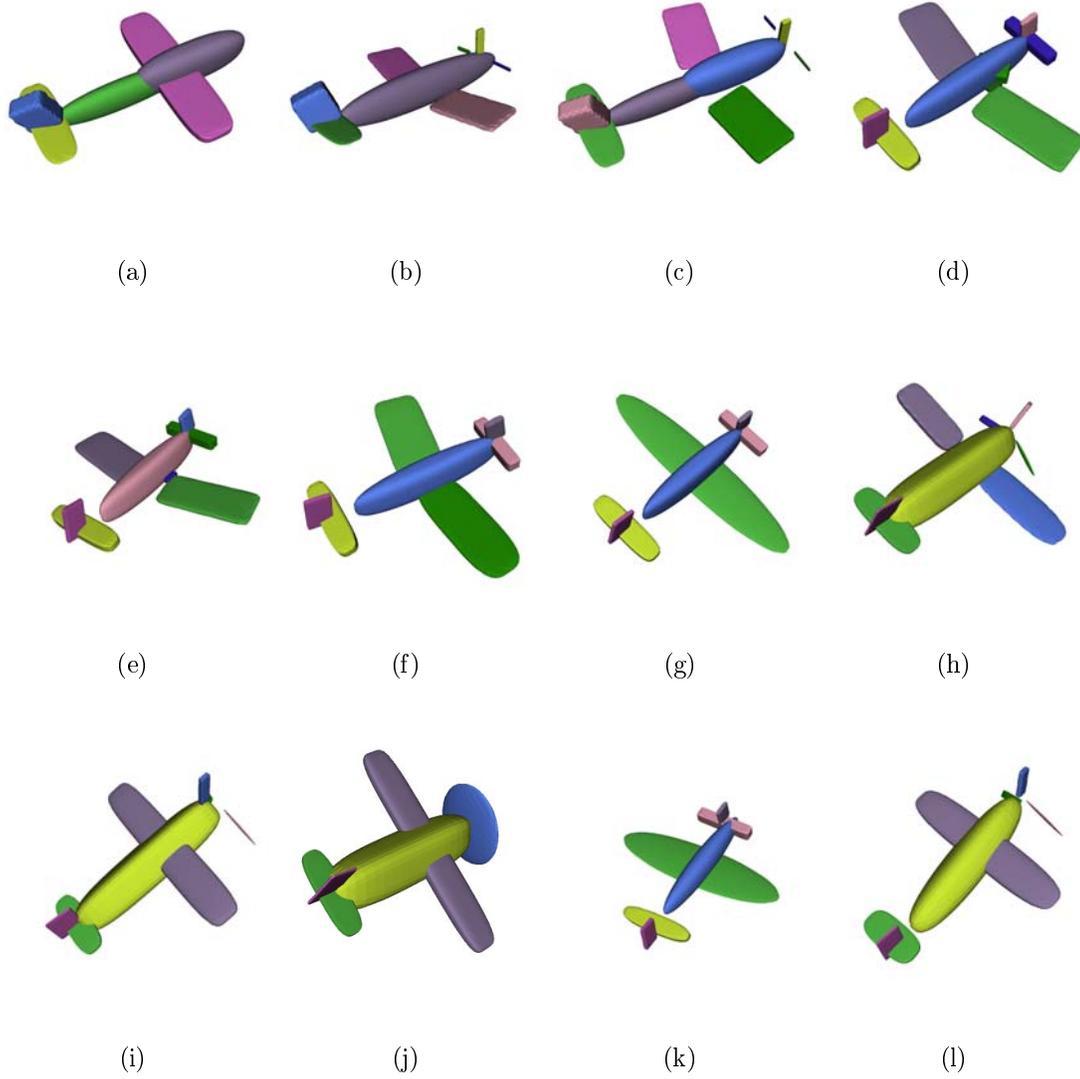


FIG. 7.11 – Descripteurs d'objets de la base de test : la classe Avions

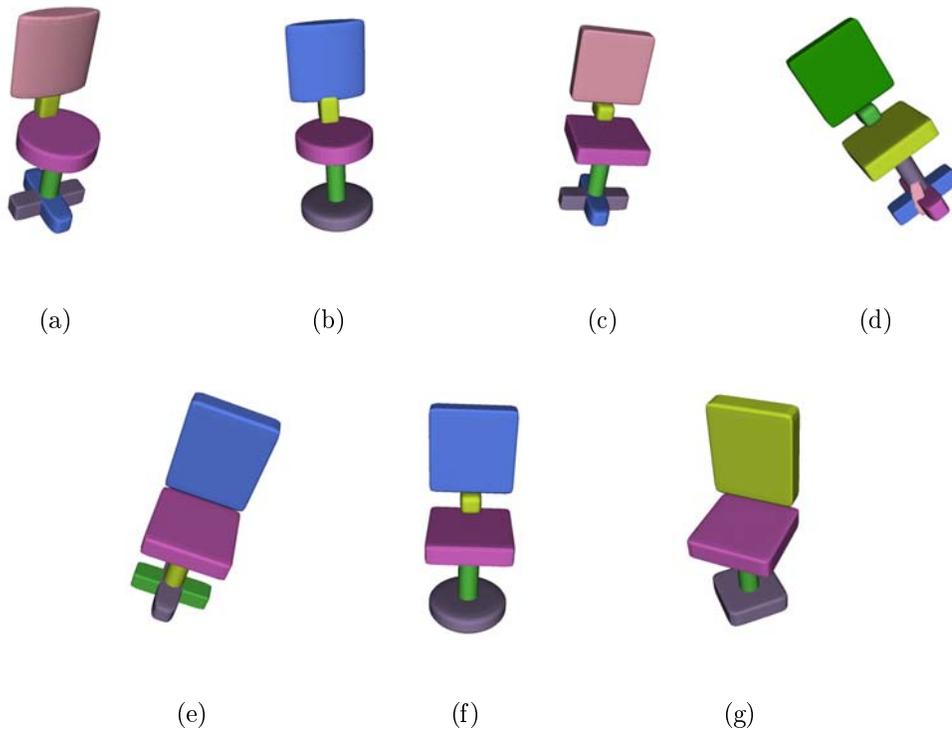


FIG. 7.12 – Descripteurs d'objets de la base de test : la classe Chaise1

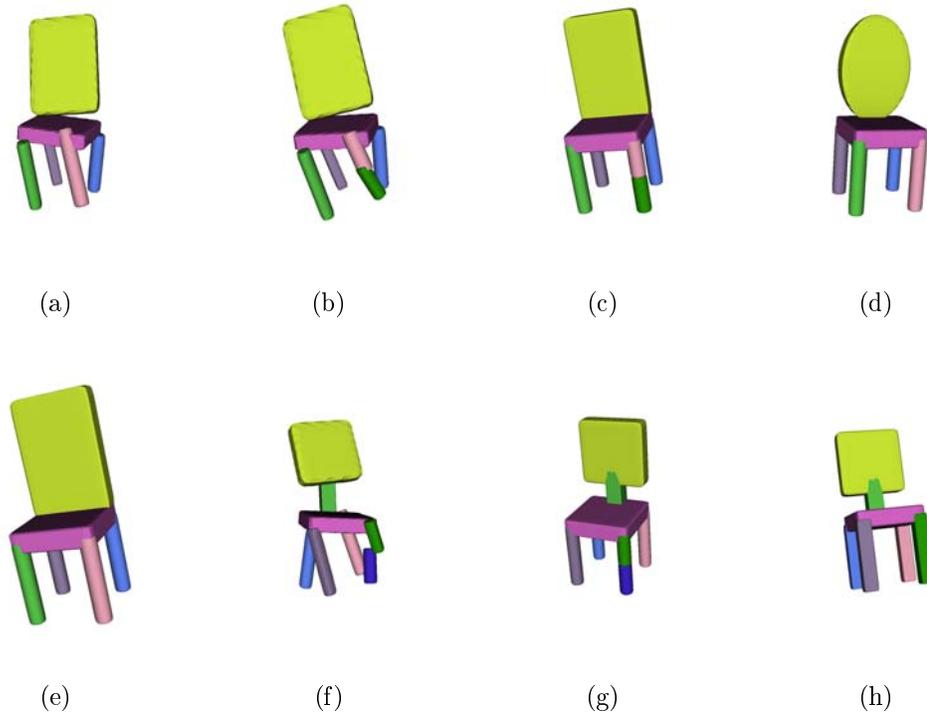


FIG. 7.13 – Descripteurs d'objets de la base de test : la classe Chaise2

Le principe de l'expérience est de fournir à une l'application un descripteur requête que celle-ci va comparer à tous les objets présents dans la base. Il en résultera la liste de tous les objets triés suivant la valeur croissante de leur distance d'édition avec le descripteur requête. Le premier élément de la liste est donc l'objet le plus ressemblant à l'objet requête et le dernier le plus éloigné au sens des critères que nous avons définis dans ce chapitre.

La notion de *rappel* a été proposée en 1955 par Kent, Berry, Luehrs, et Perry et vise à mesurer les performances d'un système d'indexation. Le *rappel* est le rapport entre le nombre réponses pertinentes (ie. objets de la classe de la requête dans la réponse) et le nombre total de réponses pertinentes possibles présentes dans la base de données (ie. cardinal de la classe de la requête).

Par exemple, pour une classe de 10 éléments, la courbe de rappel d'un système d'indexation idéal est strictement croissante jusqu'à l'abscisse 10 où la valeur de ce rappel atteint 100% puis horizontale. Ce qui signifie que les réponses pertinentes ont une distance inférieure à celle de tous les autres objets de la base et sont donc retournées en premier. La figure 7.14 présente ce rappel pour notre méthode et testé sur cette base en fonction de la taille de la liste d'objets réponses retournée. Cette courbe s'approche du profil idéal.

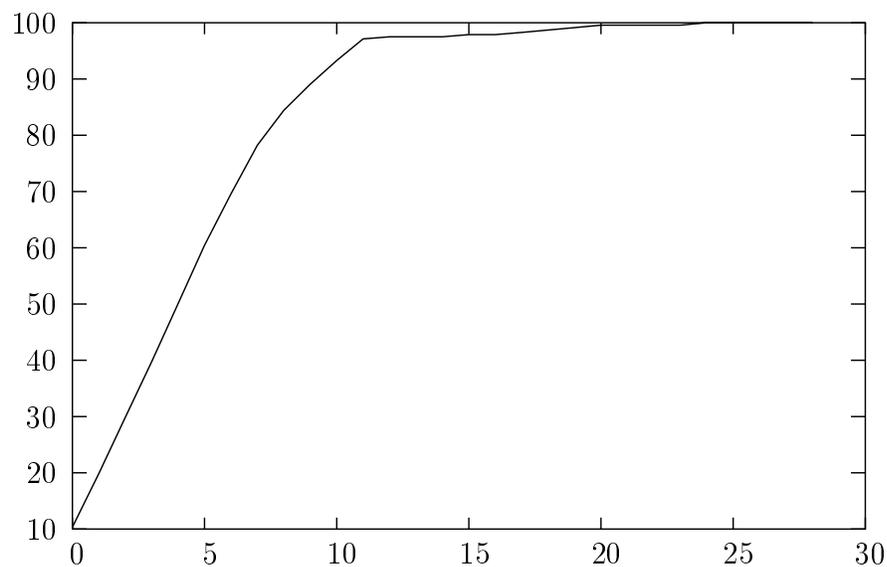


FIG. 7.14 – Courbe de rappel en fonction du nombre de d'objets

Une autre représentation des résultats est proposée dans la figure 7.15. Celle-ci est une matrice dite de classification où la case  $(i, j)$  correspond à la transposition en niveaux de gris du rang de l'objet  $j$  dans la liste triée par ordre croissant de la distance des comparaisons des objets de la base avec  $i$ . Plus une case est sombre et plus l'objet  $j$  est considéré comme similaire à l'objet  $i$ . Les objets sont présentés dans le même ordre pour

les lignes et les colonnes et sont triés suivant leur classe. Les cases de la diagonale sont donc noires puisqu'elles correspondent à la comparaison d'un objet avec lui-même ; ce qui donne naturellement le rang le plus faible (0). Dans notre cas, on a dans l'ordre de gauche à droite et de bas en haut : la classe Avions, la classe Chaise1, puis la classe Chaise2. Par sa définition, il est normal que la matrice ne soit pas symétrique (les valeurs étant celles du rang et non de la distance).

On peut clairement voir des carrés sombres sur la diagonale qui correspondent aux comparaisons intra-classes des objets. Comme le montrait déjà la courbe du rappel, cela confirme le bon comportement de notre méthode de comparaison. Lorsque la requête est un objet d'une classe donnée, sa comparaison avec les objets de cette classe donne les plus faibles distances. On peut remarquer que si les classes Avions et Chaise2 sont très bien définies la classe Chaise1 l'est moins. Les premiers objets de cette dernière sont évalués comme proches de la classe Chaise2, ce qui est assez logique puisque ces objets ont une partie de leur modèle (siège et dossier) très voisine.

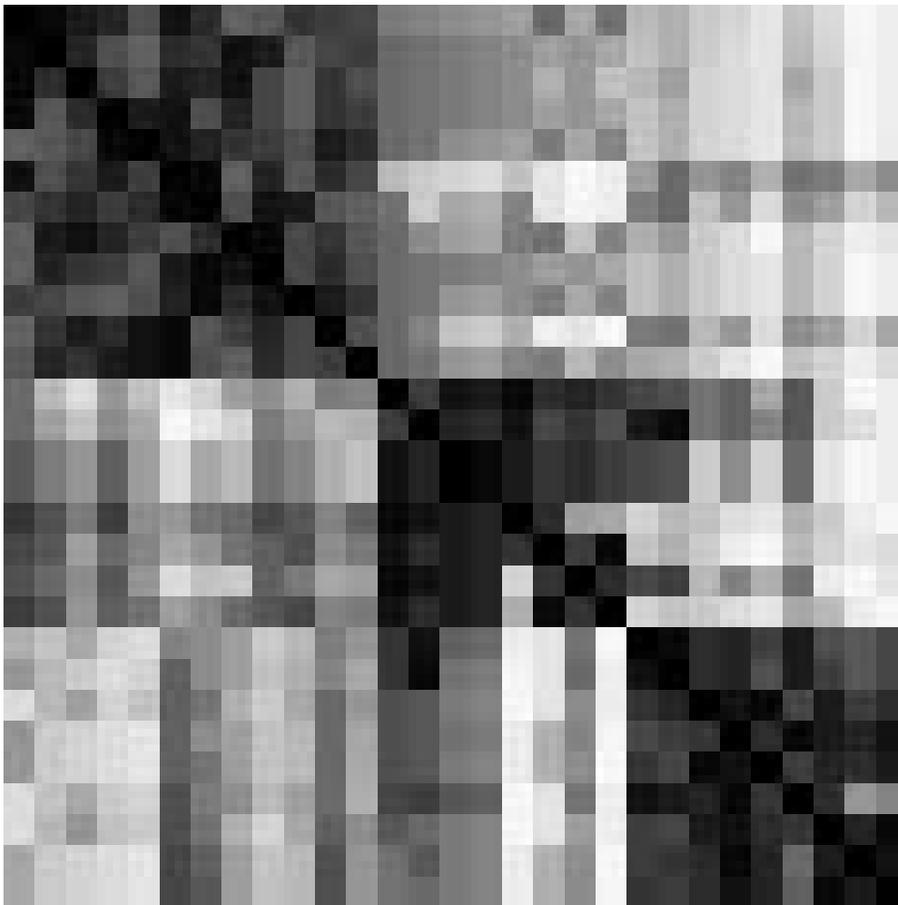
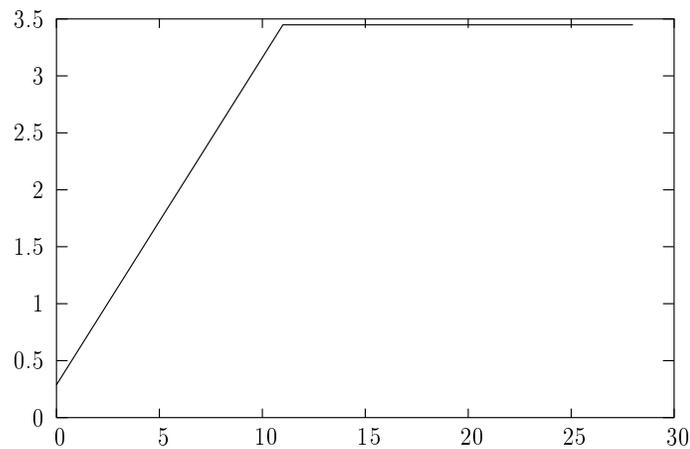


FIG. 7.15 – Matrice de classification

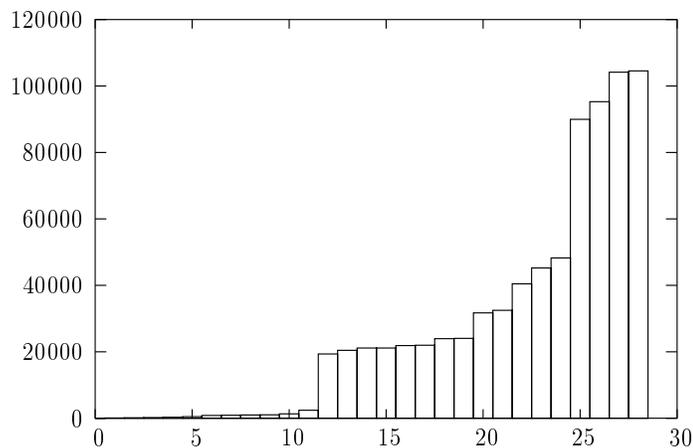
Les deux premières figures donnaient une vue globale du comportement de notre méthode de comparaison. Les figures 7.16, 7.17 et 7.18 s'intéressent à des requêtes en par-

ticulier. Les figures d'indice (a) représentent en quelque sorte le *rappel* pour la requête considérée. Les figures d'indice (b) montrent l'évolution de la valeur de la distance d'édition en fonction du rang de l'objet pour cette requête.

La figure 7.16 montre de bons résultats obtenus à partir de la requête d'un modèle de la classe Avions et plus exactement celui de la figure 7.11a. La figure 7.16a confirme que tous les objets de la classe ont bien des rangs inférieurs aux autres objets de la base, tandis que la figure 7.16b souligne un écart net au niveau de la valeur de la distance d'édition entre les objets de la classe avion et les autres (rang 12). Ces deux comportements sont tout à fait souhaitables.



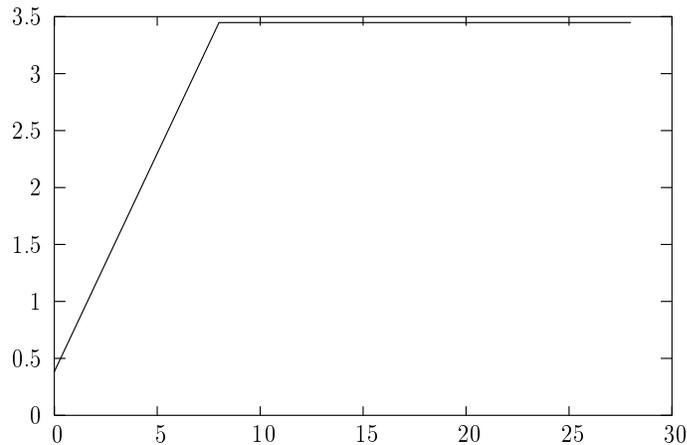
(a)



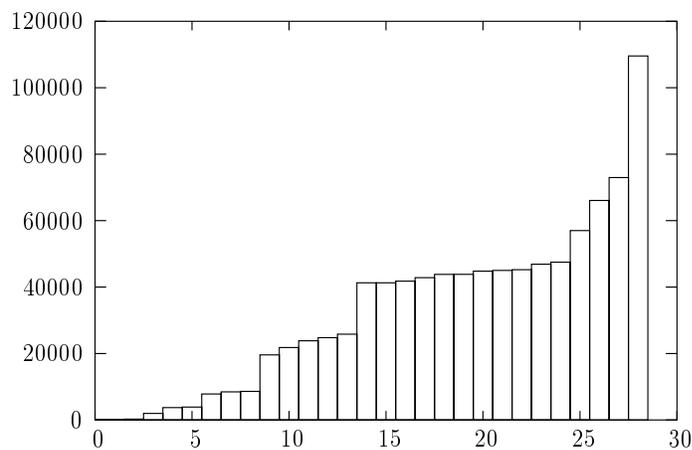
(b)

FIG. 7.16 – Résultats d'une requête du type de la classe Avions

La figure 7.17 montre les résultats obtenus à partir de la requête d'un modèle de la classe Chaise2 et plus exactement celui de la figure 7.13g. Cette classe tout comme la classe Avions est bien définie et visible dans la matrice des rangs (figure 7.15). La figure 7.17a traduit là aussi une pertinence parfaite des résultats. En revanche, on peut voir sur la figure 7.16b que la différence de distance entre les objets de cette classe et les objets les plus proches des autres classes est bien moins net.



(a)

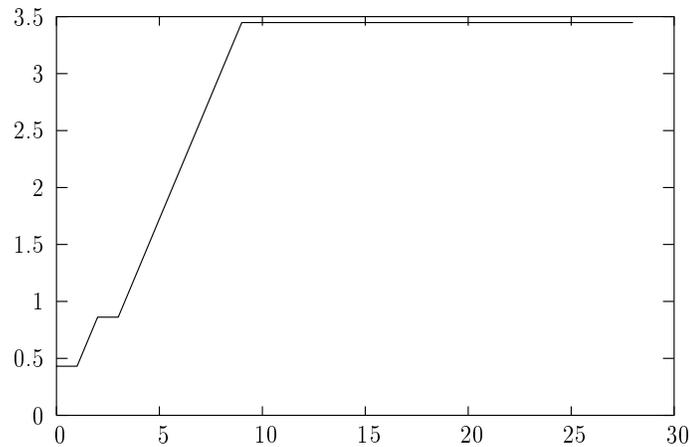


(b)

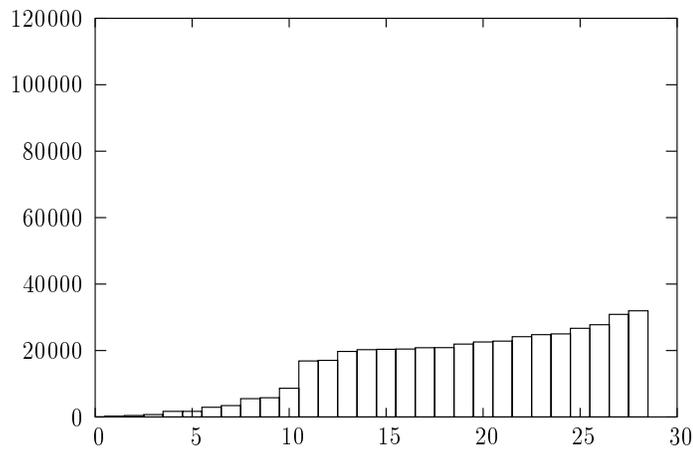
FIG. 7.17 – Résultats d'une requête du type de la classe Chaise2

La figure 7.18 présente les résultats obtenus à partir de la requête d'un modèle de la classe Chaise1 et plus exactement celui de la figure 7.12b. Il s'agit d'un objet bien visible sur la matrice des rangs car ayant des points sombres (ce qui correspond à des objets

trouvés très proches) hors de la zone de comparaison intra-classe. Ces écarts se traduisent par les paliers de la figure 7.18a indiquant deux réponses non-pertinentes. La figure 7.18b montre d'autres faiblesses. Tous les objets semblent relativement proche de ce modèle. On distingue un palier, mais cela est surtout dû au fait que le reste est très uniforme. Ce résultat moyen (au final, on trouve tout de même rapidement les bons objets) peut s'expliquer par le fait que ce descripteur est particulièrement simpliste et son graphe est linéaire, ce qui le rend trop peu discriminant.



(a)



(b)

FIG. 7.18 – Résultats d'une requête du type de la classe Chaise1

Malgré ces quelques égarements, les résultats que nous obtenons sont globalement tout à fait satisfaisants comme l'ont montré la courbe de rappel et la matrice des rangs.

Au niveau du temps de calcul, qui est un paramètre important de ce genre d'application, la durée d'une requête est assez variable d'un objet à un autre. En effet, rappelons qu'un seul arbre couvrant est calculé pour les objets de la base, tandis que toutes les possibilités sont envisagées pour le descripteur de l'objet requête. Ce dernier est donc le facteur de loin le plus important de la vitesse de résolution. Par exemple, une requête sur notre base de test à partir d'un objet de la classe `Chaise1` dont le graphe est linéaire et donc très simple prend 13 secondes tandis qu'à partir de l'avion le plus complexe de la base la résolution prend près d'une minute trente. En moyenne, le temps de résolution d'une requête est d'un peu plus de 20 secondes. Notons tout de même que nous n'avons pas fait d'optimisations poussées et que surtout l'algorithme de distance d'édition est implémenté de manière naïve.

### 7.3 Conclusion

Dans ce chapitre, nous avons défini les opérations de modification de graphe nécessaires pour l'exploitation de la distance d'édition sur notre problème. Nous avons choisi d'utiliser une opération de fusion et une de relabélisation. La détermination des coûts est une étape primordiale de cette définition. Ce fut l'occasion d'approfondir encore notre connaissance des superellipsoïdes en nous intéressant à leur mesure de similarité. Pour celle-ci, nous avons proposé différentes méthodes de calculs à partir d'une approche basée sur la différence de volumes des modèles.

Les résultats que nous obtenons sont encourageants et ne peuvent que nous inciter à évaluer notre méthode de comparaison d'objets sur une base de tests de plus grande envergure.

# Conclusion

Dans cette seconde partie, nous avons proposé une méthode de comparaison d'objets décrits par un ensemble de superellipsoïdes structuré par un graphe d'adjacence. Ce descripteur d'objet peut être le résultat de la méthode que nous avons proposée dans la première partie du mémoire ou bien issu d'une tout autre méthode. De plus, ce descripteur étant particulièrement accessible à la compréhension humaine (à l'inverse d'autres descripteurs que nous avons présentés dans le chapitre 5), il peut avoir été construit manuellement et proposé par un utilisateur en tant que requête à un moteur de recherche. Car en effet, avec cette méthode de comparaison et compte tenu de son caractère particulièrement compact, notre descripteur est maintenant apte à être utilisé dans une application d'indexation.

Étant donné le format de notre descripteur, la comparaison entre ces derniers revient à effectuer une mesure de similarité entre les graphes qui les structurent en tenant compte bien entendu des superellipsoïdes correspondantes aux sommets. Nous avons donc présenté différentes notions associées à la mise en correspondance de graphe. Les notions traditionnelles (famille des isomorphismes) ne se sont pas révélées suffisamment souples pour notre application et ne renseignent souvent que de manière qualitative. Ainsi, nous avons choisi d'utiliser une méthode justement prévue pour quantifier la différence entre deux structures plus ou moins proches : la distance d'édition.

L'étape essentielle de cette méthode est donc le choix des fonctions d'édition ou de modification et la définition de leur coût. Ces fonctions sont de deux types. D'une part, une fonction de comparaison entre primitives est nécessaire. Cette comparaison se traduit par une relabélisation d'un sommet du graphe, et le calcul de son coût revient à une mesure de similarité entre les primitives associées à ce sommet. L'autre opération modifie la structure même du graphe. Nous avons choisi de rester assez proche de notre méthode de segmentation en proposant une opération de fusion de sommets. Celle-ci est tout à fait comparable aux fusions de l'algorithme découpage-fusion. Bien entendu, ce choix n'empêche aucunement d'utiliser cette méthode sur des descripteurs du même format que le nôtre, mais obtenus d'une manière différente.

En ce qui concerne le calcul des coûts, qui est la partie la moins évidente dans notre

cas, nous nous sommes basés pour les deux opérations sur des différences de volume avant et après modification du descripteur. Le fait d'avoir des coûts dans la même unité facilite bien sûr leur cumul lors du calcul total de la distance entre les descripteurs et la comparaison de séquences d'opérations.

Les tests de validation que nous avons présentés à la fin de cette partie sont satisfaisants et attestent de la pertinence de la méthode. Cependant, nous sommes conscients qu'ils nécessitent d'être approfondis pour être complètement significatifs et c'est pourquoi nous avons comme perspective à très court terme l'évaluation de notre méthode sur une base de donnée de taille plus conséquente. Bien entendu, comme la comparaison d'objets arrive au bout de la chaîne de travail que nous avons présentée dans ce manuscrit, ce test grandeur nature nous permettra de valider complètement notre méthode de segmentation et de reconstruction qui constitue l'étape précédente.

Comme toutes les méthodes de mise en correspondance de graphes, un problème de la distance d'édition est qu'elle reste assez coûteuse en temps de calcul. D'autant plus que puisque les sommets de nos graphes ne sont pas ordonnés, nous devons faire autant de calculs de distance d'édition qu'il y a de sommets. Fixer un sommet de départ en choisissant une superellipsoïde permettrait donc de diviser la complexité de la méthode par le nombre de sommets du graphe. On peut par exemple imaginer commencer par la superellipsoïde la plus grosse ou bien celle qui a le plus grand nombre de voisins. Ainsi, dans ce dernier cas, l'origine serait le siège dans le cadre de nos comparaisons de chaises ou bien le fuselage dans le cas des avions. Bien sûr, le grand risque de cette approche est que le choix de l'origine dans le cas de deux objets similaires soit différent, ce qui conduirait à des résultats erronés.

Comme perspective à moyen terme, nous pensons utiliser de manière plus complète les informations que porte le descripteur. En effet, en ce qui concerne l'adjacence des superellipsoïdes, nous en sommes restés à l'analyse la plus simple en la considérant uniquement de manière qualitative à travers l'algorithme de distance d'édition. Cependant, il serait judicieux de prendre aussi en compte les paramètres spatiaux des jonctions comme, par exemple, leur angle ou/et leur disposition. En se plaçant dans le cadre de la distance d'édition, cela peut être géré au niveau de la relabélisation. En plus de la mesure de similarité entre les primitives, il y aurait une mesure de similarité entre les configurations de leurs jonctions. Ce problème peut paraître assez complexe si on reste dans le cas général de liaison entre des primitives 3D, d'autant plus qu'il faudrait comparer toutes les combinaisons de jonctions possibles dans le premier et le second descripteur. Mais tout cela est en fait grandement simplifié grâce à la méthode de calcul de la distance d'édition. En effet, nous n'avons pas à prendre en compte les combinaisons de jonctions car elles l'ont déjà été lors de la génération des différents arbres couvrants de graphes. Les fils de chaque noeud, et donc les jonctions, sont ordonnées dans les arbres que nous utilisons et

donc lors de la relabélisation, ainsi l'ordre des jonctions est exactement le même dans les deux descripteurs s'ils représentent des objets similaires. Le problème propre à la 3D de position relative peut aussi être résolu en prenant par exemple le père du noeud considéré.

Cette prise en compte quantitative des jonctions aurait bien sûr pour conséquence une plus grande capacité de discrimination de la méthode et donc plus de pertinence dans les résultats. Aussi, elle permettrait peut-être une réduction du temps de calcul car cela ferait un critère supplémentaire, au moment de chaque relabélisation, pour arrêter l'exploration d'un arbre couvrant.



# Conclusion générale

Tout au long de ce mémoire, nous avons détaillé une chaîne complète allant de la segmentation de points non organisés jusqu'à l'obtention d'un descripteur 3D compact et son utilisation dans une application d'indexation d'objets 3D.

Dans la première partie, nous nous sommes appliqués à construire un descripteur d'objets à partir de données tridimensionnelles très générales : un ensemble de points 3D non organisé et sans régularité supposée. Notre description se base sur une décomposition de l'objet en primitives volumiques particulièrement compactes : les superellipsoïdes. Ainsi, nous avons proposé une méthode originale de segmentation et reconstruction adaptée à des données tridimensionnelles très générales. Le descripteur qui en résulte n'est pas seulement un ensemble quelconque de superellipsoïdes, mais est structuré par un graphe d'adjacence.

Les résultats que nous obtenons sont satisfaisants d'un point de vue segmentation des données mais la représentation à l'aide de simples superellipsoïdes est assez grossière dans le cas où les différentes parties décrites ne sont pas aussi simples qu'une de nos primitives. Si cela ne permet pas d'utiliser en l'état le descripteur pour de la visualisation moyenne ou haute définition, l'information sémantique induite par le graphe d'adjacence sous-jacent nous permet, en revanche, d'envisager son utilisation dans des applications de type reconnaissance.

Dans la seconde partie, nous avons utilisé notre descripteur afin de comparer des objets tridimensionnels dans avec un objectif d'indexation. Le principal problème que nous avons eu à résoudre fut la mise en correspondance et la quantification des différences entre les structures des descripteurs des deux objets comparés. Pour ce faire, nous avons utilisé une approche par distance d'édition permettant ce calcul à partir de la définition d'opérations basiques de modification des descripteurs. Les résultats de cette méthode de comparaison que nous obtenons dans une expérience d'indexation sont encourageants et montrent que notre descripteur d'objet est apte à être utilisé en l'état pour ce type d'applications. Par exemple, une des applications pourrait être, à l'instar des moteurs de recherche tel que *google* pour les pages de texte, de proposer la possibilité de lancer des requêtes web. Notre travail sur l'indexation autorise déjà la recherche de similarité entre objets 3D sur une base, et la mesure de similarité permet une présentation du résultat du plus pertinent au

moins pertinent. Il pourrait être intéressant de développer la méthode pour faire effectuer des recherches sur des parties d'objets : être capable de dire si tel objet compose un autre. Dans le cadre de la distance d'édition, cela peut être fait en ajoutant une opération de suppression sur l'un des deux objets. Le problème serait sans doute la définition du coût qui devrait logiquement être nul (supprimer une primitive n'est pas synonyme de déformer le descripteur puisque l'on cherche simplement une partie de l'objet) mais qui dans ce cas conduirait à l'élagage radical de l'objet.

## Perspectives

Les résultats de la première partie du mémoire nous ont montré certaines limites de représentation pour une exploitation de notre descripteur dans le domaine de la visualisation. Pour ces applications de visualisation, plusieurs solutions d'amélioration de la description s'offrent à nous. Comme nous l'avons déjà évoqué dans la section consacrée aux superquadriques du premier chapitre de la première partie (cf. paragraphe 1.4), de nombreux travaux ont été faits dans le domaine des déformations appliquées aux superellipsoïdes afin justement d'augmenter leur pouvoir descriptif. Une première solution peut donc être d'ajouter des déformations à chacune des primitives. Les déformations peuvent être de deux types. Il y a des déformations globales comme celles proposées par Barr [Bar84], celles-ci modifient directement les équations des superquadriques en ajoutant des paramètres de torsion, de pliage, etc. Cette première catégorie ne semble pas être une très bonne solution. D'une part, l'équation de la primitive va devenir plus complexe à manipuler, l'approximation des données et leur segmentation devront être faites en tenant compte de ces nouveaux paramètres. D'autre part, on perdra un peu de la simplicité qui rend l'utilisation de superellipsoïdes très intuitive, et les applications de type comparaison d'objets 3D à l'aide du descripteur seront plus compliquées.

Cependant, il existe un autre type de déformations dont font partie les déformations libres (FFD). La déformation est appliquée localement à la surface, l'équation des superellipsoïdes n'est pas modifiée, la déformation n'est qu'une sur-couche. L'utilisation des FFD par Bardinet [BCA95] est très intéressante et montre une augmentation du potentiel descriptif très important. Cette solution semble mieux adaptée. Son défaut est peut-être de rendre le descripteur final un peu plus volumineux qu'en utilisant les déformations globales, mais ses capacités de représentation seraient alors incomparables. De plus, elle permet de préserver des méthodes d'approximation et de segmentation identiques en ne considérant que de simples superellipsoïdes. L'ajout de FFD ne se ferait alors qu'une fois le processus terminé en considérant chaque région et sa superquadrique associée. Ainsi, le descripteur offrira à la fois une information sur la structure de l'objet, son organisation spatiale, très simple à interpréter grâce au graphe de superellipsoïdes, ainsi que beaucoup plus de précision dans la représentation en considérant la "sur-couche" des FFD.

Sans utiliser les déformations, une seconde solution pour des applications de visualisa-

tion est d'augmenter le nombre de primitives par descripteur. En effet, cela augmentera le pouvoir descriptif de l'ensemble, mais le descripteur perdra beaucoup de son information structurelle. On en revient au problème de la conciliation entre segmentation et distorsion. Cette solution n'est donc pas la bonne pour garder une certaine universalité du descripteur. Cependant, on peut garder dans chaque descripteur à la fois une structure basse résolution et une autre haute résolution. Cela peut paraître très discutable. On peut alors en ajouter d'avantage et proposer de garder aussi les étapes intermédiaires de fusion. En effet, plus de primitives par descripteur n'est rien d'autre que le résultat d'un algorithme de segmentation où la fusion s'arrête prématurément. Ainsi, en fixant un critère haute résolution et un autre basse résolution et en stockant les étapes de fusion comprises entre ces deux critères, nous obtiendrons un descripteur multi-résolution. La visualisation se fera suivant l'ordre inverse de la fusion, la multi-résolution aurait même l'avantage d'être locale (l'utilisateur pourrait demander une plus haute résolution que sur une partie du descripteur). Une telle description n'est cependant pas vraiment efficace en étant formulée de la sorte, en effet dans le cas de visualisation progressive, il faut que les éléments d'une résolution donnée soit fonction de la résolution précédente, sinon, comme c'est le cas ici, la quantité d'informations transmise au final est beaucoup plus importante que pour le modèle haute résolution (le descripteur est certes multi-résolution mais non progressif). Cette application n'est donc pas aussi immédiate qu'elle pourrait le laisser paraître.

Bien sûr, on peut encore imaginer une solution mixte utilisant à la fois un nombre plus important de superellipsoïdes et des déformations.

Au final, le travail réalisé et présenté dans ce mémoire a montré que le choix de l'utilisation du modèle des superellipsoïdes pour des applications variées était pertinent. Ces réalisations ouvrent des perspectives très intéressantes qu'il nous tiendra à coeur d'explorer dans des travaux futurs ou à défaut, de suivre leur évolution.



# Publications

Ce travail de thèse a donné lieu aux publications suivantes :

## 1 Conférences internationales avec comité de lecture :

Laurent CHEVALIER, Fabrice JAILLET et Atilla BASKURT 3D : « Shape coding with superquadrics », IEEE ICIP2001, Thessaloniki (G), octobre 2001.

Laurent Chevalier, F. Jaillet et A. Baskurt : « Segmentation and Superquadric Modeling of 3D Objects » , WSCG03, Plzen (CZ), février 2003. vol 11(1).

## 2 Conférences nationales avec comité de lecture :

Laurent CHEVALIER, Fabrice JAILLET and Atilla BASKURT : «Description et reconstruction rapide de surfaces» , CORESA2000, Poitiers (F), octobre 2000.

Laurent CHEVALIER, Fabrice JAILLET and Atilla BASKURT : «Codage de formes 3D par des superquadriques» ,CORESA2001, Dijon (F), novembre 2001.

## 3 Journées nationales avec actes :

Laurent CHEVALIER, Fabrice JAILLET and Atilla BASKURT 3D :« Description et reconstruction rapide de surfaces», COOPANIM 2001, Villeurbanne (F), juin 2001.



# Bibliographie

- [AD01] Pierre Alliez and Mathieu Desbrun. Progressive compression for lossless transmission of triangle meshes. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings, Annual Conference Series*, pages 195–202. ACM Press / ACM SIGGRAPH, 2001.
- [AH80] Firooz A.S. and E.L Hall. Three dimensional moment invariant. *TPAMI*, 2(2) :127–135, 1980.
- [Amr02] Morade Amrani. *Modélisation et simulation d'objets déformable*. PhD thesis, Université Claude Bernard Lyon 1, 2002.
- [Bar81] Alan H. Barr. Superquadrics and angle preserving transformations. *IEEE Computer Graphics and Applications*, 1(1) :11–23, January 1981.
- [Bar84] Alan H. Barr. Global and local deformations of solid primitives. *Computer Graphics*, 18(3), 1984.
- [BCA95] Eric Bardinet, Laurent D. Cohen, and Nicolas Ayache. A parametric deformable model to fit unstructured 3d data. Research report 2617 - INRIA, 1995.
- [Béz87] P. Bézier. *Courbes et surfaces*, volume 4 of *Mathématiques et CAO*. Hermès, 2nde edition, 1987.
- [BG87] T.E. Boult and A.D. Gross. Recovery of superquadrics from 3-D information. *SPIE Intell. Robots and Comp. Vision*, 1987.
- [Bie85] Irving Biederman. Human image understanding : Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 32(1) :29–73, October 1985.
- [Bli82] James F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3) :235–256, July 1982.
- [BS87] Bajcsy and F. Solina. Three dimensional object representation revisited. In *Proc. of the 1st International Conference on Computer Vision*, pages 231–240, London, 1987.
- [CC94] Isaac Cohen and Laurent D. Cohen. A hybrid hyperquadric model for 2-D and 3-D data fitting. *Computer Vision and Image Understanding*, 63(3) :527–541, 1994.

- [CLL97] L.H. Chen, Y.T. Liu, and H.Y. Liao. Similarity measure for superquadrics. *VISP*, 144(4) :237–243, August 1997.
- [CPZ97] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree : An efficient access method for similarity search in metric spaces. In Matthias Jarke, Michael J. Carey, Klaus R. Dittrich, Frederick H. Lochovsky, Pericles Loucopoulos, and Manfred A. Jeusfeld, editors, *Proceedings of the Twenty-third International Conference on Very Large Data Bases, Athens, Greece, 26–29 August 1997*, pages 426–435, Los Altos, CA 94022, USA, 1997. Morgan Kaufmann Publishers.
- [Cru03] D. Santa Cruz. *iCompression of parametric 3D models with NURBS*. PhD thesis, École polytechnique Fédérale de Lausanne, 2003.
- [CS98] Jin-Long Chen and George C. Stockman. 3D free-form object recognition using indexing by contour features. *Computer Vision and Image Understanding : CVIU*, 71(3) :334–355, September 1998.
- [Die93] P. Dierckx. *Curve and surface fitting with splines*. Oxford University Press, 1993.
- [EM97] Paquet E. and Rioux M. Nefertiti : a query by content software for three dimensional models databases management. In *International Conference on Recent Advances in 3D Digital Imaging an Modeling*, 1997.
- [Far93] G. Farin. *Curves and surfaces for CAGD, a practical guide*. Academic Press, 3eme edition, 1993.
- [FPF99] Andrew W. Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5) :476–480, 1999.
- [GA00] Eric Galin and Samir Akkouche. Incremental polygonization of implicit surfaces. *Graphical Models*, 62 :19–39, 2000.
- [GB93] Alok Gupta and Ruzena Bajcsy. Volumetric segmentation of range images of 3d objects using superquadric models. *CVGIP : Image understanding*, 58(3) :302–326, November 1993.
- [Hop96] Hugues Hoppe. Progressive meshes. *Computer Graphics*, 30(Annual Conference Series) :99–108, 1996.
- [JLS00] Ales Jaklic, Ales Leonardis, and Franc Solina. *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer, Dordrecht, 2000. ISBN 0-7923-6601-8.
- [Koe90] Jan J. Koenderink. *Solid Shape*. MIT Press, Cambridge, Massachusetts, 1990.
- [KTSK00] Philip Klein, Srikanta Tirthapura, Daniel Sharvit, and Ben Kimia. A tree-edit-distance algorithm for comparing simple, closed shapes. In *Proceedings ACM-SIAM Symposium on Discrete Algorithms*, pages 696–704, 2000.

- [LC87] William E. Lorensen and Harvey E. Cline. Marching cubes : A high resolution 3D surface construction algorithm. *Computer graphics*, 21(4) :163–168, July 1987.
- [LD89] Chong-Huah Lo and Hon-Son Don. 3-D moment forms : Their construction and application to object identification and positioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(10) :1053–1064, October 1989.
- [LJS97] Ales Leonardis, Ales Jaklic, and Franc Solina. Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11) :1289–1295, November 1997.
- [LSM94] Ales Leonardis, Franc Solina, and Alenka Macerl. A direct recovery of superquadric models in range images using recover-and select paradigm. *Proceedings of 3rd European Conference on Computer Vision*, 1994.
- [MA00] F. Mokhtarian and S. Abassi. Automatique view selection in multi-view object recognition. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, 2000. Barcelona, Spain.
- [MB98] Messmer and Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE TPAMI : IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 1998.
- [Mur91] Shigeru Muraki. Volumetric shape description of range data using "blobby model". In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 227–235. ACM Press, 1991.
- [NHK<sup>+</sup>85] Hitoshi Nishimura, Makoto Hirai, Toshiyuki Kawai, Toru Kawata, Isao Shirakawa, and Koichi Omura. Object modeling by distribution function and a method of image generation. *The Transactions of the Institute of Electronics and Communication Engineers of Japan*, J68-D(4) :718–725, 1985. In Japanese (translated into English by Takao Fujiwara while at Centre for Advanced Studies in Computer Aided Art and Design, Middlesex Polytechnic, England, 1989).
- [NM65] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7 :308–313, 1965.
- [OFCD01] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In Bob Werner, editor, *Proceedings of the International Conference on Shape Modeling and Applications (SMI-01)*, pages 154–166, Los Alamitos, CA, May 7–11 2001. IEEE Computer Society.
- [PD90] A. Pentland and T. Darrell. Part segmentation for object recognition. *AAAI Symp on The Theory and Application of Minimal Length Encoding*, pages 105–109, March 1990.

- [Pen91] Alex Pentland. Closed-form solutions for physically based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(7), 1991.
- [Per92] E. Perna. *Modèle de surface pour la CFAO raccordement de carreaux définis par produit tensoriel*. PhD thesis, Université Claude Bernard Lyon1, 1992.
- [PMY94] J. Park, D. Metaxas, and A. Young. Deformable models with parameter functions : application to heart-wall modeling. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Seattle, WA*, pages 437–442, June 1994.
- [Pon96] Serge Pontier. Contribution à la modélisation et à la reconstruction des objets déformables à l'aide de fonctions spatiales - généralisation des formalismes existants. Master's thesis, Université Claude Bernard, Juin 1996.
- [PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flanery. *Numerical Recipes in C*, chapter 15. Cambridge University Press, 1992.
- [RJ92] N. Raja and A. Jain. Recognizing geons from superquadrics fitted to range data. *Image and Vision Computing*, 12(3) :179–189, April 1992.
- [S02] Vallette Sébastien. *Modèles de maillages déformables 2D et multirésolution surfaciques 3D sur une base d'ondelettes*. PhD thesis, INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON, 2002.
- [SB90] Franc Solina and Ruzena Bajcsy. Recovery of parametric models from range images : the case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2), 1990.
- [SKK01] Thomas Sebastian, Philip Klein, and Benjamin Kimia. Recognition of shapes by editing shock graphs. In *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01)*, pages 755–762, Los Alamitos, CA, July 9–12 2001. IEEE Computer Society.
- [SM02] M. Daoudi S. Mahmoudi. Une nouvelle méthode d'indexation 3d. In *13ème Congrès de reconnaissance de Formes et Intelligence Artificielle (RFIA2002)*, volume 1, pages 19–27, Janvier 2002. Angers.
- [SP86] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. *Computer Graphics*, 20, 1986.
- [Tau91] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11) :1115–1138, 1991.
- [TG98] Costa Touma and Craig Gotsman. Triangle mesh compression. In *Graphics Interface*, pages 26–34, June 1998.

- [TM91] Demetri Terzopoulos and Dimitri Metaxas. Dynamic 3D models with local and global deformations : deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7) :703–714, July 1991.
- [TR98] Gabriel Taubin and Jarek Rossignac. Geometric compression through topological surgery. *ACM Transactions on Graphics*, 17(2) :84–115, 1998.
- [TZ02] F. Prêteux T. Zaharia. Indexation de maillages 3d par descripteurs de forme. In *Actes 13ème Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA '2002)*, Angers, France, pages 48–57, January 2002.
- [TZ04] F. Prêteux T. Zaharia. Descripteurs de forme : Etude comparée des approches 3d et 2d/3d. In *Actes 14ème Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA '2004)*, Toulouse, France, January 2004.
- [VCD02] J.P. Vandeborre, V. Couillet, and M. Daoudi. A practical approach for 3d model indexing by combining local and global invariants. In *3D Data Processing Visualisation and Transmission (DPVT)*, 2002. Pavoda, Italy.
- [WL94] K. Wu and M. D. Levine. Recovering parametric geons from multiview range data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 159–166, Los Alamitos, CA, USA, June 1994. IEEE Computer Society Press.
- [WMW86] Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Soft objects. In Tsiyasu L. Kunii, editor, *Advanced Computer Graphics (Proceedings of Computer Graphics Tokyo '86)*, pages 113–128. Springer-Verlag, 1986.
- [YKY92] Naokazu Yokoya, Masahisa Kaneta, and Kazuhiko Yamamoto. Recovery of superquadric primitives from a range image using simulated annealing. In *Proceedings of International joint Conference on Pattern Recognition*, pages 168–172, 1992.
- [ZK01] Lin Zhou and Chandra Kambhamettu. Extending superquadrics with exponent functions : Modeling and reconstruction. *Graphical models*, 63(1) :1–20, January 2001.
- [ZN94] M. Zerroug and R. Nevatia. Using invariance and quasi-invariance for the segmentation and recovery of curved objects. *Lecture Notes in Computer Science*, 825 :317–??, 1994.