HAL
open science

Indexation multi-vues et recherche d'objets 3D

Thibault Napoléon

To cite this version:

Thibault Napoléon. Indexation multi-vues et recherche d'objets 3D. Traitement des images [eess.IV]. Télécom ParisTech, 2010. Français. NNT : . pastel-00576966

HAL Id: pastel-00576966

https://pastel.hal.science/pastel-00576966

Submitted on 15 Mar 2011

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

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

Thèse

présentée pour obtenir le grade de docteur de Télécom ParisTech

Spécialité : **Signal et images**

Thibault Napoléon

Indexation multi-vues et recherche d'objets 3D

Soutenue le 6 Juillet 2010 devant le jury composé de :

Anne Verroust-Blondet	Rapporteur
Matthieu Cord	
Jean-Luc Dugelay	Examinateur
Hervé Glotin	
Hichem Sahbi	Directeur de thèse
Francis Schmitt	

"Ce n'est pas les uns contre les autres
mais les uns avec les autres
que nous devons avancer."

À Francis,
à ma famille.

“Ce n’est pas les uns contre les autres
mais les uns avec les autres
que nous devons avancer.”

Remerciement

Résumé

Dans cette thèse, nous nous intéressons aux problématiques liées à l'indexation et à la recherche d'objets 3D par le contenu. En effet, suite à l'essor des nouvelles technologies et des réseaux de communication tels qu'Internet, leur nombre n'a cessé de croître amenant, de ce fait, des difficultés pour naviguer dans les bases de données de modèles 3D.

Dans un cadre plus particulier, nous nous sommes penchés sur des méthodes d'indexation multi-vues, caractérisant la forme à l'aide de projections 2D de l'objet en trois-dimensions. Ce type d'approches nécessite de mettre en place trois traitements distincts qui sont : le prétraitement des données en vue de les rendre invariantes aux transformations géométriques, la description pertinente de leur forme et le processus de recherche qui doit permettre de répondre aux requêtes de l'utilisateur.

Dans un premier temps, nous introduisons une nouvelle approche de normalisation et d'alignement des objets 3D nécessaire à notre processus d'indexation. La définition de la position et de l'échelle s'appuie sur la sphère minimale englobante qui offre des propriétés intéressantes pour notre caractérisation multi-vues de la forme. En ce qui concerne la recherche de l'alignement optimal, nous proposons un estimateur permettant de comparer deux poses différentes en s'appuyant sur des résultats obtenus en psychologie cognitive.

Dans une seconde partie, nous définissons trois nouveaux descripteurs de forme basés sur des projections 2D. Le premier décrit la forme d'une silhouette à l'aide d'un ensemble de pixels. Celui-ci permet d'appliquer des opérations ensemblistes sur les signatures, afin de les comparer. Dans un second temps, nous utilisons des informations de convexités et de concavités [1][2][3] pour décrire le contour des projections de nos objets 3D. Ces mesures ont permis de définir deux descripteurs apparentés, basés sur une compression par histogrammes et par DCT. Enfin, nous proposons un dernier descripteur où les projections 2D sont associées des informations d'orientation de la surface. Cette caractérisation que nous appelons "carte de normales" est synthétisée à l'aide de cœfficients de Fourier.

Finalement, le processus de recherche que nous introduisons permet d'interroger la base de données à l'aide d'objets 3D, de photos ou de dessins au trait. Afin de garantir une recherche efficace en temps et en pertinence des résultats, nous proposons deux optimisations. La première s'appuie sur la fusion de résultats à l'aide de différents opérateurs d'agrégation tandis que la seconde élimine rapidement les objets éloignés de la requête au moyen d'un élagage précoce.

Abstract

Table des matières

Notations **v**

1 Introduction **1**

2 Contexte **7**
2.1 Introduction . . . 7
2.2 L'indexation et la recherche appliquées aux objets 3D . . . 7
2.2.1 Qu'est ce que l'indexation d'objets 3D ? . . . 8
2.2.2 L'indexation d'objets basée sur le contenu . . . 9
2.2.3 Le choix des types de requêtes . . . 9
2.2.4 Les points clés de l'indexation et de la recherche d'objets 3D . . . 10
2.2.5 Contraintes et possibilités du processus d'indexation et de recherche 14
2.3 Modélisation et représentation des objets 3D . . . 18
2.3.1 Modélisation et acquisition des modèles 3D . . . 19
2.3.2 Représentation et maillages 3D . . . 21
2.4 Conclusion . . . 24

3 Etat de l'art **25**
3.1 Introduction . . . 25
3.2 Normalisation des objets 3D . . . 25
3.2.1 Normalisation en translation et en échelle . . . 26
3.2.2 Normalisation en rotation . . . 28
3.3 Classification des méthodes d'indexation 3D . . . 36
3.4 Etat de l'art sur les méthodes d'indexation 3D/3D . . . 38
3.4.1 Méthodes globales . . . 38
3.4.2 Méthodes locales . . . 39
3.4.3 Méthodes par partitions de l'espace . . . 40
3.4.4 Méthodes par transformées . . . 43
3.4.5 Méthodes par squelettes . . . 48
3.5 Etat de l'art sur les méthodes d'indexation "2.5D/3D" . . . 51
3.5.1 Méthodes par cartes de courbures . . . 51
3.5.2 Méthodes par coupes . . . 51
3.5.3 Méthodes par images de profondeurs . . . 52
3.6 Etat de l'art sur les méthodes d'indexation 2D/3D . . . 55
3.6.1 Méthodes basées région . . . 56
3.6.2 Méthodes basées contour . . . 58

3.7 Etat de l'art sur les autres méthodes d'indexation 3D 58
3.7.1 Méthodes par fusion de descripteurs 58
3.7.2 Méthodes de recherche partielle 59
3.8 Conclusion 60

4 Normalisation et estimation de l'alignement 61
4.1 Introduction 61
4.2 Centrage et mise à l'échelle 62
4.2.1 Recherche de la normalisation optimale 62
4.2.2 Sphère minimale englobante 66
4.2.3 Définition de la normalisation 66
4.3 Estimation de l'alignement 68
4.3.1 Recherche de l'alignement pertinent 68
4.3.2 Critère de minimisation de l'enveloppe visuelle 69
4.3.3 Critère simplifié 74
4.3.4 Ordonnancement des axes de rotations et réflexions 75
4.4 Conclusion 77

5 Indexation d'objets 3D 81
5.1 Introduction 81
5.2 Extraction multi-vues de silhouettes 82
5.2.1 Choix des vues 82
5.2.2 Principe d'extraction 86
5.2.3 Conclusion 91
5.3 Descripteur par ensemble de pixels (EP) 91
5.3.1 Descripteur de forme 92
5.3.2 Principe du codage synthétique 93
5.4 Descripteurs multi-échelles de convexités/concavités (MCC) 94
5.4.1 Informations de convexités/concavités des contours 94
5.4.2 Approche multi-échelles 101
5.4.3 Codage synthétique de l'information 103
5.5 Descripteur par cartes de normales (CN) 106
5.5.1 Extraction des normales 106
5.5.2 Codage synthétique des cartes de normales 108
5.5.3 Prise en compte de l'ordonnancement des axes 108
5.6 Conclusion 111

6 Recherche d'objets 3D 113
6.1 Introduction 113
6.2 Fonctionnement du processus de recherche 114
6.2.1 La requête de l'utilisateur 114
6.2.2 La comparaison avec la base de données 115
6.2.3 Interface homme-machine 115
6.2.4 Schéma global 116

6.3 Définition, analyse et description des requêtes . . . 117
6.3.1 Requête de type “objet 3D” . . . 117
6.3.2 Requête de type “photos” . . . 118
6.3.3 Requête de type “dessins au trait” . . . 119
6.3.4 Conclusion . . . 120
6.4 Principe de comparaison . . . 121
6.4.1 Mesures de dissimilarité . . . 121
6.4.2 Comparaison de deux descriptions . . . 131
6.4.3 Liste des résultats . . . 133
6.5 Optimisations de la recherche . . . 134
6.5.1 Optimisation multi-descripteurs . . . 135
6.5.2 Optimisation par élagage . . . 138
6.6 Outil et prototype . . . 142
6.6.1 Moteur de recherche basé sur les dessins au trait . . . 143
6.6.2 Moteur d’Indexation et de Recherche d’Objets 3D “Miro3D” . . . 148
6.7 Conclusion . . . 151

7 Expérimentations et résultats 155
7.1 Introduction . . . 155
7.2 Bases de données et estimation de la pertinence . . . 155
7.3 Expérimentations sur la normalisation et l’alignement . . . 156
7.3.1 Normalisation . . . 156
7.3.2 Estimation de l’alignement . . . 159
7.4 Expérimentation sur l’indexation et la recherche d’objets 3D . . . 167
7.4.1 Temps d’extraction et de comparaison . . . 167
7.4.2 Choix du nombre de vues . . . 168
7.4.3 Descripteur par ensemble de pixel (EP) . . . 171
7.4.4 Descripteur par convexités/concavités (MCC-H et MCC) . . . 172
7.4.5 Descripteur par cartes de normales (CN) . . . 176
7.4.6 Optimisation par fusion de descripteurs . . . 178
7.4.7 Optimisation par élagage précoce . . . 179
7.4.8 Comparaison des nos descripteurs . . . 182
7.4.9 Expérimentation sur des requêtes 2D . . . 182
7.5 Etude comparative . . . 186
7.5.1 Comparaison avec Princeton . . . 187
7.5.2 Concours SHREC . . . 189
7.6 Conclusion . . . 190

8 Conclusions et perspectives 191

A Etat de l’art sur les mesures de dissimilarité 197

B Calcul de la sphère minimale englobante 201

C Méthode de double rendu **205**

D Bases de données et estimation de la pertinence **211**

Bibliographie **221**

Publications **233**

Notations

Voici la liste des notations utilisées dans ce manuscrit :

$\mathbb{R}^3$	Espace de dimension $d = 3$
$\overrightarrow{x}$, $\overrightarrow{y}$ et $\overrightarrow{z}$	Axes du repère 3D
x_i, y_i et z_i	Les cordonnées 3D du point i dans le repère 3D
$\mathcal{O}$	Un objet 3D
$\mathcal{M}$	Maillage d'un objet 3D
$\mathcal{F}_i$	$i^{ème}$ facette, ou face, d'un objet 3D
$\mathcal{P}_i$	$i^{ème}$ sommet d'un objet 3D
A_i, B_i et C_i	Sommets de la $i^{ème}$ face triangulée d'un objet 3D
$\mathcal{N}_i$	Normale de la $i^{ème}$ face
$\mathcal{G}$	Centre de gravité d'un objet 3D
$\mathcal{G}_i$	Centre de gravité de la $i^{ème}$ face
$\mathcal{G}_{i,s}$	Centre de gravité surfacique de la $i^{ème}$ face
$\mathcal{S}$	Aire de la surface d'un objet 3D
$\mathcal{S}_i$	Aire de la $i^{ème}$ face
$\mathcal{C}_b$	Le centre de la boite englobante
$\mathcal{R}_b$	Le rayon de la boite englobante
$\mathcal{R}_{max}$	Le rayon maximum
$\mathcal{R}_{moy}$	Le rayon moyen
A^T	Transposée de la matrice A
$\mathbf{T}_{x,y,z}$	Translation dans $\mathbb{R}^3$
$\mathbf{S}_{x,y,z}$	Homothétie dans $\mathbb{R}^3$
$\mathbf{R}_{\rho,\phi,\theta}$	Rotation dans $\mathbb{R}^3$
θ	Normalisation (translation + homothétie)
$\mathcal{E}_\Theta$	Enveloppe visuelle d'un objet 3D
σ	Concaténation de transformations affines
$\mathcal{O}_i$	Un objet située en position i dans la liste de résultats
$\mathcal{O}_r$	L'objet requête lors d'une interrogation de la base
$\mathcal{D}_\mathcal{O}$	Le descripteur associé à l'objet 3D
$C(\mathcal{O})$	La classe de l'objet $\mathcal{O}$
$\|C(\mathcal{O})\|$	Cardinale de la classe de l'objet $\mathcal{O}$ (ou taille)
$NN(\mathcal{O}_r)$	Valeur du *nearest neighbor* pour l'objet requête $\mathcal{O}_r$
$FT(\mathcal{O}_r)$	Valeur du *first tier* pour l'objet requête $\mathcal{O}_r$
$ST(\mathcal{O}_r)$	Valeur du *second tier* pour l'objet requête $\mathcal{O}_r$
$DCG(\mathcal{O}_r)$	Valeur du *discounted cumulative gain* pour l'objet requête $\mathcal{O}_r$
$Md(\mathcal{O}_i, \mathcal{O}_j)$	Mesure de dissimilarité entre une projection des objets $\mathcal{O}_i$ et $\mathcal{O}_j$
$D(\mathcal{O}_i, \mathcal{O}_j)$	Distance entre les objets $\mathcal{O}_i$ et $\mathcal{O}_j$
$\|\|..\|\|_i$	Distance L_i

CHAPITRE 1

Introduction

Contexte

Depuis plusieurs années, grâce à l'essor des nouvelles technologies et des réseaux de communication tels qu'Internet, la création, l'utilisation et la distribution de modèles tri-dimensionnels [1] sont devenues plus faciles. En effet, l'apparition sur le marché de logiciels de modélisation ainsi que le développement de scanners 3D a permis aux entreprises, aux musées et aux particuliers de créer, ou d'acquérir, des objets 3D divers et variés. Ce passage en trois dimensions permet de plonger le client, ou l'utilisateur, dans un univers virtuel proche de la réalité où tout est possible, rendant ainsi son expérience visuelle plus riche. C'est aussi grâce à ces modèles 3D que les entreprises et les musées stockent des informations nouvelles leur permettant une étude approfondie des documents. Finalement, grâce aux nouvelles technologies de partage de données qui se sont développées avec Internet, la distribution de l'ensemble de ces représentations numériques devient de plus en plus facile et permet au grand public de bénéficier de ce contenu interactif.

Cependant, à cause de ces nouvelles technologies, les objets 3D deviennent de plus en plus nombreux et leur stockage massif nécessite la création de bibliothèques spécialisées [2]. C'est pour cette raison que les industries automobiles, les musées ou les studios de jeux constituent, en interne, leurs propres bases qui contiennent, en général, plusieurs milliers d'objets 3D. Grâce à celles-ci, il devient possible d'archiver de manière systématique et pérenne les documents 3D tout en garantissant une mise à jour et une maintenance efficace des bases de données par le biais d'outils spécifiques. L'autre atout majeur de ces bibliothèques, qui intéresse particulièrement les industriels, est l'accès rapide aux documents et à leurs moyens de visualisation.

Ces dernières années, le monde de l'informatique fait face à un changement d'habitudes qui donne un nouvel élan aux objets 3D. En effet, les nouvelles technologies 3D, poussées par les studios de cinéma et de jeux, ont permis au grand public de se familiariser avec des univers en trois dimensions. Le regain d'intérêt envers ces technologies vient de la puissance des images 3D pour représenter les univers virtuels, en opposition aux environnements 2D qui n'offrent qu'une pâle représentation de ceux-ci. Associé à des techniques de visualisation avancées, il est possible de visionner un modèle 3D sous différents angles

1. Nous utiliserons indifféremment les terme *modèles 3D* et *objets 3D* pour désigner des documents en trois dimensions.
2. Nous appellerons aussi ces bibliothèques *bases de données* dans la suite de ce manuscrit.

de vues, avec différents éclairages et dans un monde virtuel riche de détails. Ainsi, grâce à cette nouvelle dimension, les musées vont pouvoir plonger le visiteur dans une expérience nouvelle où il deviendra plus acteur que spectateur. Finalement, c'est pour ces différentes raisons que l'on archive désormais les objets en trois plutôt qu'en deux dimensions favorisant de ce fait, l'essor des technologies associées telles que par exemple l'indexation et la recherche 3D.

Motivations

Pouvoir parcourir de telles bases de données est une tâche fastidieuse dès lors que leur taille devient importante. En effet, archivés tels quels, l'accès aux objets 3D demeure chaotique, car étant en général stockés de manière désordonnée. Afin de répondre efficacement à ce problème, des techniques puissantes de recherche ont été mises en place pour offrir des possibilités d'interrogation multiples. Même si leur fonctionnement reste transparent pour l'utilisateur, il est néanmoins intéressant de se pencher sur la partie immergée de l'iceberg, à savoir : le processus d'indexation. Celui-ci a pour but de décrire, par un moyen quelconque, les informations contenues dans le document et ceci de manière très synthétique. En fonction des caractéristiques choisies, l'utilisateur pourra rechercher un objet 3D en fonction de sa forme, sa couleur, sa date de création ou toute autre information le concernant.

Les moteurs de recherche actuels offrent à l'utilisateur un moyen d'interrogation souvent réduit aux seuls mots-clés pour rechercher un document précis. Dans cette approche, un expert associe au préalable à chaque document un ensemble de mots le décrivant. Cette méthode est très adaptée au document de type texte où l'information est contenue dans un ensemble de phrases, qu'il est généralement possible de synthétiser en quelques mots-clés. En revanche, dès lors que les documents contiennent une information visuelle telle que des photos ou des objets 3D, ces techniques de recherche montrent une certaine faiblesse. Par exemple, la description de la forme d'un modèle 3D est trop fortement soumise à la subjectivité de l'expert qui ne décrira pas nécessairement un objet de la même façon que son collègue.

Pour palier cette difficulté, plusieurs scénarios sont possibles afin d'obtenir rapidement un modèle 3D présent dans une grande bibliothèque. En effet, en basant l'indexation sur l'aspect visuel du document, il est possible de dégager trois types de requêtes plus adaptées à l'utilisateur, à savoir : un objet 3D, un ensemble de photos ou un jeu de dessins au trait. Dans un tel processus d'indexation par le contenu, la forme de chaque document est codée dans une description synthétique [3] facilitant sa comparaison avec la requête de l'utilisateur, permettant finalement de retourner les documents qu'il recherche. Afin de faciliter l'interrogation et la navigation dans des bases de données de grandes tailles, il est indispensable de fournir à l'utilisateur des moyens d'interrogation divers et variés.

3. Nous utiliserons aussi les termes *représentation* et *signature* dans la suite de ce manuscrit.

Objectifs

De nombreuses méthodes d'indexation d'objets 3D ont vu le jour dans les dernières années, poussées par l'apparition massive de ces modèles accessibles en ligne. Celles-ci cherchent à répondre en un temps minimum aux requêtes que l'utilisateur fait sur des bibliothèques contenant plusieurs milliers de documents. Cependant, les approches actuelles restent encore, soit imprécises soit trop coûteuses en temps de calcul et ne permettent donc pas de répondre aux besoins de l'utilisateur. De plus, une grande partie de ces approches n'offre qu'un seul moyen d'interrogation, à savoir : un objet 3D, rendant ainsi difficile la recherche puisque l'utilisateur n'en a pas en sa possession. Il nous a donc semblé nécessaire de proposer de nouveaux outils permettant de surpasser ces difficultés.

L'une des principales difficultés dans l'élaboration de tels outils reste l'extraction automatique d'informations sur la forme, la couleur ou toutes autres propriétés d'un objet 3D. En effet, il faut réussir à capturer de manière synthétique les différentes caractéristiques d'un modèle sans pour autant garder toute l'information qu'il contient. La description ne doit pas dépasser les seuils de "sur-indexation" [4] et de "sous-indexation" [5], qui la rendraient inefficace.

D'autre part, afin de répondre du mieux possible aux requêtes de l'utilisateur, ces méthodes doivent être robustes aux transformations géométriques de l'espace, mais aussi aux modes de représentation des objets 3D ou aux imperfections qui ont pu apparaitre lors de sa création. Le but étant de pouvoir répondre aux demandes de l'utilisateur, même si la requête qu'il fournit ne correspond pas exactement à des objets de la base de données. En permettant la mise en correspondance d'objets proches, mais pas similaires, nous évitons de ne renvoyer aucun résultat.

Un autre point essentiel du processus de recherche dans une bibliothèque indexée, est la comparaison de la requête de l'utilisateur avec l'ensemble des modèles présents. En effet, afin de renvoyer à l'utilisateur les documents pertinents, il est nécessaire d'associer, à chaque objet de la base de données, un score ou une distance par rapport à la requête. Les processus d'indexation et de recherche tentent donc de construire une mesure de dissimilarité qui offre une certaine élasticité permettant de prendre en compte une partie des contraintes abordées précédemment. Pour résoudre les problèmes de rapidité de calcul, des méthodes d'accélération peuvent être mises en place afin d'élaguer très rapidement les modèles trop éloignés de la cible de l'utilisateur.

4. La "sur-indexation" est le fait de décrire trop finement l'objet 3D. Ainsi, la signature est peu synthétique et rend difficile la recherche des informations pertinentes.

5. La "sous-indexation" est le fait de capturer trop peu d'informations sur le modèle 3D. Cela entraine une impossibilité à dégager les propriétés importantes de l'objet.

Applications

La principale application de l'indexation d'objets 3D reste évidemment la recherche et l'exploration de bases de données de documents 3D. En effet, grâce à cet outil puissant, des utilisateurs comme les industriels, les studios de cinéma et les musées peuvent naviguer dans leurs bibliothèques. Associés à des techniques de visualisation avancées, les outils de recherche deviennent indispensables et améliorent l'accès à l'information au sein d'une structure. Finalement, c'est leur simplicité d'installation et d'utilisation qui peut faciliter leurs développements dans un futur proche.

Une autre application de l'indexation d'objets 3D est la réalité augmentée. En effet, associée à une requête de type photo, il peut être possible de retrouver dans une base de données le document correspondant à l'objet réel qu'un utilisateur observe. Dans ce sens, l'observation d'une œuvre d'art pourrait être associée à la prise d'une photo qui, grâce à une recherche dans la base, permettrait la représentation virtuelle de l'objet observé dans son environnement initial, enrichissant de ce fait l'expérience du visiteur. Même si ce type d'outil reste complexe à mettre en place, il fait cependant partie des futures applications possibles de l'indexation de modèles 3D.

Enfin, une application intéressante est la recherche de similarité entre deux objets préalablement indexés, ceci en vue de les regrouper dans une même classe. En effet, certains musées possèdent une gigantesque collection d'objets d'art et cherchent donc, lors de l'acquisition de nouvelles œuvres, à les regrouper avec les objets du même thème. Grâce à l'indexation d'objets 3D cette tâche devient possible, apportant de ce fait une aide considérable pour repérer les ressemblances éventuelles entre différents modèles.

Aperçu des contributions

Cette thèse présente de nouvelles méthodes d'indexation et de recherche d'objets 3D basées sur des approches dites "2D/3D" [6] et "2.5D/3D" [7]. Dans celles-ci, l'extraction des informations caractéristiques d'un modèle ne se fait pas directement sur le modèle 3D, mais sur une représentation en deux dimensions de celui-ci. Ce type d'approche offre en général une meilleure sélection des propriétés intrinsèques d'un modèle et permet d'interroger la base de données directement avec des objets 3D, mais aussi avec des photos ou des dessins faits par un utilisateur. Les nouvelles méthodes présentées cherchent à répondre de manière rapide à une requête tout en assurant une qualité des résultats très satisfaisante.

L'une des contributions de cette thèse est l'élaboration d'une nouvelle approche permettant le prétraitement des objets 3D dans le but de les rendre invariants aux trans-

6. On appel "2D/3D" des méthodes caractérisant la forme d'un objet tridimensionnel (3D) à l'aide d'informations en deux dimensions (2D) obtenues généralement par projection.

7. On appel "2.5D/3D" des méthodes caractérisant la forme d'un objet tridimensionnel (3D) à l'aide d'informations qui se situent entre trois et deux dimensions. Il s'agit généralement d'informations de profondeurs ou d'orientation de surfaces (3D) contenues dans des projections (2D) de l'objet 3D.

formations géométriques de l'espace qui sont : la translation, le changement d'échelle et la rotation. Cette méthode de normalisation se décompose en deux parties utilisant une estimation de l'alignement pour le traitement de la rotation et une sphère englobante pour les deux autres transformations. Cette nouvelle approche a fait l'objet d'une publication dans *EURASIP Journal on Image and Video Processing* [76].

Une fois qu'une certaine robustesse face aux transformations de l'objet a été acquise, nous capturons des silhouettes, du modèle 3D à indexer, afin d'extraire une description en deux dimensions. Sur ces silhouettes, nous utilisons des représentations ensemblistes permettant de décrire la forme des objets, mais aussi des mesures de convexités et de concavités des contours ou encore des informations d'orientations de surfaces à travers les normales.

Une fois le travail de description de l'objet 3D effectué, il faut pouvoir répondre de manière rapide et pertinente à une requête de l'utilisateur. Pour traiter ces spécifications, nous proposons une fusion de descripteurs ainsi qu'un processus d'élagage précoce. Cela permet, d'une part, d'améliorer les résultats et, d'autre part, de réduire l'espace de recherche afin de faire une analyse plus fine des ressemblances avec les objets pertinents de la base de données. Le descripteur basé sur les informations de convexités et de concavités ainsi que l'accélération du processus par élagage ont été publiés dans [75].

La dernière contribution importante de notre travail de thèse est la possibilité de rechercher un objet 3D présent dans la base de données au moyen de photos ou de dessins au trait fait par l'utilisateur. En effet, dans de nombreux cas, il est difficile d'obtenir une représentation en trois dimensions d'un modèle que l'on souhaite rechercher. Il est donc indispensable, dans ce cas, d'offrir à l'utilisateur des moyens d'interrogations variés basés sur des dessins par exemple. C'est grâce à l'extraction d'informations de forme en deux dimensions que cette amélioration nous a été possible. Finalement, afin de prouver qu'il est possible de concrétiser cette idée, nous avons développé un prototype fonctionnant sur un terminal mobile. L'ensemble de ces travaux traitant des requêtes de types "photos" ou "dessins au trait" ont fait l'objet d'une publication [77].

Organisation du manuscrit

Cette thèse s'organise comme suit :

- Introduction (1).
- Chapitre 2 : Contexte.
- Chapitre 3 : Etat de l'art.
- Chapitre 4 : Normalisation et estimation de l'alignement.
- Chapitre 5 : Indexation d'objets 3D.
- Chapitre 6 : Recherche d'objets 3D.
- Chapitre 7 : Expérimentations et résultats.
- Conclusions et perspectives (8).

Dans le chapitre 2, nous introduisons le contexte de nos recherches sur l'indexation 3D. Il débute avec la définition de "processus d'indexation et de recherche d'objets 3D". Puis enchaîne avec l'acquisition des modèles tridimensionnels et la représentation de ceux-ci.

Nous étudions dans cette thèse l'indexation et la recherche appliquée aux objets 3D. Nous proposons donc dans le chapitre 3 un état de l'art détaillé traitant des méthodes de normalisations des modèles 3D. Nous revenons aussi sur les méthodes d'indexation dites "3D/3D", "2.5D/3D" et "2D/3D" afin de bien comprendre la position de notre travail dans la littérature existante.

Dans le chapitre 4, nous étudions les deux parties qui constituent notre nouvelle méthode de normalisation et d'estimation de l'alignement d'un objet 3D. Une présentation détaillée de cette approche est faite afin de bien comprendre l'intérêt de ces deux étapes.

Le chapitre 5 présente la sélection et l'extraction des informations nécessaires à la description synthétique d'un objet 3D. Nous y proposons trois nouveaux descripteurs multi-vues "2D/3D" et "2.5D/3D" codifiant les caractéristiques de forme intrinsèques à un modèle permettant sa comparaison efficace lors d'une requête de l'utilisateur.

La mise en correspondance des informations stockées dans la signature, permettant à l'utilisateur d'interroger la base de données, est discutée dans le chapitre 6. Nous y expliquons le processus de recherche d'un document à partir d'un objet 3D, d'un ensemble de photos ou d'un jeu de dessins au trait. C'est aussi dans ce chapitre que nous introduisons le prototype et l'application développés au cours de cette thèse.

Finalement, dans le chapitre 7, nous optimisons, étudions et validons nos choix à travers les résultats de la chaîne de traitement complète. Pour finir, nous effectuons une comparaison de notre approche, par rapport aux méthodes existantes, afin de comprendre les points forts, mais aussi les faiblesses de nos travaux.

CHAPITRE 2

Contexte

Résumé *: Ce chapitre dresse le contexte de cette thèse. Nous y introduisons les différentes phases de l'indexation basée sur le contenu en insistant particulièrement sur les types de requêtes, proposés à l'utilisateur, et sur les contraintes qui régissent le processus. Pour finir, nous proposons un bref aperçu des méthodes de modélisation et de représentation d'objets 3D.*

2.1 Introduction

Ce chapitre a pour but d'introduire le contexte de cette thèse, à savoir : l'indexation d'objets tridimensionnels. Il est en effet intéressant de se plonger au cœur de cet univers, régi par une multitude de contraintes offrant des possibilités diverses et variées quant à l'élaboration et l'utilisation d'un tel processus. Le but premier de nos travaux est la satisfaction de l'utilisateur qui passe par une pertinence des résultats et une bonne ergonomie. Ces deux notions sont le plus souvent liées et doivent faire l'objet d'une attention particulière. Celle-ci passe par la définition de contraintes d'utilisations permettant le traitement des données dans un souci d'accessibilité. C'est dans cette optique que nous allons présenter les principes fondamentaux de l'indexation de documents, et en particulier des objets 3D, ainsi que les critères qui ont guidé nos choix dans la suite de ce manuscrit.

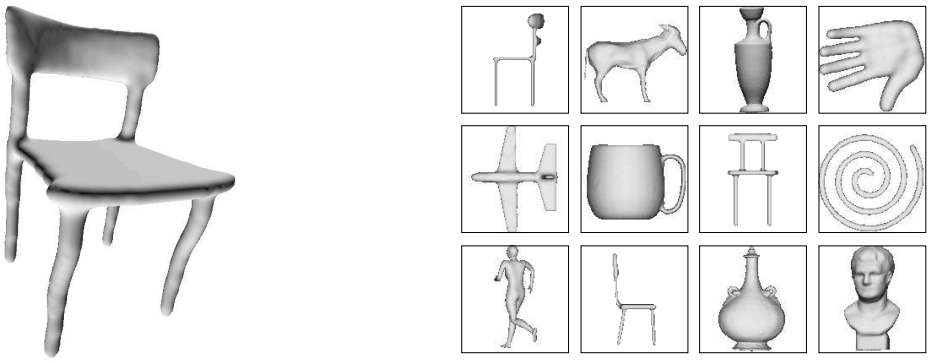
Nous débutons ce chapitre par l'introduction des principaux aspects de l'indexation et de la recherche d'objets 3D à travers la section 2.2. Nous nous attardons en particulier sur les types de requêtes, les points clés mais aussi sur les contraintes de ce processus. Nous poursuivons, dans la section 2.3, la présentation du contexte avec les modes d'acquisitions et de représentations des modèles 3D, indispensables pour nos traitements futurs.

2.2 L'indexation et la recherche appliquées aux objets 3D

Le processus d'indexation et de recherche que nous souhaitons construire a pour objectif l'interrogation d'une base de données d'objets 3D. De ce fait, il est nécessaire, avant toute chose, de définir les types de requêtes qui sont offerts à l'utilisateur. Ceux-ci doivent garantir une utilisation ergonomique en s'appuyant sur des moyens d'interrogations variés. Afin de rendre le processus efficace en temps et en pertinence des résultats, il est nécessaire de bien comprendre les points clés d'un tel système basé sur un ensemble de contraintes. Celles-ci vont de la machine à l'utilisateur en passant par les données qui sont, pour nous, les objets 3D.

2.2.1 Qu'est ce que l'indexation d'objets 3D ?

L'indexation est un procédé qui permet de codifier le contenu d'un document dans le but d'accélérer l'interrogation de grandes bases de données (voir figure 2.1). Le document peut être de différents types tels que par exemple : un texte, une photo, un son ou, dans notre cas, un objet en trois dimensions. Cette opération d'indexation consiste à analyser les différentes propriétés du modèle 3D afin d'en extraire les principales caractéristiques. Cette description est une représentation synthétique, mais pertinente du contenu. La principale tâche consiste donc à sélectionner les données importantes, présentes dans l'objet, et de les codifier dans une description adaptée à l'information extraite. Finalement, il est possible de comparer rapidement deux objets en analysant leur description.



(a) *Requête* (b) *Base de données*

FIGURE 2.1 – *Le but du processus d'indexation et de recherche d'objets 3D est de savoir quels modèles de la base de données (b) ressemble à la requête (a) en terme de forme.*

Cette étape d'indexation peut être effectuée par un humain qui va comprendre et analyser le document, le décrire dans un langage naturel et enfin transposer cette description une représentation synthétique. Mais cette tâche peut aussi, dans certains cas, être automatisée par un ordinateur qui va effectuer ces mêmes opérations sous la supervision d'un expert ou non. Cependant, cette tâche est complexe est nécessite de construire des outils "intelligents" qui analysent le contenu de chaque modèle. En effet, si l'homme est capable de comparer très rapidement deux objets, il n'en est pas de même pour un processus automatique et particulièrement dans les cas où les objets sont ressemblants, mais pas identiques (voir remarque 2.1).

> **Remarque 2.1.** *Face à la reconnaissance de formes, un être humain est incapable de préciser le processus lui permettant d'effectuer cette tâche. Néanmoins, des travaux en psychologie cognitive menés par Marr [69] ont permis de mettre en évidence certains niveaux de représentations de ce processus complexe. Niveau 1 : La reconnaissance des contours de l'objet. Niveau 2 : La prise en compte de l'information de profondeurs. Niveau 3 : La représentation dans un univers 3D indépendant de l'orientation.*

L'ampleur des bases de données, constituées par les industriels ou les musées, oblige souvent ceux-ci à se tourner vers des solutions logicielles. En effet, l'utilisation d'outil d'indexation automatique permet de réduire le travail fastidieux de la description des documents. Cependant, avant même de chercher à caractériser un objet 3D, il faut se poser la question du type d'informations que l'on souhaite extraire.

2.2.2 L'indexation d'objets basée sur le contenu

Généralement, la description d'un document est basée sur une représentation par mots-clés. Celle-ci a été choisie car elle est adaptée aux documents textes qui ont bénéficié les premiers d'une indexation. C'est d'ailleurs ce type de requête que l'on retrouve, sur Internet, dans les moteurs de recherches classiques tels que : *Google*[1] pour les pages Web, *Google Images*[2] pour les images, *Google Vidéos*[3] pour les vidéos, ou encore *Google 3D Warehouse*[4] pour les objets 3D.

Cependant, ce type de représentation n'est pas sans inconvénient. En effet, il est souvent difficile de décrire le contenu de médias complexes tels que des images, des objets 3D ou des vidéos avec un ou plusieurs mots-clés. Cela nécessite de coder les caractéristiques de forme, de couleur ou de texture avec des mots rendant, de ce fait, la description très subjective. Il est donc clair que ce type de requête, même s'il reste proche de l'utilisateur, est insuffisant pour des objets 3D. C'est pour cette raison que, depuis une quinzaine d'années, les méthodes d'indexations basées sur le contenu se multiplient (voir chapitre 3).

En indexation d'objets 3D, les documents à traiter sont souvent des modèles contenant peu d'informations texturelles ou de couleurs. Il est donc nécessaire de décrire et de coder leur forme. Dans ce cadre, il existe deux grandes catégories permettant d'extraire cette information visuelle. La première capture la forme directement à partir du modèle en trois dimensions, ce sont les approches dites "3D/3D" (voir chapitre 3 section 3.4). La seconde passe par une projection de l'objet dans un espace en deux dimensions avant de le décrire, ce sont les approches dites "2.5D/3D" et "2D/3D" (voir sections 3.5 et 3.6 du même chapitre).

2.2.3 Le choix des types de requêtes

La construction d'un tel processus d'indexation et de recherche peut s'effectuer de deux manières différentes. La première s'appuie sur la description synthétique de l'objet 3D pour définir le moteur de recherche qui va permettre l'interrogation de la base de données. Inversement, la seconde cherche avant toute chose à définir le type des requêtes qui vont être

1. http://www.google.fr/
2. http://images.google.fr/
3. http://video.google.fr/
4. http://www.google.fr/3dwarehouse/

formulées par l'utilisateur, avant de réfléchir aux méthodes permettant d'analyser leurs caractéristiques. Même si la première approche reste la plus utilisée dans le monde de la recherche, la seconde nous a paru plus pertinente car tournée vers l'utilisateur final.

En effet, dans certains cas, l'utilisateur qui souhaite rechercher un objet 3D n'a pas forcément une copie de celui-ci. Ainsi, il peut être intéressant de lui offrir plusieurs scénarios pour interroger la bibliothèque, à travers des requêtes 2D par exemple. Cette nouvelle formulation, proche de la pensée humaine (voir Lemaire [67]) permet de décrire l'aspect visuel d'un objet tout en offrant une interrogation simple et proche de l'utilisateur. La figure 2.2 présente un exemple des trois types de requêtes proposés, à savoir :

- **Des dessins de contours au trait** : Il s'agit d'un moyen très simple pour l'utilisateur de rechercher un objet 3D. En effet, ce mode d'interrogation lui permet de transcrire l'image mentale de sa recherche dans une représentation adaptée au système. Afin d'affiner sa recherche, il lui est possible de fournir éventuellement des dessins représentant différents angles de vues.

- **Des photos** : Dans le même esprit que les dessins de contours, une requête par photos est envisageable. En effet, il est possible de détourer rapidement l'objet que l'on souhaite rechercher sur une photo (voir remarque 2.2) et ainsi d'interroger la base de données de modèles 3D, éventuellement à partir de plusieurs photos. C'est grâce à ce type de requête que l'utilisateur pourrait, par exemple, retrouver l'objet 3D correspondant à une œuvre d'art face à lui.

- **Un objet 3D** : Finalement, bien qu'étant le moyen d'interrogation le moins adapté à l'utilisateur, ce type de requête offre une grande robustesse grâce à l'information qu'elle véhicule. En effet, la requête étant un objet 3D, la description choisie sera identique à celle des modèles de la base. Il est intéressant de noter que c'est l'unique moyen d'interrogation pour les méthodes dites "3D/3D".

> **Remarque 2.2.** *La séparation de l'arrière-plan et du premier plan dans une photo est un sujet de recherche actif possédant des méthodes performantes. La comparaison effectuée par McGuinness et O'Connor [70] met en évidence quatre méthodes de segmentations adaptées à ce problème. Pour chacune d'entre elles, l'utilisateur donne un ou plusieurs échantillons du fond et de l'objet afin de guider les algorithmes. En quelques clics de souris il est ainsi possible de séparer l'objet voulu du reste de l'image.*

2.2.4 Les points clés de l'indexation et de la recherche d'objets 3D

Du fait que le processus d'indexation et de recherche doive permettre une interrogation efficace d'une base de données, il est nécessaire qu'il réponde à un enchaînement précis d'opérations. On le décompose souvent en deux phases : indexation puis recherche, ainsi

(a) *Dessin au trait* (b) *Photo* (c) *Objet 3D*

FIGURE 2.2 – *Cette figure montre un exemple des trois types de requêtes permettant d'interroger une base de données d'objet 3D. Il s'agit d'un contour aux traits (a), d'une photo d'un objet réelle (b) et d'un modèle 3D (c). Même si cette figure ne montre qu'une requête unique pour (a) et (b), il est possible de fournir plusieurs dessins ou photos.*

qu'en deux temps : "hors-ligne" et "en-ligne". Il faut bien noter que les deux phases d'un tel système, même si elles reposent sur des aspects différents, ne sont pas totalement dissociables. On peut les décrire comme suit :

- **Phase d'indexation** : Celle-ci permet d'analyser les objets afin de décrire leur forme et de la transcrire dans une représentation adaptée à la machine. La figure 2.3 présente cette phase d'indexation qui se décompose en trois étapes, permettant l'élaboration d'une description pertinente de la forme. La première est une étape de prétraitement où les objets sont normalisés et positionnés de telle sorte que deux objets d'une même classe soient décrits de la même manière (voir chapitre 4). C'est aussi dans cette partie que l'on applique les propriétés d'affichage permettant le rendu d'attributs spécifique de l'objet 3D (couleur, éclairage...). La seconde, est l'étape principale, qui va extraire les informations de forme contenues dans les objets 3D afin de les décrire de manière pertinente (voir chapitre 5). Finalement, la dernière étape permet de transcrire la description de forme dans une représentation synthétique permettant un stockage efficace (voir également chapitre 5). C'est cette représentation qui fait office d'élément de base pour la comparaison de deux objets.

- **Phase de recherche** : Celle-ci est la seule partie du processus visible pour l'utilisateur. En effet, cette étape est effectuée à chaque fois qu'il interroge la base de données. C'est lors de celle-ci que la comparaison de la description de la forme de sa requête, avec les descriptions des objets 3D stockées durant la phase précédente, a lieu. C'est finalement grâce à une mesure pertinente que le système va être capable de renvoyer des résultats cohérents. Ceux-ci vont être présentés à l'utilisateur sous forme d'une liste d'objets, ou de classes, en fonction de l'application choisie (voir remarque 2.3 et figure 2.4).

Remarque 2.3. *Lors d'une recherche de l'utilisateur, il est nécessaire de fournir la liste des résultats avec un minimum de clarté. Dans ce sens, nous proposons deux approches différentes permettant de lister les objets proches de la requête. La première, appelée "approche par ϵ" propose de renvoyer à l'utilisateur les objets de la base qui ont une distance inférieure à ϵ (critère de distance : $d < \epsilon$). La seconde, nommée "approche par k plus proches voisins" permet de ne renvoyer que les k objets les plus proches de la requête de l'utilisateur (critère de nombre).*

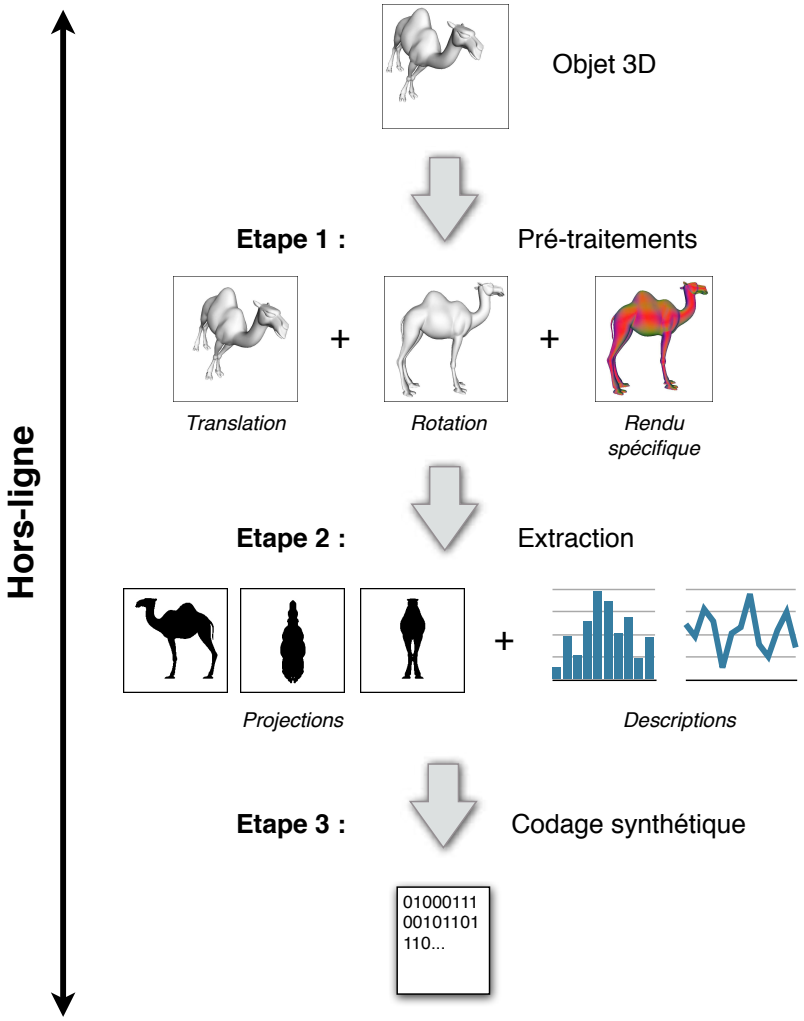


FIGURE 2.3 – *Schéma illustrant les différentes étapes de l'indexation d'objet 3D. Partant d'un modèle 3D, une première étape permet de normaliser en translation et en rotation l'objet et de préparer son rendu. Une seconde extrait une description de sa forme, à partir de projections 2D par exemple. Enfin, une dernière étape transcrit la description dans un langage synthétique adapté à la machine.*

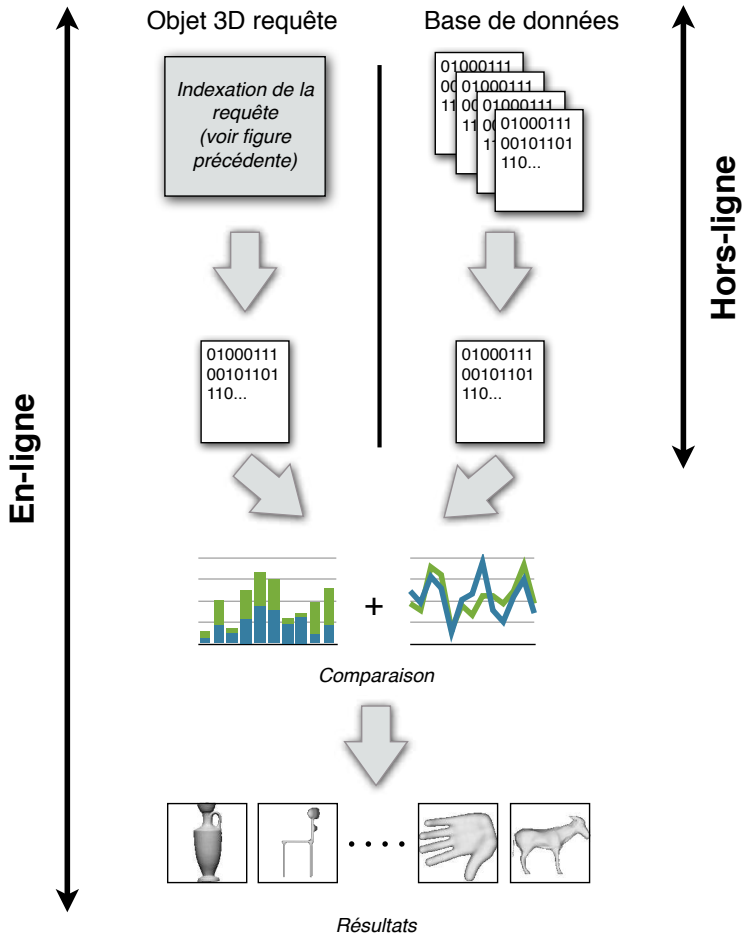


FIGURE 2.4 – *Schéma illustrant les différentes étapes de la recherche d'objets 3D. À partir d'un modèle d'une bibliothèque préalablement indexée et d'une requête de l'utilisateur aussi indexée, une comparaison des descriptions de forme est faite afin de donner une distance entre les deux modèles 3D. Pour donner une liste de résultats à l'utilisateur, la comparaison de la requête est faite avec chaque objet de la base de données.*

En plus du découpage en deux phases, défini précédemment, il est possible de regrouper les étapes d'indexation et de recherche en deux temps. En effet, certains traitements peuvent être effectués au préalable, "hors-ligne", permettant ainsi un traitement plus rapide des requêtes utilisateur, ou alors directement lors d'une demande, "en-ligne" :

– **Traitement hors-ligne** : Il est composé de toutes les étapes qu'il est nécessaire d'effectuer pour initialiser ou ajouter un objet à la base de données. Il s'agit des étapes de normalisation, de description de la forme, de codage dans la signature synthétique et du stockage. C'est, le processus d'indexation a proprement parlé, celui effectué sur chaque modèle 3D. La figure 2.3 présente l'ensemble des étapes de ce traitement.

– **Traitement en-ligne** : Celui-ci est, quant à lui, défini par les opérations à effectuer à chaque fois que l'utilisateur interroge la base. Il s'agit, du processus de recherche qui comprend, la description de la requête, son codage dans la signature synthétique et la comparaison de celle-ci avec l'ensemble des objets 3D de la base. Il n'est pas possible de le rendre "hors-ligne" car cela nécessiterait de connaître toutes les requêtes possibles, ce qui n'est pas envisageable dans le cas d'une recherche par rapport à la forme. En revanche, dans une recherche par mots-clés, il serait plus aisé de rendre cette étape, en partie, "hors-ligne". Les étapes de ce traitement sont présentées dans la figure 2.4.

2.2.5 Contraintes et possibilités du processus d'indexation et de recherche

Nous avons vu les points clés d'un système d'indexation et de recherche d'objets 3D, et ceci, en relation avec les contraintes définies par l'accessibilité envers les utilisateurs. Cependant, afin d'offrir une certaine efficacité, que ce soit en terme de pertinence ou de rapidité, il est nécessaire de définir des contraintes orientant les performances de cette chaîne de traitements. En effet, même si l'utilisateur a des moyens efficaces pour interroger la base de modèles 3D, il est indispensable que les résultats qui lui sont proposés répondent à un certain nombre de critères. L'utilisateur peut, par exemple, souhaiter un système robuste aux articulations d'objets 3D, rapide ou encore extrêmement pertinent. Cependant, certaines combinaisons de ces critères sont impossibles à réaliser à l'heure actuelle à cause de la puissance de calcul dont nous disposons. C'est donc dans ce sens que les contraintes du processus complet doivent être définies afin de répondre du mieux possible au besoin final de l'utilisateur.

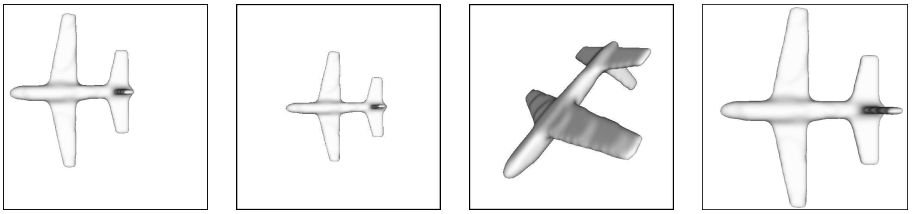
Afin de répondre efficacement à ses besoins, il est souhaitable de définir, au préalable, les contraintes de celui-ci, tout en gardant à l'esprit qu'une certaine flexibilité doit parfois être mise en place pour le satisfaire. En effet, il est par exemple difficile de construire un système très pertinent, tout en garantissant un temps de réponse faible. Dans ce cas, il faut trouver le juste milieu permettant d'offrir des résultats raisonnables dans un temps qui l'est tout autant. Une autre possibilité, plus souple, est d'offrir à l'utilisateur un outil lui permettant de régler ce compromis rapidité/pertinence.

Pour bien comprendre l'ensemble des contraintes possibles, nous dressons une liste non exhaustive de celles-ci, classées selon qu'elles viennent des objets, de la machine qui traite les différentes opérations ou de l'utilisateur.

Contraintes sur les objets

– *Invariances aux transformations de l'espace* : Il s'agit de rendre la description de l'objet invariante aux transformations simples de l'espace, qui sont : la translation, le changement d'échelle et la rotation. Pour résoudre ce problème, il est fréquent qu'une étape de prétraitements soit appliquée avant la description d'un objet 3D

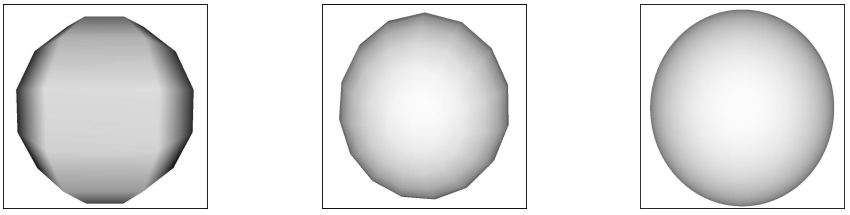
afin de le centrer, le normaliser en taille et de l'orienter d'une manière précise. Ainsi, la distance entre un objet 3D $\mathcal{O}$ et un objet $\sigma(\mathcal{O})$, où σ est la composition de transformations quelconque, est négligeable (voir figure 2.5).



(a) *Translation* (b) *Echelle* (c) *Rotation* (d) *Normalisé*

FIGURE 2.5 – *Exemple d'un même objet ayant subit des transformations affines. (a), (b) et (c) représentent respectivement l'objet 3D ayant subi, une translation, une rotation et un changement d'échelle, tandis que (d) représente l'objet normalisé.*

– *Invariances à la résolution* : Lors de sa création par modélisation ou par acquisition (voir section 2.3), la surface d'un objet 3D peut être définie avec une résolution particulière. De ce fait, il est nécessaire de caractériser la forme en faisant, si possible, abstraction de cette résolution. Cependant, celle-ci est très souvent liée à la qualité visuelle du modèle 3D car jouant un rôle dans la définition de la surface. Ainsi, pour traiter cette invariance, il peut-être utile de travailler avec des approches dites "2D/3D" (voir section 3.6) car ces artefacts n'apparaissent alors que sur les contours des silhouettes. Pour les méthodes "3D/3D" (voir section 3.4) et "2.5D/3D" (voir section 3.5) il est indispensable de construire une description de la forme invariante à la résolution (voir figure 2.6).



(a) *98 faces* (b) *450 faces* (c) *2000 faces*

FIGURE 2.6 – *Exemple d'un même objet (une sphère) avec trois résolutions différentes. (a), (b) et (c) représentent respectivement l'objet 3D en basse, moyenne et haute résolution, soit 98, 450 et 2000 faces.*

– *Invariances aux petites variations de la surface* : Afin de satisfaire l'utilisateur, il est souhaitable de retrouver des objets proches de sa requête sans pour autant qu'ils y soient identiques. Il peut donc être intéressant que le processus d'indexation soit invariant aux petites variations de la surface telles que par exemple le bruit ou les

trous dans le maillage (voir figure 2.7). Cette invariance s'obtient à travers l'utilisation d'une mesure de dissimilarité adaptée ainsi que par l'utilisation d'approches "2D/3D" et "2.5D/3D" permettant de faire abstraction des erreurs de maillage (voir paragraphe 2.2.2).

(a) *Original* (b) *Bruit sur la surface* (c) *Déplacements de quelques sommets*

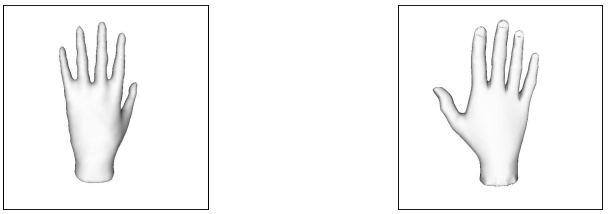
FIGURE 2.7 – *Exemple d'un même objet avec différentes petites variations de surface. (b) et (c) représentent respectivement l'objet (a) avec, du bruit et un déplacement de quelques sommets.*

– *Invariances aux articulations* : Certains objets 3D, sont dits articulés quand ils comportent des jointures flexibles entre deux zones, exemples : la jambe d'un humain, la tête d'un animal ou le manche d'une pince. Ainsi, dans un esprit proche de l'invariance précédente, il est souhaitable, pour certaines méthodes, de prendre en compte ces articulations. Ainsi, il devient possible de considérer des objets articulés comme étant proches, voire similaires. Cette contrainte peut être facilement traitée par des descriptions de la forme à base de graphes (voir section 3.4.5 du chapitre 3) ou dans la mesure de dissimilarité, qui est alors fortement élastique (voir figure 2.8).

FIGURE 2.8 – *Exemple d'un même objet 3D articulé.*

– *Invariance aux miroirs* : Finalement, il peut être intéressant de considérer les objets miroirs. Ceux-ci sont des modèles 3D identiques, à une réflexion près, par exemple : une main gauche et une main droite. Pour résoudre cette invariance, il est nécessaire de mettre en place une description qui s'affranchisse de la réflexion ou de traiter celle-ci lors de la mesure de dissimilarité grâce à une distance adaptée (voir figure 2.9).

Pour conclure, notons que les descripteurs que nous proposerons dans la suite de cette



(a) *Main gauche* (b) *Main droite*

FIGURE 2.9 – *Exemple d'une main gauche (a) et d'une main droite (b).*

thèse chercherons à traiter, autant que possible, chacune de ces contraintes. En particulier, nous définirons, pour l'une de nos signatures, une mesure de dissimilarité "élastique" permettant de prendre en compte les articulations d'objets.

Contraintes sur la machine

- *Puissance de calcul* : Un processus complet d'indexation et de recherche d'objets 3D peut être amené à traiter des milliers, voir des millions d'objets. Dans ce cadre, il faut être capable, dans une première phase, d'indexer un objet dans un temps raisonnable de l'ordre de quelques secondes car se faisant "hors-ligne" sauf pour la requête (voir paragraphe 2.2.4). En revanche, dans la seconde phase, celui de la recherche, il faut être capable de répondre à la demande de l'utilisateur dans un temps bref, si possible inférieur à une seconde. En effet, il est impératif que cette tâche soit rapide afin d'offrir un service efficace aux utilisateurs. Pour résoudre cette contrainte, il est préférable de structurer la base de données, à l'aide d'un arbre par exemple, afin d'accélérer la comparaison. Pour résumer, la puissance de calcul nécessaire au système d'indexation doit être minimum pour la phase "en-ligne" (voir paragraphe 2.2.4) alors qu'elle peut être plus importante pour la phase "hors-ligne".

- *Capacité de stockage* : L'indexation d'objets 3D cherche à décrire de manière efficace, mais synthétique, la forme d'un modèle. Dans ce cadre, il faut être capable d'extraire les informations pertinentes, et uniquement celles-ci, afin de limiter l'espace de stockage. Nous nous rendons cependant compte que la pertinence est souvent liée au nombre d'informations (même s'il peut être fastidieux de les retrouver) et donc à leur taille en mémoire. Il est donc nécessaire de trouver le juste milieu entre pertinence des informations et espace de stockage. Cette contrainte devient très importante dans le cas ou la base de données comporte des millions d'objets, multipliant ainsi la place nécessaire pour stocker les signatures. Une solution efficace, mais coûteuse en temps, est l'ajout d'une phase de compression des données.

Contraintes de l'utilisateur

- *Pertinence des résultats* : Avant toutes choses, ce que souhaite l'utilisateur du système est un ensemble d'objets ressemblant à sa requête. Nous avons vu dans le

paragraphe 2.2.4 que pour satisfaire cette demande, une description de la forme de chaque objet est extraite, et ceci, afin de la comparer à la requête. Ainsi, la pertinence des résultats est fortement liée aux informations choisies pour décrire chaque objet, mais aussi à la mesure permettant de les comparer. Dans ce cadre, il est important de capturer des informations qui caractérisent la forme intrinsèque de l'objet pour ainsi renforcer la recherche d'objets similaires. De plus, il est indispensable de fournir une mesure de dissimilarité adaptée (invariance aux petites déformations, aux articulations...) pour offrir des résultats robustes. Finalement, la fusion de plusieurs descriptions, ou signatures, peut permettre d'améliorer la pertinence globale du processus.

– *Temps de réponse* : Une autre attente de l'utilisateur est la rapidité de la recherche. Il est clair que, dans la plupart des cas, la réponse à une interrogation de la base de données doit se faire en moins d'une seconde, ceci afin d'offrir une bonne réactivité à l'utilisateur. Cependant, ce temps de réponse est souvent très fortement lié à la pertinence des résultats. Afin de résoudre cette contrainte, il peut être nécessaire de mettre en place une stratégie d'élagage permettant un accès rapide aux objets 3D intéressants. Pour répondre aux attentes de l'utilisateur, il est raisonnable d'orienter l'aspect vitesse ou pertinence lui offrant, de ce fait, un système de recherche qui lui est adapté. Finalement, il peut être judicieux de proposer une recherche en "temps contraint" où le temps d'interrogation est fixé par l'utilisateur.

– *Types de requêtes* : Les moyens d'interroger la base de données d'objets 3D sont une contrainte importante (voir paragraphes 2.2.2 et 2.2.3). En effet, l'accessibilité du système va être définie par l'ensemble des moyens permettant de rechercher un modèle 3D. D'une manière générale, l'utilisateur veut un moyen simple de retrouver rapidement l'objet qu'il s'est représenté mentalement. De ce fait, il faut lui permettre de transcrire sa pensée en une requête exploitable par le processus de recherche. Finalement, c'est en multipliant les moyens d'interrogations que l'on facilite l'utilisation du moteur de recherche d'objets 3D.

2.3 Modélisation et représentation des objets 3D

Dans cette thèse, nous nous intéressons à l'indexation appliquée aux objets en trois dimensions. Leur acquisition passe par deux étapes importantes qui sont : la modélisation et la représentation. Un intérêt tout particulier doit être apporté à ces phases car elles conditionnent la qualité du modèle 3D ainsi que ses propriétés qui peuvent influencer la pertinence de l'indexation et donc de la recherche.

Afin de bien comprendre l'influence des données sur le processus de description complet, nous présentons dans cette partie les différents types de modélisation allant de la modélisation manuelle à la reconstruction par photos en passant par les scanners 3D. Nous introduisons aussi les différentes méthodes de représentation des objets 3D. Finale-

ment, pour chacun d'entre eux, nous dresserons un bilan des avantages et inconvénients de chaque méthode.

2.3.1 Modélisation et acquisition des modèles 3D

Les modèles 3D disponibles sur Internet ou dans les bibliothèques spécialisées représentent généralement des objets réels, mais peuvent aussi venir de l'imaginaire de leurs créateurs. Les méthodes de modélisation ou d'acquisition se séparent en deux groupes selon qu'elles permettent la représentation numérique exacte d'un objet réel ou qu'elles offrent une plus grande souplesse ouvrant la voie à la création artistique.

Il existe trois grandes familles de méthodes permettant la création d'un modèle 3D informatique. Qu'elles soient logicielles ou mécaniques, elles offrent toutes des solutions pour donner des informations précises sur la forme et éventuellement sur l'apparence (couleurs, textures...) de l'objet à créer. La première est une approche se basant sur des logiciels de modélisation tandis que les deux autres utilisent des lasers ou des capteurs photographiques.

Modélisation manuelle (infographie)

C'est le moyen le plus courant pour créer un objet 3D. À partir d'un logiciel de modélisation 3D tel que *Maya*[5], *3D Studio Max*[6] ou encore *Softimage*[7] pour les payants ou *Blender*[8] et *Pov-Ray*[9] pour les gratuits, il est possible de façonner des scènes complexes et donc des modèles 3D.

Le principe général de ces modeleurs 3D se base sur la manipulation de primitives simples. Celles-ci se composent généralement de cubes, de sphères, de cônes, mais aussi de courbes de Bézier [10] ou de "NURBS" [11] permettant une plus grande liberté de modélisation. Finalement, l'infographiste ajoute, supprime, modifie ces primitives afin de créer un modèle 3D. Ces logiciels permettent aussi de définir des attributs propres à l'objet tels que : la couleur, la texture, le type de surface (pierre, bois, métal...) ainsi que des points de vue et des lumières permettant de créer animations et rendus réalistes. Ces outils sont très largement utilisés dans la Conception Assistée par Ordinateur [12] qui permet de créer des pièces manufacturées de haute précision.

5. http://www.autodesk.fr/maya
6. http://www.autodesk.fr/3dsmax
7. http://www.autodesk.fr/softimage
8. http://www.blender.org/
9. http://www.povray.org/
10. Ce terme désigne des courbes polynomiales paramétriques.
11. Ce terme est l'abréviation de *Non-Uniform Rational Basis Splines*.
12. Nous utiliserons l'abréviation *CAO* pour désigné cette technique dans la suite du manuscrit.

Scanner 3D

Une méthode efficace pour numériser un objet réel en trois dimensions est l'utilisation d'un scanner 3D. Celui-ci permet de recueillir de manière extrêmement précise la forme d'un objet, mais aussi, dans certains cas, ses attributs de surface tels que sa couleur, sa texture... Ces outils sont souvent utilisés par les industriels ou les musées à des fins d'archivage.

Les scanners 3D transforment la surface d'un objet réel en un nuage de points cohérent permettant une visualisation dans un univers en trois dimensions. Afin de bien comprendre ce mode d'acquisition, il est intéressant de faire une analogie avec la photographie. En effet, comme un appareil photo, un scanner capture des informations sur la scène qu'il observe. Là où le capteur photographique détermine la couleur de la scène, le scanner en détermine la position. L'une des différences majeure entre ces deux outils est le caractère passif du premier – analyse passive de l'image perçue grâce à un capteur – et actif du second – analyse active au moyen d'une source de lumière contrôlée (laser) –. Finalement, il est intéressant de noter que pour numériser l'intégralité de l'objet il est nécessaire de le placer sur une table tournante et d'effectuer plusieurs acquisitions.

Reconstruction 3D par photos

Les techniques de stéréovision permettent aussi de numériser en 3D un objet réel. Dans ces approches, la reconstruction est obtenue à partir de photographies de celui-ci prises dans différents angles de vue. En étudiant les corrélations entre ces photos, il est possible de déterminer la position, mais aussi la couleur de chaque pixel et ainsi reconstruire l'objet 3D. Ces approches sont généralement utilisées dans les musées pour numériser des œuvres de grande valeur, impossible à acquérir par laser.

Il existe trois familles de méthodes permettant la reconstruction de modèles 3D. La première appelée approche par silhouette [13] et décrite par Sullivan et Ponce [114] détermine l'enveloppe visuelle initiale du modèle 3D à partir d'un ensemble de silhouettes. Cette méthode est robuste, mais reste limitée à des formes simples. La seconde variété d'approches, introduite par Horn [50], se base sur l'analyse de l'ombrage d'un objet [14] donnée par les propriétés de réflexion diffuse des surfaces. Elle est difficile à mettre en place car extrêmement dépendante des conditions d'éclairage du modèle. Finalement, une troisième classe d'approche traitée par Seitz et Dyer [108] utilise l'information couleur contenue dans les photographies de l'objet. La combinaison de ces méthodes permet d'améliorer la robustesse comme le montrent les travaux de Hernandez [47] sur la fusion d'informations de silhouettes et de textures.

13. Dans la littérature ces approches sont communément appelées *Shape From Silhouettes*.
14. Dans la littérature ces approches sont communément appelées *Shape From Shading*.

Avantages et inconvénients de chaque méthode

La force des modeleurs 3D vient de la liberté de création qu'ils offrent ainsi que de leur faible coût. Ils nécessitent cependant d'avoir une bonne connaissance des arts graphiques afin d'obtenir un rendu réaliste. Inversement, les scanners 3D permettent de numériser des objets réels de manière rapide, précise (dixième de micromètre) et en haute résolution (plusieurs millions de points). Ils restent néanmoins onéreux et inappropriés pour les modèles possédant de nombreuses parties cachées, par exemple. Finalement, les approches par reconstruction 3D permettent de numériser un objet de manière efficace au moyen d'algorithmes précis, et ceci, de manière passive. Celles-ci demeurent peu employées car le temps de reconstruction peut atteindre plusieurs heures par objet et cela avec seulement une trentaine de photos.

En conclusion, les modeleurs 3D offrent une grande liberté à moindre coût tandis que les scanners 3D numérisent rapidement des objets réels. Les approches par reconstruction ont, quant à elles, le grand intérêt d'être passives, intéressant de ce fait les musées.

2.3.2 Représentation et maillages 3D

Suite à la numérisation d'un objet 3D le problème de la représentation de sa surface se pose (voir section 2.3.1). En effet, si nous prenons l'exemple des scanners 3D, ils ne fournissent pas un objet solide, mais un nuage de points. Il est donc nécessaire de relier ces points afin de pouvoir analyser la surface du modèle ou pour l'afficher dans un univers en trois dimensions.

Il existe principalement trois types de représentations. Polygonales : elle permet de définir la surface d'un objet sous forme de polygone. Volumiques : un objet est représenté par la juxtaposition de "pixels 3D", ou voxels. Par construction de solide : elle utilise des primitives simples et des opérations booléennes pour construire des objets 3D complexes. Notons que les descripteurs que nous proposons dans la suite de cette thèse sont adaptés aux trois représentations. Cependant, nous ne travaillerons qu'avec des représentations polygonales.

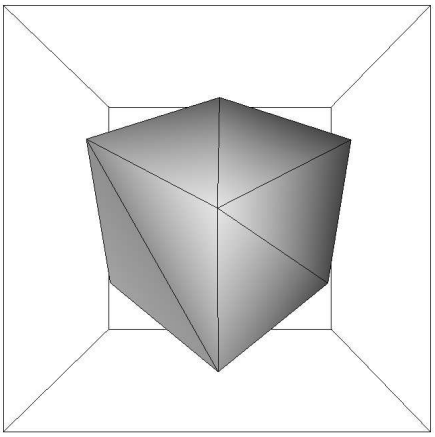
Représentation polygonale

Cette méthode utilise la juxtaposition de polygones simples – souvent des triangles, d'où le terme "objet triangulé" – pour représenter la surface d'un objet, tel que sur la figure 2.10. Elle est largement utilisée pour des raisons de simplicité, mais aussi car elle permet une visualisation rapide des objets aux moyens de technologies dédiées telles qu'*OpenGL* [15] ou *Direct X* [16].

Pour représenter un objet de ce type, il suffit de discrétiser sa surface en une multitude de facettes jointes, composées chacune de sommets reliés par des arrêtes. C'est l'enve-

15. http://www.opengl.org/
16. http://www.microsoft.com/windows/directx/

loppe de l'objet 3D et non son volume qui est donc représentée. L'intérêt majeur de cette méthode est la liberté de modification qu'elle procure. En effet, il est facile de déplacer, souder ou diviser des primitives. L'ajout d'une texture est intuitif en attribuant une couleur à chaque sommet ou une image à une face. Le stockage de cette représentation passe par l'énumération des sommets et des faces de l'enveloppe comme dans les formats ".off", ".obj", ".3ds" ou ".stl" par exemple.



```
# Entete
OFF
# Sommets Faces
8 12 0
-1.0 1.0 1.0
1.0 1.0 1.0
1.0 -1.0 1.0
-1.0 -1.0 1.0
-1.0 1.0 -1.0
1.0 1.0 -1.0
1.0 -1.0 -1.0
-1.0 -1.0 -1.0
```

```
3 0 1 2
3 0 2 3
3 1 5 6
3 1 6 2
3 4 0 3
3 4 3 7
3 3 2 6
3 3 6 7
3 4 5 1
3 4 1 0
3 5 4 7
3 5 7 6
```

FIGURE 2.10 – *Représentation polygonale d'un cube (à gauche) avec sa version informatique en format ".off" listant ses 8 sommets et 12 faces (à droite).*

Pour un objet "triangulé", il est possible de formaliser cette représentation de la manière suivante :

Considérons un objet 3D noté $\mathcal{O}$ constitué d'un ensemble $\mathcal{S}$ de m sommets $\mathcal{S}_i$:

$$\mathcal{S} = \{\mathcal{S}_i | 1 \leq i \leq m\} \tag{2.1}$$

L'objet $\mathcal{O}$ peut alors d'écrire sous la forme :

$$\mathcal{O} = \bigcup_{i=1}^{n} \mathcal{S}_i \tag{2.2}$$

Le maillage $\mathcal{M}$ de $\mathcal{O}$ est composé d'un ensemble $\mathcal{F}$ de n faces triangulaires $\mathcal{F}_i$:

$$\mathcal{F} = \{\mathcal{F}_i | 1 \leq i \leq n\} \tag{2.3}$$

où chaque face $\mathcal{F}_i$ est composée de trois sommets appartenant à l'objet 3D :

$$\mathcal{F}_i = (A_i,\ B_i,\ C_i)\ avec\ 1 \leq i \leq n,\ A_i,\ B_i,\ C_i \in \mathcal{S} \tag{2.4}$$

Le maillage $\mathcal{M}$ peut alors s'écrire sous la forme :

$$\mathcal{M} = \bigcup_{i=1}^{n} \mathcal{F}_i \tag{2.5}$$

Représentation volumique

Cette approche se base sur la notion de voxel[17], qui est l'équivalent, pour l'objet 3D, du pixel pour l'image 2D. Grâce à la juxtaposition de ces briques, il est possible de constituer un objet complexe. En effet, il suffit à partir d'une numérisation, par scanner 3D par exemple, de placer un voxel à la position et distance données pour chaque point sondé. L'utilisation de voxels permet, en particulier, de représenter des modèles 3D avec une résolution fixée à l'avance.

La notion de voxel étant particulièrement appropriée aux scanners, on retrouve cette approche en imagerie médicale. En effet, les "Scanners" et les "IRM "[18] permettent d'obtenir des images 2D de coupes où celles-ci représentent une troisième dimension. De fait, il est assez facile de reconstruire un ensemble de pixels en trois dimensions, donc des voxels. L'un des points faibles de ce type de rendu est l'apparition d'*aliasing*[19], ou crénelage, entre chaque voxel, introduit par l'aspect discret de la représentation.

Représentation par construction de solides "CSG"

Par opposition aux représentations introduites précédemment, celle par construction de solides n'est plus approchée mais exact, car basée sur des opérations mathématiques. Pour cela, elle utilise un ensemble d'opérateurs booléens permettant de combiner des primitives simples. Cette méthode est souvent utilisée par les industriels en "CAO" et par certains rendus par lancé de rayons (voir remarque 2.4).

> **Remarque 2.4.** *Le lancé de rayon est une technique qui permet de représenter une scène 3D, non plus en affichant dans un espace 3D les objets qui la composent, mais en calculant ce que "voit" chaque pixel de l'image. En effet, pour chaque point qui compose le rendu final, un rayon est lancé pour voir quels objets il rencontre et ainsi définir la couleur qu'il recevra. Cette couleur dépendra des ombres, réflexions et réfractions de la lumière induites par les objets et leur matériau.*

De sa nature mathématique, la représentation par construction de solides décrit les frontières des volumes complexes de manière exacte. Les primitives de base sont les cubes, les sphères, les cylindres, les cônes... et les opérateurs sont l'union, la différence et l'intersection. Le positionnement de l'objet se fait à l'aide de transformations géométriques telles que : la translation, la rotation ou l'homothétie. Dans cette représentation, l'objet final peut être décrit sous forme arborescente, où la succession des étapes qui ont permis de construire le modèle 3D sont enregistrées. La racine est l'objet final, chaque nœud correspond à une opération booléenne, ou une transformation géométrique, et chaque feuille correspond à un des volumes de base.

17. On retrouve souvent ce terme dans la littérature qui est en fait l'abréviation de *VOlumetric piXELS*.
18. Il s'agit de l'abréviation du terme *Imagerie à Résonance Magnétique*.
19. Il s'agit d'un effet visuel indésirable provoqué par l'insuffisance de la définition de la surface et qui prend habituellement la forme de dentelures et de contours brisés.

Avantages et inconvénients de chaque représentation

Les représentations polygonales bénéficient d'une grande liberté d'édition et d'une très grande popularité leur permettant de prendre place dans les jeux vidéo, le cinéma ainsi que dans la plupart des logiciels 3D. Cependant, elles ne permettent pas de représenter précisément certaines surfaces, celles des sphères par exemple, et nécessitent un espace de stockage important pour les objets comportant des milliers de faces. Les représentations par voxels ont les mêmes inconvénients auxquels se rajoute la difficulté d'édition. En revanche, ces approches permettent la représentation simple des coupes 2D, largement utilisée en imagerie médicale. Finalement, les constructions de solides offrent un espace de stockage réduit, une qualité de rendu exceptionnel, mais sont coûteux en calcul et possèdent un nombre limité de primitives de bases rendant difficile la représentation d'objets complexes.

Pour conclure, la représentation polygonale reste la plus utilisée car elle s'appuie sur le développement conjoint de techniques d'accélération matérielles et d'interfaces de programmation puissantes offrant, de ce fait, une accessibilité supérieure aux autres méthodes.

2.4 Conclusion

Nous avons présenté dans ce chapitre le contexte de nos travaux sur l'indexation et la recherche d'objets 3D. Il s'est étendu sur le fonctionnement d'un processus complet d'indexation, des moyens d'acquisitions et de représentations, des bases de tests et finalement sur les mesures de pertinences associées. Nous avons clairement vu, qu'avant toutes choses, les critères voulus pour la méthode doivent être clairement définis afin de garantir un fonctionnement satisfaisant pour l'utilisateur. C'est finalement grâce au choix d'approches, dites "2.5D/3D" et "2D/3D", que nous allons pouvoir offrir de nombreux moyens de requêtes, mais aussi une extraction efficace et effective, des informations de forme contenues dans un objet 3D.

CHAPITRE 3

Etat de l'art

Résumé *: Nous proposons dans ce chapitre un état de l'art non exhaustif sur les méthodes de normalisations d'objets 3D ainsi que sur les approches cherchant à caractériser leur forme. Nous nous attarderons en particulier sur les trois types de descripteurs à savoir : "3D/3D", "2.5D/3D" et "2D/3D".*

3.1 Introduction

Dans cette thèse, nous nous intéressons à l'indexation et à la recherche d'objets 3D dans son ensemble. En effet, nous étudions à la fois les techniques de prétraitement nécessaire à rendre invariant le modèle 3D par rapport aux transformations de l'espace, mais aussi à la recherche d'une description pertinente de celui-ci, afin de pouvoir le comparer rapidement à la requête d'un utilisateur. Notre intérêt se porte sur une description de la forme des modèles 3D, et non sur d'autres caractéristiques, telles que la couleur par exemple. C'est grâce à cette analyse qu'une comparaison efficace, au moyen d'une mesure de dissimilarité adaptée, permettra de retrouver les objets stockés dans la base qui sont proches de la requête fournie par l'utilisateur. Dans cette thèse, nous avons focalisé nos travaux sur trois axes qui sont : la normalisation, l'indexation 3D et la recherche d'objet. Nous présentons donc dans cette partie un état de l'art sur les méthodes d'indexation et de recherche d'objets 3D qui existent dans la littérature, ainsi que sur les prétraitements effectués en amont de ces techniques.

Ce chapitre ayant pour but d'établir un état de l'art sur l'ensemble des méthodes permettant l'indexation et la recherche d'objet 3D, nous présentons dans la section 3.2 les approches permettant la normalisation des objets 3D par rapport aux translations, changements d'échelles et rotations. Ensuite, après avoir présenté notre classification des méthodes existantes dans la section 3.3, nous établirons dans les sections 3.4, 3.5 et 3.6 un état de l'art sur les méthodes d'indexations 3D/3D, 2.5D/3D et 2D/3D. Nous dégagerons pour chaque technique les principaux avantages et inconvénients avant de conclure.

3.2 Normalisation des objets 3D

La normalisation est une étape de prétraitement, parfois nécessaire, à la description de la forme d'un objet 3D. En effet, pour rendre cette signature invariante aux transformations de l'espace, telles que la translation, le changement d'échelle ou la rotation, de nombreuses méthodes définissent un centre, une échelle et une position pour chaque objet.

Ainsi, il est possible de décrire par la suite deux objets similaires de la même manière, sans se soucier de leur représentation dans l'espace $\mathbb{R}^3$. Il faut cependant noter que cette étape n'est nécessaire que si la description de forme extraite n'est pas elle-même invariante aux transformations affines. En général, cette normalisation s'effectue en deux temps, le premier où le centrage et la mise à l'échelle sont effectués et la seconde où une orientation canonique est recherchée pour chaque classe d'objets. Dans la section 3.2.1, nous introduirons les méthodes permettant de régler le problème de la translation et du changement d'échelle, tandis que dans la section 3.2.2 nous reviendrons sur les méthodes permettant de fixer l'orientation des modèles 3D.

3.2.1 Normalisation en translation et en échelle

La normalisation en translation et en échelle permet de définir, pour chaque objet 3D, un centre et une échelle. Ces prétraitements consistent à déplacer le centre de l'objet au centre du repère 3D, d'une part, où la difficulté réside dans le choix du centre de l'objet 3D, et à définir une échelle locale à l'objet d'autre part. Nous dressons ici, une liste représentative, des méthodes de la littérature permettant de traiter ces deux notions.

Le centrage

- **Le centre de gravité** : C'est le moyen le plus simple de définir le centre pour un objet non maillé (voir chapitre 2) c'est-à-dire dont les sommets ne sont pas reliés entre eux. Pour son calcul, le centre de gravité, noté $\mathcal{G}$, est calculé sur l'objet $\mathcal{O}$ discrétisé en m sommets notés $\mathcal{P}_i$:

$$\mathcal{G} = \frac{1}{m}\sum_{i=1}^{m} P_i \tag{3.1}$$

  Même si ce calcul est très peu coûteux en temps et simple à mettre en place, il est peu robuste aux changements de surface de l'objet 3D. En effet, ce centre ne tient pas compte de la surface de celui-ci,déf mais seulement de ses sommets. De ce fait, si les sommets du modèle ne sont pas uniformément répartis sur sa surface, le centre devient peu cohérent. Cette définition n'est donc à utiliser que dans le cas où aucun maillage n'est disponible et que les sommets sont uniformément répartis sur la surface. Ce type d'objet peut être obtenu au moyen d'un scanner 3D ou par une reconstruction à l'aide de photos (voir section 2.3.1).

- **Le centre de gravité surfacique** : Aussi appelé centre d'inertie, il correspond au barycentre des sommets qui composent le maillage de l'objet 3D (voir section 2.3.2). C'est grâce à l'utilisation de ce maillage, noté $\mathcal{M}$, que cette définition se rapproche d'un calcul sur une surface continue de l'objet 3D. En effet, dans cette formulation, chaque facette qui contribue au calcul du centre, est pondérée par son aire. Posons $\mathcal{S}_i$ l'aire d'une facette $\mathcal{F}_i$ composée de trois sommets A_i, B_i et C_i :

$$\mathcal{S}_i = \frac{1}{2}\|\overrightarrow{A_iC_i} \wedge \overrightarrow{A_iB_i}\| \tag{3.2}$$

Alors le centre de gravité surfacique $\mathcal{G}_{s,i}$ du maillage défini par les m facettes $\mathcal{F}_i$ vaut :

$$\mathcal{G}_{s,i} = \frac{1}{\mathcal{S}} \sum_{i=1}^{m} \mathcal{S}_i \mathcal{G}_i = \frac{1}{\mathcal{S}} (\mathcal{S}_1 \mathcal{G}_1 + ... + \mathcal{S}_m \mathcal{G}_m) \tag{3.3}$$

où $\mathcal{S}$ représente l'aire de l'objet 3D et avec :

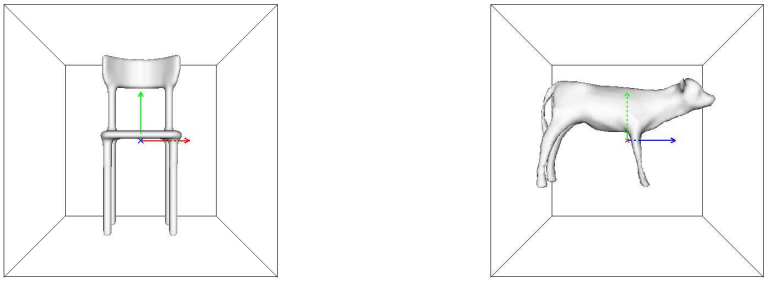
$$\mathcal{G}_i = \frac{1}{3}(A_i + B_i + C_i) \tag{3.4}$$

Ce centre de gravité surfacique est utile et robuste lorsqu'un maillage est disponible. Il est de plus assez peu coûteux car il demande uniquement de calculer l'aire des facettes.

– **Le centre de la boite englobante** : La boite englobante est définie comme étant la plus petite boite englobant l'intégralité de l'objet 3D. Son centre $\mathcal{C}_b$ est défini part :

$$\mathcal{C}_b = \left(\frac{x_{min} + x_{max}}{2}, \frac{y_{min} + y_{max}}{2}, \frac{z_{min} + z_{max}}{2} \right) \tag{3.5}$$

où x_{min}, x_{max}, y_{min}, y_{max}, z_{min} et z_{max} sont les coordonnées minimales et maximales de l'objet 3D. Cette définition de centre est identique à celle appelée "EBB", pour *Extended Bounding Box*, par Vranic [134]. Cette notion d'objet englobant, ici une boite, est très utile lors des approches multi-vues où plusieurs projections de l'objet 3D sont nécessaires. En effet, dans ce cas, en plaçant les plans de projections sur la surface de cet objet englobant, il est facile de projeter l'objet dans son intégralité car son enveloppe est alors connue (voir figure 3.1).



(a) *Une chaise* (b) *Un chien*

FIGURE 3.1 – *Exemple de deux objets centrés dans leur cube englobant. (a) représente une chaise tandis que (b) représente un chien. Ce cube n'étant pas nécessairement le cube minimum englobant, cette enveloppe n'assure pas la maximisation de l'aire des projections (voir chapitre 4.)*

– **Le centre de symétrie** : Il a été défini par Podolak *et al.* [97] et se calcul après définition des plans de symétrie miroir de l'objet. Il correspond à l'intersection des trois

plans orthogonaux de symétrie miroir maximale. Même si ce centre est robuste pour les objets comportant des symétries, il reste ambigu lorsque celui-ci n'en comporte pas.

La mise à l'échelle

- **Le rayon maximum** : En considérant l'un des centres $\mathcal{C}$ introduits ci-dessus, le rayon maximum est défini comme la distance maximale entre le centre et chacun des m sommets $\mathcal{P}_i$ de l'objet 3D :

$$\mathcal{R}_{max} = \max_i \|\mathcal{C} - \mathcal{P}_i\|_2 \tag{3.6}$$

  Cette échelle est équivalente à celle nommée "CBC", pour *Canonical Bounding Cube*, par Vranic [134] lorsque l'on choisit le centre de gravité surfacique pour $\mathcal{C}$. Il est utile de noter que cette échelle permet de faire entrer l'objet 3D dans une sphère de rayon $\mathcal{R}_{max}$. Il est ainsi possible d'obtenir des projections 2D où l'objet est visible dans son intégralité.

- **Le rayon moyen** : En considérant l'un des centres $\mathcal{C}$ introduits ci-dessus, le rayon moyen est défini comme la distance moyenne entre le centre et chacun des m sommets $\mathcal{P}_i$ de l'objet 3D :

$$\mathcal{R}_{moy} = \frac{1}{m} \sum_{i=1}^{m} \|\mathcal{C} - \mathcal{P}_i\|_2 \tag{3.7}$$

  Vranic [134] définit "CC2", pour *Canonical Cube 2*, comme étant $4 * \mathcal{R}_{moy}$ avec le centre de gravité. Même si dans cette définition, l'objet ne rentre pas dans l'enveloppe définie par la sphère de rayon $\mathcal{R}_{moy}$, elle permet de supprimer les points trop éloignés du centre, éventuellement dû à des erreurs lors de l'acquisition du modèle 3D.

- **Le rayon de la boite englobante** : Il s'agit du rayon de la boite englobante défini ci-dessus et il est identique à celui de "l'EBB" défini par Vranic [134]. Dans le cas où l'on considère la boite englobante, le rayon vaut :

$$\mathcal{R}_b = \|\mathcal{C}_b - \mathcal{P}_{max}\|_2 \tag{3.8}$$

  où $\mathcal{P}_{max}$ représente le sommet de l'objet 3D le plus distant du centre de la boite englobante. Ce rayon est souvent utilisé, en association avec le centre de la même boite englobante, dans les approches "2.5D/3D" et "2D/3D", car elle permet de définir les plans de projections comme étant les faces du cube.

3.2.2 Normalisation en rotation

La normalisation en rotation, ou alignement de l'objet 3D permet de définir la pose canonique d'un objet 3D. C'est un prétraitement indispensable pour un grand nombre

d'approches "2.5D/3D" et "2D/3D", qui leur permet de s'affranchir des rotations éventuelles de l'objet. Pour résoudre ce problème, il faut définir pour chaque objet les trois axes qui définissent le repère 3D associé au modèle. Il est possible ensuite, si nécessaire de définir le signe, ou sens, de ces axes afin de régler les problèmes de réflexions. C'est dans ce cadre que nous présentons les méthodes existantes permettant de trouver l'orientation canonique de chaque objet 3D.

– **Analyse en Composantes Principales (ACP)** : L'analyse en composantes principales, aussi appelée "ACP" est couramment utilisée en analyse de données pour trouver les axes principaux d'un nuage de points. Dans le cas des objets 3D, on l'utilise pour calculer les trois axes du repère 3D dans lequel sera représenté le modèle. L'ACP étant la méthode la plus couramment utilisée dans la littérature, plusieurs variantes ont fait leur apparition pour traiter le problème d'alignement des objets 3D. En effet, même si elle est rapide à calculer et robuste dans la plupart des cas, elle peut être instable dans le cas discret d'un maillage. Des améliorations ont donc été proposées pour palier ces problèmes. D'une part, Paquet *et al.* [95] propose de pondérer les centres de gravité des facettes par leur surface, comme c'est le cas dans le calcul du centre de gravité surfacique (voir section 3.2.1). D'autre part, Vranic et Saupe [135][136][134] étendent les travaux de Paquet et calculent l'ACP dans le cas continu, noté "ACPC" ou "ACP Continue" (ACPC), et permet ainsi une meilleure robustesse dans le cas d'un maillage 3D.

  Pour estimer l'alignement d'un modèle 3D il est donc nécessaire de calculer une ACP sur l'ensemble des centres des facettes d'un objet. Pour réaliser cette opération, les objets doivent être composés de polygones dont les sommets comportent toujours trois points coplanaires, c'est-à-dire des objets "triangulés" (voir chapitre 2 section 2.3.2). Le but de l'ACP est de trouver une base où la projection d'un objet (ou nuage de points) est invariante par similitude.

  Pour estimer l'orientation d'un objet, en fonction de ses faces, l'analyse en composantes principales continue définie par Vranic et Saupe [135][136][134] fixe la rotation en déterminant les trois axes principaux de l'objet 3D. Pour cela, il calcule la matrice de covariance $\mathcal{I}$ de dimension $d = 3$ sur l'ensemble des sommets des facettes du maillage $\mathcal{M}$:

$$\begin{aligned} \mathcal{I} &= \frac{1}{\mathcal{S}} \int_{v \in \mathcal{M}} vv^T dv = \frac{1}{\mathcal{S}} \sum_{i=1}^{m} \int_{v \in \mathcal{F}_i} vv^T dv \\ &= \frac{1}{12\mathcal{S}} \sum_{i=1}^{m} \mathcal{S}_i (A_i A_i^T + B_i B_i^T + C_i C_i^T + 9 * G_i G_i^T) \end{aligned} \tag{3.9}$$

  où A_i, B_i, C_i et G_i représentent les vecteurs de dimensions d, correspondant aux coordonnées x, y et z des points, A, B et C ainsi que du centre de gravité surfacique $\mathcal{G}$ de la face $\mathcal{F}_i$.

Le principe est de trouver l'axe u, qui maximise la variance du nuage de points. La variance de la projection de la surface $\mathcal{M}$ de $\mathcal{O}$ sur u s'écrit :

$$u^T \mathcal{I} u \tag{3.10}$$

La matrice $\mathcal{I}$ étant carrée, symétrique et définie positive, elle est alors diagonalisable dans une base orthonormée. Notons P la matrice des vecteurs propres et Δ la matrice diagonale issue de $\mathcal{I}$ composée de trois valeurs propres $\lambda_1,\ \lambda_2$ et λ_3 rangées par ordre décroissant $\lambda_1 > \lambda_2 > \lambda_3$:

$$u^T P^T \Delta P u = (Pu)^T \Delta \underbrace{(Pu)}_{v} \tag{3.11}$$

Le maximum de la variance est alors obtenu pour le vecteur propre v_1 associé à la plus grande valeur propre λ_1 :

$$v_1^T \mathcal{I} v_1 = \lambda_1 \tag{3.12}$$

De même, le vecteur qui explique la plus grande part de l'inertie restante est le second vecteur propre et le troisième exprime l'inertie résiduelle. Finalement, pour définir l'orientation de l'objet 3D, il suffit de diagonaliser la matrice de covariance des faces, ayant subi une normalisation en centre et en échelle. La matrice de passage P est donc la matrice de rotation qui définit l'alignement du modèle 3D.

Même si la version continue de l'ACP formulée par Vranic permet de résoudre les problèmes liés au maillage de l'objet 3D, il reste des incertitudes sur l'ordre des axes et leur orientation. Vranic [134] propose, pour résoudre ce problème, une méthode permettant de définir l'orientation des trois axes principaux. Pour cela il cherche le signe des moments d'ordres 2 le long des trois axes principaux calculés précédemment. Cela consiste à calculer les distances signées entre les faces de l'objet $\mathcal{O}$ et les trois plans principaux définis par les axes $\overrightarrow{x}\overrightarrow{y}$, $\overrightarrow{y}\overrightarrow{z}$ et $\overrightarrow{x}\overrightarrow{z}$. Cela revient à déterminer une matrice diagonale M de valeurs : $sign(f_x),\ sign(f_y)\ et\ sign(f_z)$. Le calcul des signes des axes x et y est le suivant :

$$\begin{aligned} f_t &= \frac{1}{\mathcal{S}} \iint_{v \in \mathcal{O}} sign(v_t) \cdot {v_t}^2 dv \\ &= \frac{1}{\mathcal{S}} \sum_{i=1}^{m} \iint_{v \in \mathcal{F}_i} sign(v_t) \cdot {v_t}^2 dv \\ &= \frac{1}{6\mathcal{S}} \sum_{i=1}^{m} \mathcal{S}_i \cdot M_i^t \end{aligned} \tag{3.13}$$

où $t = x$, y ou z et avec :

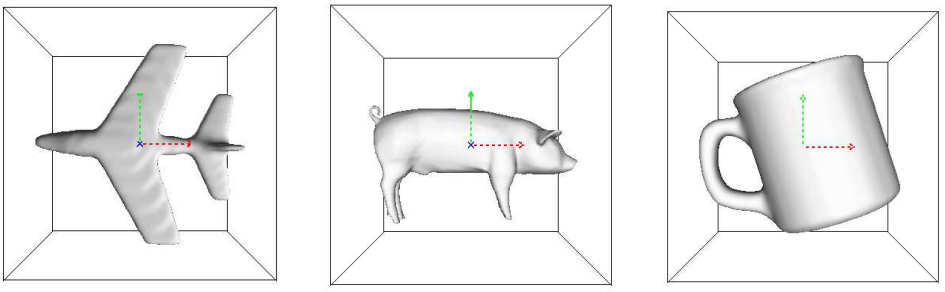
$$M_i^t = \begin{cases} sign(t_{A_i})J_i^t & si\ sign(t_{A_i}) = sign(t_{B_i}) = sign(t_{C_i}) \\ sign(t_{A_i})(-J_i^t + 2L_i^{tA}) & si\ sign(t_{A_i}) \neq sign(t_{B_i}) = sign(t_{C_i}) \\ sign(t_{B_i})(-J_i^t + 2L_i^{tB}) & si\ sign(t_{B_i}) \neq sign(t_{C_i}) = sign(t_{A_i}) \\ sign(t_{C_i})(-J_i^t + 2L_i^{tC}) & si\ sign(t_{C_i}) \neq sign(t_{A_i}) = sign(t_{B_i}) \end{cases} \tag{3.14}$$

où J_i^t et L_i^{t*} valent :

$$J_i^t = (t_{A_i})^2 + (t_{B_i})^2 + (t_{C_i})^2 + t_{A_i}t_{B_i} + t_{A_i}t_{C_i} + t_{B_i}t_{C_i} \tag{3.15}$$

$$\begin{aligned} L_i^{tA} &= \frac{(t_{A_i})^4}{(t_{B_i} - t_{A_i})(t_{C_i} - t_{A_i})}, \\ L_i^{tB} &= \frac{(t_{B_i})^4}{(t_{A_i} - t_{B_i})(t_{C_i} - t_{B_i})}, \\ L_i^{tC} &= \frac{(t_{C_i})^4}{(t_{A_i} - t_{C_i})(t_{B_i} - t_{C_i})} \end{aligned} \tag{3.16}$$

Finalement, l'ACP est un moyen simple de calculer l'orientation d'un objet 3D en calculant ses trois axes principaux. Il est cependant nécessaire de noter que même si elle reste efficace dans de nombreux cas, l'ACP n'assure pas une orientation pertinente pour tous les objets. En particulier, elle est souvent peu robuste à l'ordre des axes et à leur orientation, ou signe (voir figure 3.2).



(a) *Un avion* (b) *Un cochon* (c) *Une tasse*

FIGURE 3.2 – *Exemple de trois objets alignés avec l'ACP Continue de Vranic [134]. (a) représente un avion, (b) un cochon et (c) une tasse. Nous voyons d'ailleurs que ce dernier n'est pas correctement aligné avec l'ACPC à cause de son anse.*

– **Analyse en Composantes Principales par Normales** : Aussi appelé "ACPN", ce moyen d'orienter l'objet 3D est très proche de l'ACP présentée précédemment. Dans l'ACPN, le calcul des axes principaux du modèle 3D ne se fait plus sur les

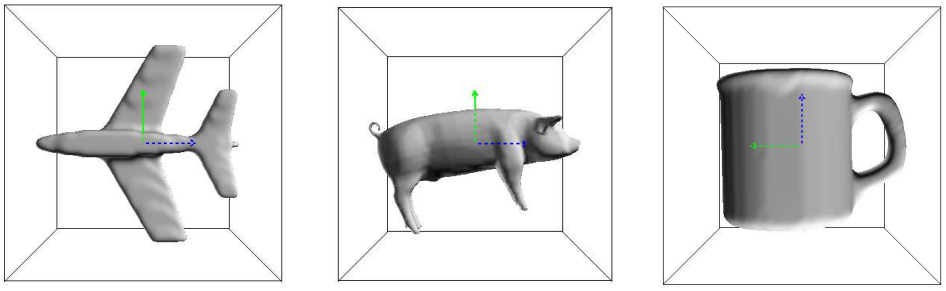
sommets des facettes, mais sur les normales de celles-ci. Dans ce cas, la matrice de covariance à diagonaliser devient :

$$\mathcal{M} = \frac{1}{\mathcal{S}} \sum_{i=1}^{m} \mathcal{S}_i (N_i N_i^T) \tag{3.17}$$

où N_i représente le vecteur de dimension d, correspondant aux coordonnées x, y et z de la normale $\mathcal{N}$ de la face $\mathcal{F}_i$ donnée par :

$$\mathcal{N}_i = \frac{\overrightarrow{A_iC_i} \wedge \overrightarrow{A_iB_i}}{\|\overrightarrow{A_iC_i} \wedge \overrightarrow{A_iB_i}\|} \tag{3.18}$$

Cette ACP est très utile pour les modèles comportant des symétries, mais hérite des mêmes soucis que l'ACP classique sur les sommets, à savoir l'ordre des axes et leur orientation (voir figure 3.3).



(a) *Un avion* (b) *Un cochon* (c) *Une tasse*

FIGURE 3.3 – *Exemple de trois objets alignés avec l'ACPN. (a) représente un avion, (b) un cochon et (c) une tasse. Nous voyons que le deuxième objet n'est pas correctement aligné avec l'ACPN.*

– **Orientation par axes de symétrie** : Pour orienter un objet 3D, Podolak *et al.* [97] proposent de calculer les plans de symétrie du modèle. Pour cela, ils définissent la notion de symétrie pour les plans intersectant l'objet à travers le calcul de la "transformée de symétrie de réflexion planaire" ("PRST") qu'ils introduisent. Ils choisissent ensuite comme premier axe, la normale du plan qui a la symétrie maximale. Le second est alors le plan de symétrie maximale orthogonal à la normale sélectionnée précédemment. Finalement, le dernier axe est obtenu de manière similaire en recherchant le plan de symétrie maximale ayant une normale perpendiculaire aux normales des deux premiers.

  Même si cette formulation offre généralement une orientation plus naturelle des objets 3D, son point faible reste son coût de calcul important.

– **Orientation des objets manufacturés en position debout** : Dans cette approche, Fu *et al.* [39] se sont posés la question de savoir comment les humains ar-

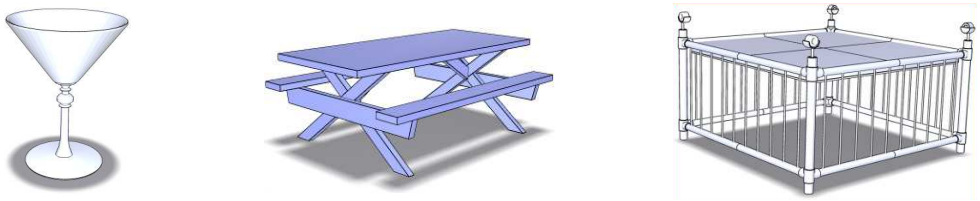
rivent à orienter un objet 3D de manière automatique. Ils se sont alors rendu compte que la fonction de l'objet joue un rôle déterminant dans sa pose. Cette fonctionnalité étant similaire pour des objets de la même classe, ils ont décidé d'apprendre les orientations au moyen d'un algorithme d'apprentissage. Dans cette méthode, on ne cherche plus l'orientation complète d'un objet, mais sa base, qui permet de le poser sur une surface.

Considérant un ensemble d'objets dont on connaît la base, comment faire pour inférer la base d'un nouvel objet ? Les principes de reconnaissance de forme pourraient retrouver l'objet le plus ressemblant, mais cela demanderait un temps trop important pour la phase d'alignement. Ainsi, les auteurs ont décidé de ne considérer, pour chaque modèle 3D, qu'un sous-ensemble de bases possibles et d'utiliser ensuite les données d'apprentissage. Pour réduire l'espace des possibilités, un calcul de l'enveloppe convexe associé à une segmentation de celui-ci en surfaces planes permet de ne garder qu'une douzaine de bases.

Afin de sélectionner une base parmi les autres, quatre informations sont associées à chacune :
- La stabilité : elle est calculée en projetant le centre de gravité de l'objet et de l'enveloppe convexe sur la base.
- La symétrie : en utilisant les travaux de Podolak *et al.* [97], les plans de symétrie sont définis comme étant perpendiculaires à la base et passant par l'un des quatre points caractéristiques suivants : le centre de gravité projeté sur la base, le barycentre des polygones de la base, le barycentre de la projection de l'enveloppe convexe sur la base et le barycentre de la base.
- Le parallélisme : cette notion correspond au nombre de polygones parallèles à la base.
- La visibilité : elle est définie comme étant le pourcentage de l'objet qui reste visible lorsqu'on le pose sur la base.

Finalement, pour comparer chacune des bases sélectionnées associées à leurs attributs, une classification est mise en place. Les méthodes utilisées sont une forêt aléatoire associée à un "SVM". Les résultats obtenus sont très robustes pour l'ensemble des objets manufacturés, mais nécessite un temps de calcul important dû à l'extraction des différentes bases et de leurs attributs. Nous pouvons cependant noter que même si cette méthode d'alignement fonctionne bien pour certaines classes d'objets, elle reste difficile à mettre en place dans le cas d'objets quelconques (voir figure 3.4).

- **Alignement basé sur les propriétés de symétrie** : Dans cette approche, Chaouch [24][27] propose d'orienter un objet à partir de ses propriétés de symétrie. Cependant, afin d'améliorer les temps de calcul et de tirer parti des points forts de l'ACP Continue définie par Vranic [134], il propose une méthode en deux temps.



(a) *Un verre* (b) *Une table de jardin* (c) *Un parc*

FIGURE 3.4 – *Exemple de trois objets alignés selon la position debout. (a) représente un verre, (b) une table de pique-nique et (c) un parc d'enfant. Nous voyons d'ailleurs que ce dernier n'est pas correctement aligné avec l'algorithme de position debout.*

Dans la première phase, l'idée est d'étudier les symétries de réflexions détectées via l'ACPC. Grâce à cette étude, il est possible de classer l'objet 3D dans l'une des cinq familles d'objets polyédriques défini par Cromwell [31] (voir figure 3.5) :

1. G_C : Les objets 3D ayant une symétrie cyclique. Dans ce cas, n plans miroirs ($n > 1$) sont détectés. Dans le cas où n est pair, 2 axes principaux sont retenus, 1 sinon.
2. G_D : Les objets 3D ayant une symétrie diédrique. Dans ce cas, n plans miroirs ($n > 1$) sont détectés, ainsi qu'un plan perpendiculaire. Dans le cas où n est pair, 3 axes principaux sont retenus, 2 sinon.
3. G_R : Les objets 3D ayant une symétrie polyédrique. Ce groupe comporte trois sous-groupes : G_T symétrie tétraédrique (1 axe retenu), G_O symétrie octaédrique (3 axes retenus) et G_I symétrie icosaédrique (1 axe retenu).
4. G_U : Les objets 3D ayant une seule symétrie miroir. Dans ce cas, 1 axe est retenu.
5. G_Z : Les objets 3D n'ayant aucune symétrie miroir. Dans ce cas aucun axe n'est retenu.

Dans une seconde phase, un "coût d'invariance locale de translation" ("CILT") est calculé pour connaître les autres axes. Pour cela, l'objet est décrit sur différentes directions au moyen d'un descripteur, ensuite une mesure de symétrie est appliquée, pour caractériser les invariances locales de translation dans cette direction. Finalement, la direction choisie est celle qui maximise le CILT. Le processus complet est décrit par l'algorithme 3.1.

Pour conclure, cette méthode d'alignement est robuste et permet d'aligner toutes les classes d'objets. Cependant, l'orientation donnée ne tient pas compte de la fonctionnalité des modèles et peut donc renvoyer un alignement incohérent par rapport à la sémantique. De plus, le coût de calcul est important, en particulier lorsque $n = 0$ (voir figure 3.6).

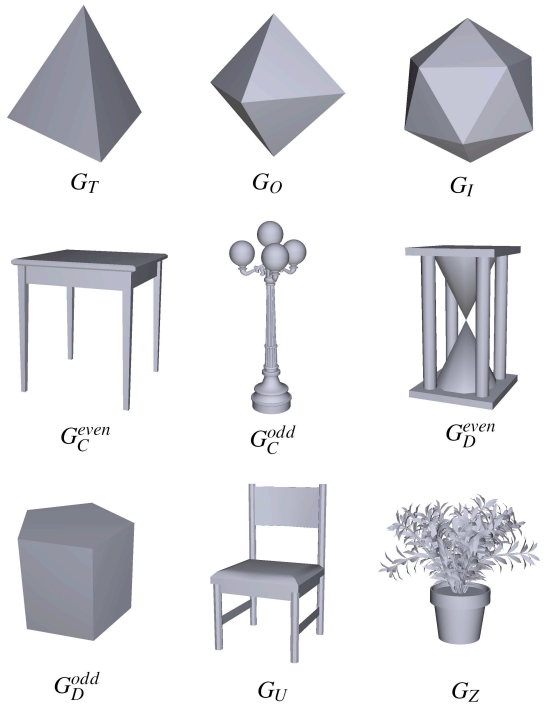
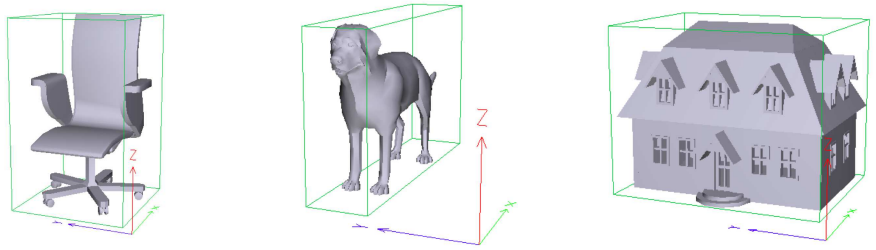


FIGURE 3.5 – *Exemples de modèles 3D appartenant aux différentes classes de symétrie miroir : tétraèdre* $\in G_T$*, octaèdre* $\in G_O$*, icosaèdre* $\in G_I$*, table* $\in G_C^{even}$*, lampadaire* $\in G_C^{odd}$*, sablier* $\in G_D^{even}$*, 5-prisme* $\in G_D^{odd}$*, chaise* $\in G_U$ *et plante* $\in G_Z$*.*

Algorithme 3.1 Etapes de l'alignement

1: Tester la réflexion des trois plans principaux déduit de l'ACPC.
2: Sélectionner les n axes d'orientation pertinents suivant la classe de l'objet.
3: **Si** $n \geq 2$ **Alors**
4: Retourner les trois axes d'alignement.
5: **Fin Si**
6: **Si** $n = 1$ **Alors**
7: Retourner le premier axe d'alignement.
8: **Fin Si**
9: **Si** $n = 0$ **Alors**
10: Retourner le premier axe d'alignement obtenu en calculant la réflexion maximale sur l'objet 3D.
11: **Fin Si**
12: **Si** $n \in \{0, 1\}$ **Alors**
13: Chercher la direction qui maximise le coût d'invariance locale de translation (CILT).
14: **Fin Si**



(a) *Une chaise* (b) *Un chien* (c) *Un bâtiment*

FIGURE 3.6 – *Exemple de trois objets alignés selon ses axes de symétries. (a) représente une chaise, (b) un chien et (c) une maison.*

3.3 Classification des méthodes d'indexation 3D

L'analyse des différentes techniques d'indexation d'objets 3D montre qu'il existe de nombreux types de signatures permettant de décrire les informations intrinsèques à chaque modèle. Dans ce cadre, nous avons choisi de classer, dans notre état de l'art, les méthodes existantes en trois groupes, séparant les méthodes basées sur l'information véhiculée par la géométrie de l'objet, de celles qui utilisent des projections 2D de l'objet associées à des informations de profondeurs et de celles qui ne nécessitent finalement que d'informations en deux dimensions. Le découpage choisi est le suivant :

- **Les approches 3D/3D** : Ce type d'approche utilise l'information 3D contenue dans la représentation du modèle 3D pour calculer une description pertinente de sa forme. Le choix de cette signature fait apparaître cinq groupes au sein des approches 3D. Les méthodes "globales" qui cherchent à décrire l'objet dans son ensemble, contrairement aux approches "locales" qui basent leur description sur des informations locales de sa forme. Il y aussi les approches qui partitionnent l'espace afin de calculer la description et celles basées sur une transformée ou un calcul de moments. Finalement, il y a les approches par "squelette" qui décrivent la forme intrinsèque de l'objet 3D à l'aide d'un graphe par exemple (voir figure 3.7).

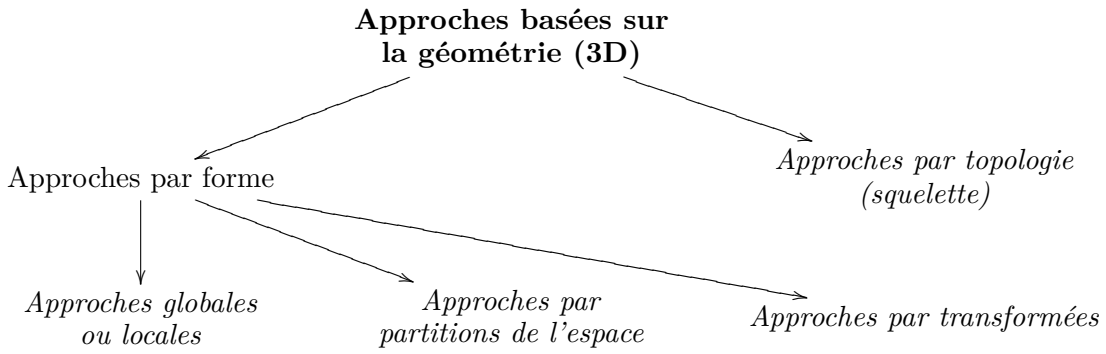


FIGURE 3.7 – *Classification des approches 3D existantes en trois sous-groupes.*

– **Les approches "2.5D/3D"** : Ces méthodes utilisent des projections 2D de l'objet 3D associées à des informations 3D pour décrire la forme de l'objet, d'où leur appellation deux dimensions et demi. Dans la plupart des cas, il s'agit d'informations de profondeurs ou de courbures, encapsulées dans une image ou dans une série de coupes de l'objet 3D. Nous avons donc séparé cette partie en trois groupes : les approches par images de profondeurs, celles par cartes de courbures et celles par coupes (voir figure 3.8).

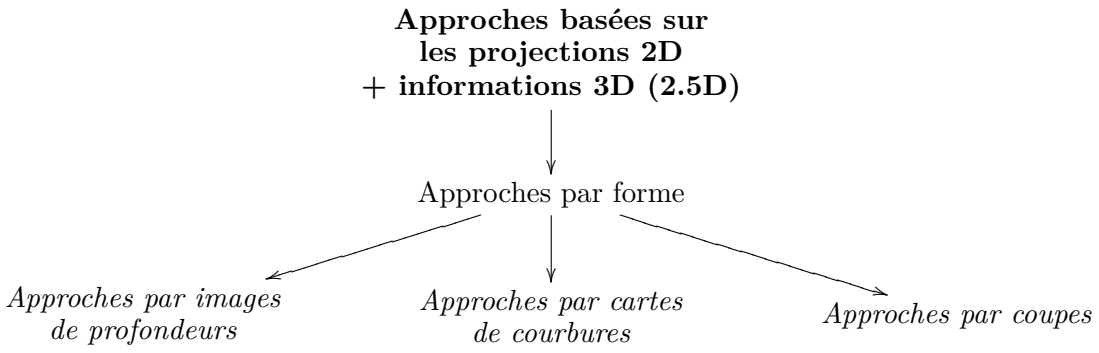


FIGURE 3.8 – *Classification des approches 2.5D existantes en trois sous-groupes.*

– **Les approches 2D/3D** : Ce type de méthode base sa description uniquement sur les informations contenues dans la projection de l'objet 3D sur un plan 2D. La majeure partie de ces approches utilise non pas une, mais plusieurs projections afin de décrire la forme de l'objet dans différents angles de vues. Nous séparons ces approches en deux groupes : les méthodes basées sur les régions et celles utilisant les contours (voir figure 3.9).

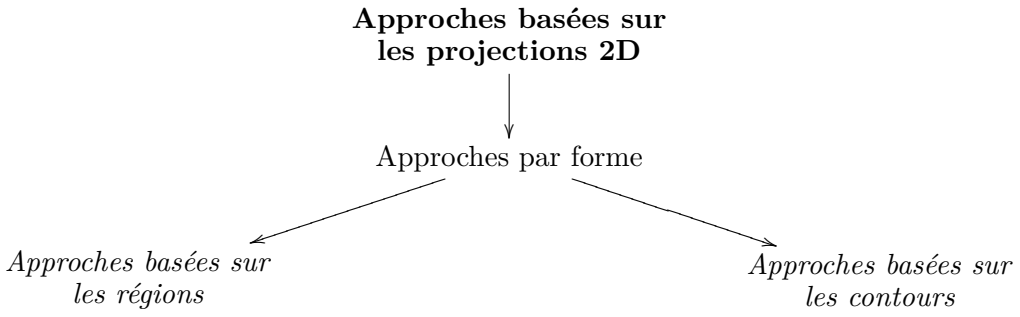


FIGURE 3.9 – *Classification des approches 2D existantes en deux sous-groupes.*

L'état de l'art que nous dressons ici ne pouvant être exhaustif, nous invitons le lecteur à consulter les travaux de Bimbo et Pala [16], Bustos *et al.* [19], Tangelder et Veltkamp [118], Shilane *et al.* [111], Dugelay *et al.* [34] et Zaharia et Prêteux [140], qui effectuent une comparaison entre différentes approches d'indexation et de recherche d'objets 3D. En dernier lieu, et pour connaître les performances des méthodes récentes sur des bases de données spécifiques, nous conseillons de regarder les résultats des concours annuels *SHREC*

qui ont pour but d'évaluer sur un même jeu de données les approches des différentes équipes de recherches (voir annexe D). Dans ce sens, le lecteur pourra consulter les évaluations de Shilane *et al.* [111], Veltkamp et Haar [130] et Godil *et al.* [45] pour des études faites sur des bases d'objets 3D dites "généralistes" (voir annexe D) ou les articles de Veltkamp et Haar [129], Biasotti et Attene [12], Muthuganapathy et Ramani [73] et Hartveldt *et al.* [46], pour des évaluations sur de bases de documents 3D dites "spécifiques" ("CAO", modèles articulés...) (voir annexe D).

3.4 Etat de l'art sur les méthodes d'indexation 3D/3D

Dans cette partie, nous nous intéressons à la première famille d'approches dégagées précédemment, dans la section 3.3, à savoir, les méthodes cherchant à extraire la forme d'un objet 3D directement à partir du modèle en trois-dimensions. Nous présentons donc ici un état de l'art non exhaustif des méthodes importantes résolvant le problème de l'indexation de formes 3D. Nous organisons cet inventaire selon l'ordre suivant :

1. Méthodes globales (3.4.1).
2. Méthodes locales (3.4.2).
3. Méthodes par partitions de l'espace (3.4.3).
4. Méthodes par transformées (3.4.4).
5. Méthodes par squelettes (3.4.5).

3.4.1 Méthodes globales

Histogramme de cordes : Dans cette approche Paquet et Rioux [93] proposent de capturer la forme d'un objet 3D en étudiant la direction de ses faces. Afin d'obtenir une mesure efficace, ils utilisent deux descriptions distinctes. La première est construite sur l'angle de la normale de chacune des faces, tandis que la seconde est basée sur la direction des cordes entre le centre de masse de l'objet et ses faces. Pour chaque description, les auteurs synthétisent l'information au moyen de trois histogrammes : le premier pour l'angle par rapport aux deux axes principaux de l'objet, le second distingue ces deux axes en construisant un histogramme pour chacun d'eux, le troisième est bidimensionnelle et fait la relation entre ces mêmes axes.

Même si cette méthode est pertinente pour les objets uniformément maillés, elle reste insuffisante dans le cas de modèles où la connectivité et la précision du maillage sont chaotiques (absence de face, faces de tailles différentes, faces double...). Elle reste intéressante par sa rapidité et son pouvoir discriminant dans le cas d'objets uniformément maillés tels que ceux issus de scanners 3D.

Distributions de forme : Afin de mesurer les propriétés géométriques d'un objet 3D Osada *et al.* [90][91] transforment, dans le descripteur "D2", un objet 3D en une distribution de forme. Les auteurs ont évalué cinq fonctions basées sur des ensembles aléatoires de points qui sont :

1. L'angle entre trois points de la surface.
2. La distance entre le centre et un point du modèle.
3. La distance entre deux points.
4. La racine carrée de l'aire d'un triangle formé par trois points.
5. La racine cubique du volume du tétraèdre formé par quatre points.

Les auteurs montrent que la troisième distribution semble la meilleure. Les résultats obtenus avec cette approche permettent de distinguer la forme globale de l'objet sans avoir un pouvoir discriminant suffisant pour capturer les petites variations du maillage (voir figure 3.10). Cependant, grâce à sa rapidité et son invariance aux transformations géométriques elle peut être utilisée en amont d'autres méthodes afin d'élaguer une partie des objets 3D de la base de données.

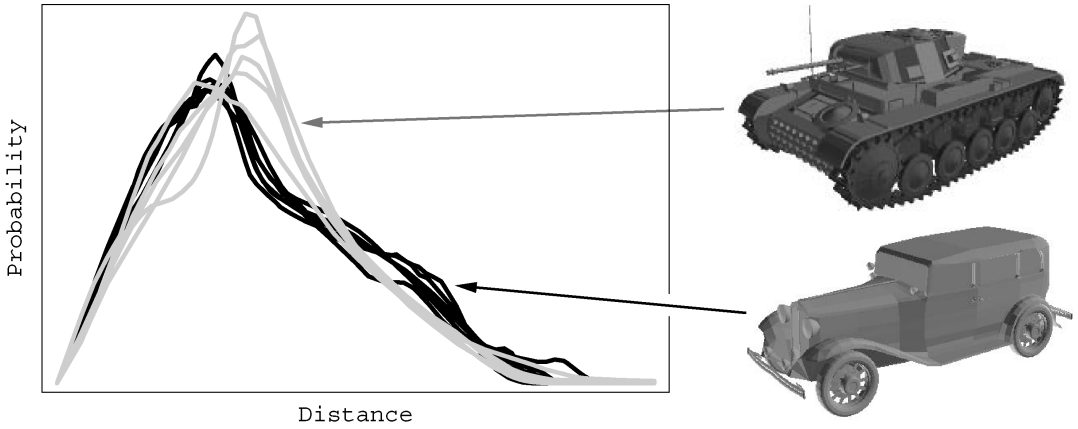


FIGURE 3.10 – *Distribution de forme de cinq tank (en gris) et de six voitures (en noir).*

Une amélioration de cette approche a été faite par Ohbuchi *et al.* [86] où la distribution de forme est caractérisée par la distance entre deux points aléatoires du modèle 3D et l'angle entre les normales des deux faces auxquelles appartiennent les points. Les auteurs montrent que cette extension offre des performances nettement supérieures (13% pour le "Plus proche voisin") avec un coût très légèrement plus important.

3.4.2 Méthodes locales

Images gaussiennes étendues : Cette représentation introduite en 1983 par Horn [51], et souvent notée "EGI" pour *Extended Gaussian Images*, permet de coder la forme d'un objet 3D. Basé sur une fonction sphérique, ce descripteur partitionne la sphère unité (sphère de Gauss) en espaces angulaires égaux. Chaque facette du modèle 3D contribue alors, par son aire, à la composante de l'histogramme qui représente la direction de sa normale. L'un des principaux inconvénients de cette approche est sa dépendance très forte

à l'orientation des faces. En effet, supposons qu'un objet 3D soit représenté d'une part sous une forme voxelisée, et d'autre part sous sa forme originale maillée. Alors, le descripteur "EGI" sera totalement différent à cause de l'orientation différente des facettes dans les deux représentations.

Pour améliorer cette version des "EGI", Kang et Ikeuchi [59] puis Wang *et al.* [137] ajoutent une information de distance dans leur descripteur qu'ils nomment respectivement "CEGI" et "MSEGI". Cette nouvelle donnée permet de pondérer la contribution de chaque facette et améliore, de ce fait, la pertinence des résultats.

Finalement, même avec les améliorations apportées aux "EGI" originales, ce type d'approche reste sensible à l'orientation définie par la topologie des objets 3D et nécessite, d'autre part, une étape de normalisation pour garantir l'invariance par rapport aux transformations géométriques. Il est donc préférable de les utiliser en plus d'autres méthodes plutôt que seules.

Spectre de forme : Introduit par Zaharia et Preteux [142][144] et défini comme la distribution d'un index de forme caractérisant localement la géométrie d'une surface 3D, le spectre de forme "SF3D" permet de décrire la forme d'un objet 3D. L'index de forme présenté par Koenderink et van Doorn [64] est défini comme la valeur de la coordonnée angulaire de la représentation polaire du vecteur des courbures principales. Soient p un point d'une surface 3D et k_p^1 et k_p^2, avec $k_p^1 \geq k_p^2$, les courbures principales en ce point alors l'index de forme IF_p s'exprime par la formule :

$$IF_p = \frac{1}{2} - \frac{1}{\pi} \arctan \frac{k_p^1 + k_p^2}{k_p^1 - k_p^2} \tag{3.19}$$

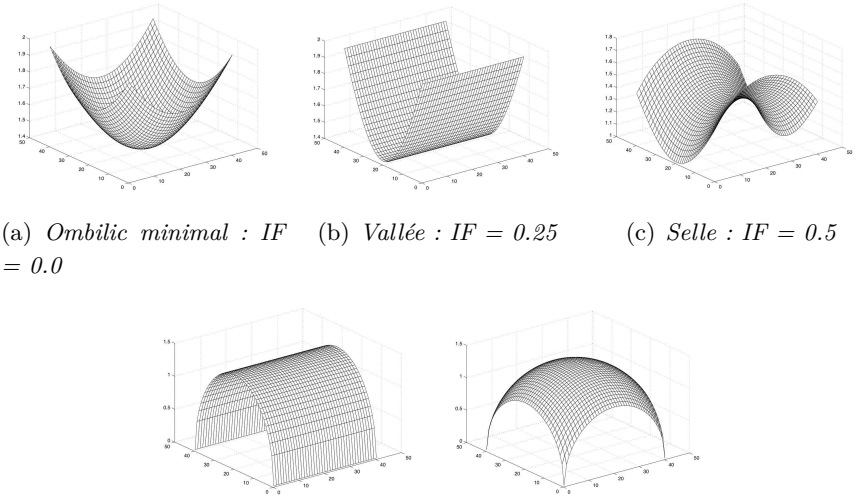
L'index de forme permet de caractériser différentes formes élémentaires (voir figure 3.11). Finalement, chaque facette du maillage contribue par son aire à la composante du maillage qui appartient à l'intervalle de l'index de forme associé à chaque composante d'un histogramme. Afin de ne pas négliger les cas où l'index de forme n'est pas défini (surface plane par exemple), les auteurs introduisent des notions supplémentaires basées sur l'aire de ces facettes.

Même si ce nouveau descripteur est très compact, il est pénalisé par sa grande sensibilité à la topologie ainsi que par sa nécessité d'avoir des objets dont le maillage est uniforme.

3.4.3 Méthodes par partitions de l'espace

Histogrammes de forme : Ankerst *et al.* [6] propose une façon de partitionner uniformément l'espace selon trois représentations (voir figure 3.12) :

1. Coquilles "SHELLS" : Ce partitionnement permet, au moyen de coquilles concentriques autour du centre de l'objet, de s'affranchir d'éventuelles rotations de celui-ci. Afin de pouvoir englober l'ensemble du modèle, la dernière coquille n'a pas de frontière supérieure.

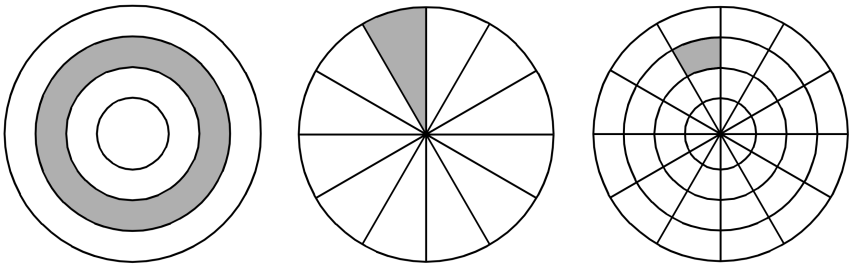


(a) *Ombilic minimal : IF = 0.0* (b) *Vallée : IF = 0.25* (c) *Selle : IF = 0.5*

(d) *Crête : IF = 0.75* (e) *Ombilic maximal : IF = 1.0*

FIGURE 3.11 – *Formes élémentaires et leur indice de forme "IF" associé pour le point p de cordonnée* $x = 25$ *et* $y = 25$.

2. Secteurs "SECTORS" : Cette représentation basée sur une décomposition angulaire de taille homogène est calculée en projetant les faces d'un polyèdre régulier sur la sphère unité.
3. Combinés "SECSHEL" : Cette dernière décomposition est la combinaison des deux premières alliant ainsi les forces de chacune.



(a) *4 coquilles* (b) *4 secteurs* (c) *combinaison coquilles/secteurs*

FIGURE 3.12 – *Représentation des trois partitionnements proposés : par coquilles, par secteurs et leur combinaison (coquilles + secteurs).*

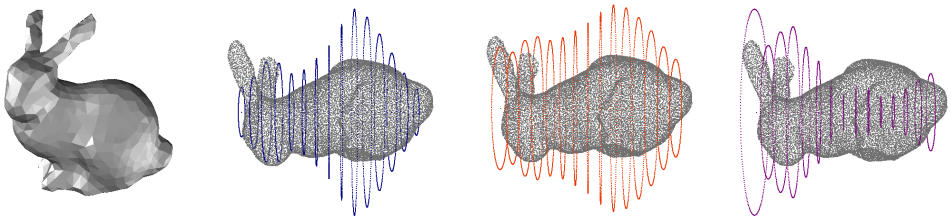
Finalement, chaque descripteur est un histogramme représentant la répartition du tirage d'un ensemble de points uniformément répartis sur la surface de l'objet 3D.

Évalués sur la base du *Princeton Shape Benchmark* [110], les résultats du descripteur "SECSHEL" sont meilleurs que "SECTORS" ainsi que "SHELLS" et se place dans la

moyenne des autres approches testées.

Statistiques sur les axes principaux : Dans leur approche, Ohbuchi *et al.* [89] décrivent la forme d'un objet 3D en calculant des statistiques le long des trois axes principaux de l'objet. Ils partitionnent chacun d'eux en N composantes et calculent pour chacune trois statistiques (voir figure 3.13) :

1. Le moment d'inertie autour de l'axe.
2. Le distance moyenne entre la surface du modèle et l'axe.
3. Le variance de la distance entre la surface du modèle et l'axe.



(a) *Objet 3D* (b) *Moment d'inertie autour de l'axe* (c) *Distance moyenne de la surface à l'axe* (d) *Variance de la distance de la surface à l'axe*

FIGURE 3.13 – *Exemples de statistiques autour d'un des axes principaux. (a) représente l'objet 3D, (b) le moment d'inertie autour de l'axe, (c) la distance moyenne de la surface à l'axe et (d) la variance de la distance de la surface à l'axe. Les statistiques sont évaluées sur une partition de l'axe* $\overrightarrow{x}$ *en* $N = 15$ *composantes.*

Le descripteur final est donc composé de neuf histogrammes, trois par axe, calculés après une normalisation de l'objet en échelle et en rotation. Afin de garantir la mise en correspondance robuste de deux descripteurs, les auteurs basent leur mesure de dissimilarité sur la distance euclidienne associée à une mesure élastique par programmation dynamique.

Le descripteur proposé par les auteurs est robuste aux petites déformations du maillage et ne nécessite pas que l'objet soit correctement maillé. En revanche cette description est trop fortement liée au calcul des axes principaux, ce qui reste une étape difficile et nécessite un temps de calcul important lié à la mise en correspondance élastique.

Points saillants : Afin de capturer l'information de forme d'un modèle 3D, Tangelder et Veltkamp [117] représentent l'objet par un ensemble de points saillants pondérés. Après une normalisation par analyse en composantes principales, le cube unité est partitionné uniformément en $25 * 25 * 25$ cellules, permettant ainsi la construction d'un histogramme. Afin de sélectionner un point saillant pour chacune d'entre elles ainsi qu'un poids, les auteurs comparent trois approches :

1. Le point ayant la courbure Gaussienne la plus forte avec comme poids cette mesure en valeur absolue.

2. La position des facettes de la cellule pondérée par leur aire. Le poids correspond ici à la variation moyenne des normales.
3. Le centre de masse de tous les sommets appartenant à la cellule avec un poids fixé à 1.

Afin de prendre en compte les petites déformations, les auteurs utilisent une version modifiée de la distance "EMD" afin de construire une distance (voir annexe A). Cette nouvelle mesure apporte le principal inconvénient de l'approche, à savoir son coût de calcul.

Contextes de forme 3D : Initialement introduit en deux dimensions, les contextes de forme permettent de décrire localement la forme d'un objet. Kortgen *et al.* [65] appliquent ce principe à un modèle 3D normalisé en décrivant le voisinage de N points uniformément répartis sur la surface de l'objet 3D. Pour chacun d'entre eux, les auteurs construisent un histogramme qui décrit la position des $N-1$ autres points. Afin de coder la notion de position, ils utilisent la décomposition en "coquilles/secteurs" faite par Ankerst *et al.* [6]. Ainsi, le descripteur final est un ensemble de $N = 200$ histogrammes de $C = 432$ composantes ("coquilles/secteurs").

En plus du manque de pertinence de la description, cette approche est très coûteuse en espace mémoire et en temps de calcul. En effet, afin de comparer les différents histogrammes, les auteurs proposent de rechercher le meilleur appariement entre ceux-ci d'une complexité proche de $\mathcal{O}(N^3)$.

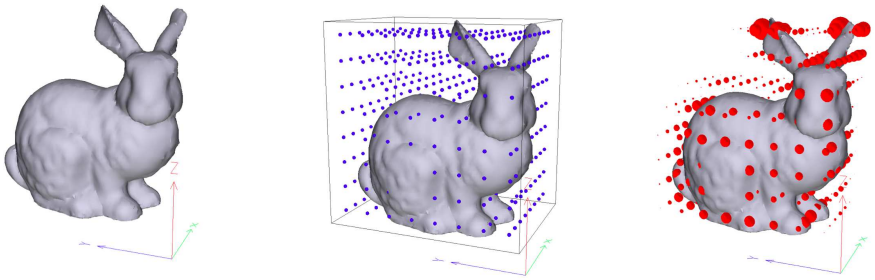
Descripteur Gaussien 3D : Le principe du descripteur "3DGA" introduit par Chaouch [24] est de caractériser et d'amplifier localement le voisinage de la surface 3D. Pour cela, l'auteur définit des fonctions gaussiennes qui mesurent l'influence des points de la surface, des objets 3D, sur des cellules régulièrement réparties dans l'espace englobant le modèle. À partir d'un échantillonnage de la boite englobant l'objet 3D en N^3 cellules, représentées par leur centre q_{ijk}, l'auteur calcul la contribution de la surface $\mathcal{M}$ du modèle envers chaque cellule, avec σ la bande, comme suit :

$$g_{ijk}(q_{ijk}, \mathcal{M}, \sigma) = \iint_{p \in \mathcal{M}} e^{\frac{-|p-q_{ijk}|_2}{\sigma^2}} dp \tag{3.20}$$

Ce descripteur compact, illustré sur la figure 3.14, a montré sa robustesse par rapport aux autres méthodes "3D/3D" et se place à la première position sur la base *Princeton Shape Benchmark* pour ce type d'approche.

3.4.4 Méthodes par transformées

Moments géométriques : Les moments géométriques permettent de décrire la forme générale d'un maillage 3D. Leur utilisation pour l'indexation de modèles tridimensionnels apparait pour la première fois dans l'article de Paquet et Rioux [94] où ils sont calculés après une normalisation en rotation car ils ne sont pas invariants envers cette transformation. Dans cet article les auteurs calculent les moments sur la surface discrétisée de l'objet



(a) *Objet représentant un lapin* (b) *Echantillonnage de la boite englobante* (c) *Représentation des coefficients g_{ijk}*

FIGURE 3.14 – *Exemple du calcul du descripteur "3DGA". (a) représente l'objet 3D, (b) l'échantillonnage de la boite englobante en 8^3 cellule de centre q_{ijk} (en bleu) et (c) montre la contribution des cœfficients g_{ijk} (les boules rouges sont proportionnels aux cœfficients).*

par la formule :

$$M_{qrs} = \sum_{i=1}^{m} \mathcal{S}_i (x_i - x_{\mathcal{G}_i})^q (y_i - y_{\mathcal{G}_i})^r (z_i - z_{\mathcal{G}_i})^s \tag{3.21}$$

Voir notation dans la section . L'ordre $(q, r$ et $s)$ choisi dépend de la granularité de description souhaitée et permet ainsi une recherche multi-échelle.

Dans le même esprit, Saupe et Vranic [107] utilisent, dans le descripteur "EXT", des moments géométriques pour représenter une fonction sphérique permettant de capturer la forme des objets 3D. Le calcul des moments se fait avec la formule :

$$M_{qrs} = \sum_{i,j=1}^{m} r(u_{ij}) \Delta s_{ij} x_{ij}^q y_{ij}^r z_{ij}^s \tag{3.22}$$

où $r(u_{ij})$ représente la distance maximale entre le centre de l'objet et sa surface en suivant la direction $u_{ij} = (cos\phi_i sin\theta_j, sin\phi_i sin\theta_j, cos\theta_j)$ et où Δs_{ij} donne l'aire de cette surface sur la boule unité. Les expérimentations faites par les auteurs montrent qu'un ordre de 31 est suffisant pour caractériser la forme des modèles 3D et que l'augmentation de l'ordre tend même a réduire la pertinence de la méthode. Finalement, cette approche reste peu discriminante comme le montrent les comparaisons effectuées par Vranic [134].

Moments de Zernike 3D : Dans leur approche Novotni et Klein [84][85] utilisent les moments de Zernike 3D introduit par Canterakis [22] pour décrire la forme d'un objet 3D à l'aide de fonctions polynomiales dans le même esprit que les moments de Zernike 2D. L'un des principaux intérêts des moments de Zernike est leur invariance à la rotation facilitant ainsi le prétraitement des modèles. Les auteurs procèdent à l'extraction du descripteur sur l'objet voxelisé en quatre étapes distinctes :

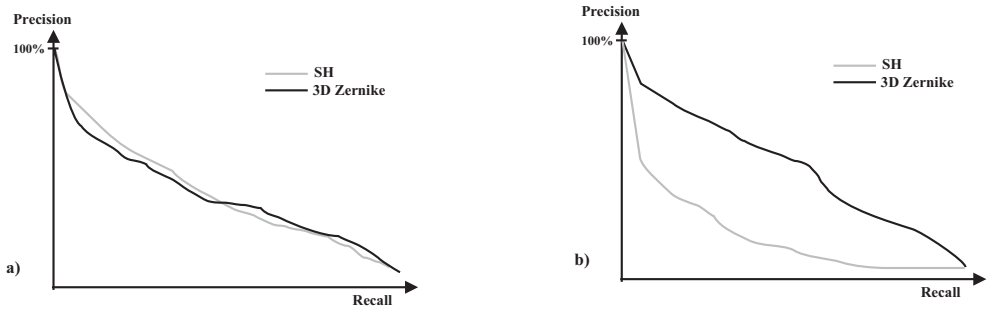
1. Centrage et mise à l'échelle du modèle après calcul du centre de gravité.

2. Calcul des moments géométriques M_{qrs} d'ordre : $q, r, s > 0$ et $q + r + s \leq N$ (voir approche précédente).
3. Calcul des moments de Zernike 3D Ω_{nl}^{m} avec :

$$\Omega_{nl}^{m} = \frac{3}{4\pi} \sum_{q+r+s \leq n} \overline{\mathcal{X}_{nlm}^{qrs}} M_{qrs} \tag{3.23}$$

   où $\mathcal{X}_{nlm}^{qrs}$ représente la combinaison linéaire des moments géométriques.
4. Calcul des normes $F_{nl} = \|\Omega_{nl}\|$ comme descripteur de l'objet.

Des expérimentations faites par les auteurs sur la base du *Princeton Shape Benchmark* [110] montrent des résultats équivalents ou meilleurs que les approches par harmoniques sphériques (voir figure 3.15).



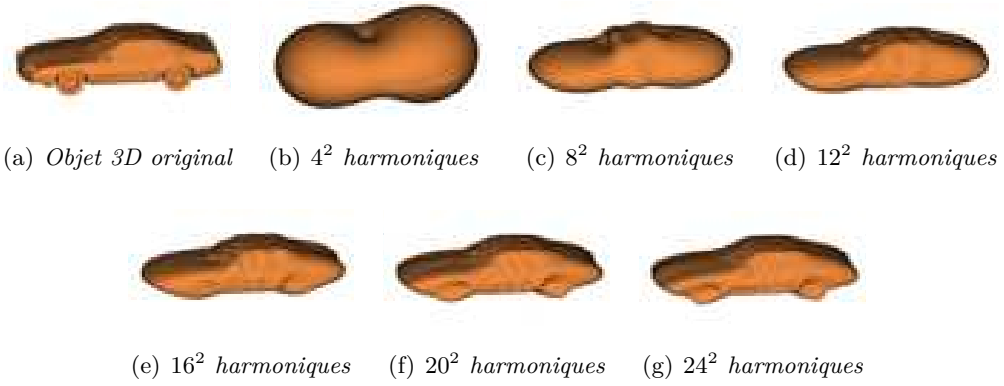
(a) *Classe des "avions"* (b) *Classe des "chaises"*

FIGURE 3.15 – *Graphique précision/rappel des méthodes "Zernike 3D" (en noir) et "harmoniques sphériques" (en gris) pour la classe des "avions" (a) et des "chaises" (b).*

Descripteurs par fonctions sphériques : L'ensemble de ces approches utilisées entre autres par Kazhdan *et al.* [63] et Vranic [134], dans les descripteurs "SHD" et "GEDT", permettent de caractériser un objet 3D par une fonction sphérique. Celle-ci est calculée sur un ensemble de cellules de la sphère unité qui peuvent être un échantillonnage uniforme de sa surface ou de son volume. La fonction sphérique donne alors une description de l'objet pour chacune des cellules définies. Ces fonctions sont souvent la base d'approches plus complexes telles que les harmoniques sphériques ou les descripteurs de symétries.

Harmoniques sphériques : Les harmoniques sphériques permettent de représenter, avec une invariance en rotation, des fonctions calculées sur une sphère S^2. Grâce à celles-ci, il est possible de décomposer toutes fonctions sphériques $f(\theta, \phi)$ comme la somme de ses harmoniques. La grande force des harmoniques sphériques est qu'elles permettent de construire des descripteurs invariants en rotation, simplifiant ainsi les étapes de normalisation des modèles 3D. Il s'agit en quelque sorte d'une transformée de Fourier appliquée aux fonctions sphériques.

Cet outil a été largement utilisé par Vranic *et al.* [136][133][132], par exemple dans le descripteur "REXT", pour la description d'un objet 3D. Afin de décrire la forme du modèle voxelisé, il a considéré plusieurs fonctions sphériques telles que par exemple : $r(u_{ij})$ qui représente la distance maximale entre le centre de l'objet et sa surface en suivant la direction u_{ij}, $s(u_{ij})$ qui est la norme du produit scalaire entre u_ij et la normale unitaire du triangle contenant le point $r(u_ij)u_ij$, ou encore la fonction complexe $r(u_{ij}) + js(u_{ij})$. Les résultats obtenus permettent une bonne robustesse tout en gardant un descripteur compact grâce à la décomposition en harmoniques sphériques (voir figure 3.16).



(a) *Objet 3D original* (b) 4^2 *harmoniques* (c) 8^2 *harmoniques* (d) 12^2 *harmoniques*

(e) 16^2 *harmoniques* (f) 20^2 *harmoniques* (g) 24^2 *harmoniques*

FIGURE 3.16 – *Représentation multi-résolution d'une voiture avec la fonction $r(u_{ij})$ utilisée pour calculer le descripteur avec les harmoniques sphériques.*

Dans le même esprit, Funkhouser *et al.* [41] ont utilisé cette méthode pour coder l'information contenue dans la fonction sphérique définie par l'intersection d'un ensemble de sphères concentriques avec les voxels du modèle 3D. Cette approche a l'inconvénient de dépendre fortement de la résolution de l'objet voxelisé.

Finalement, Kazhdan *et al.* [63][62] ont aussi utilisé les harmoniques sphériques pour rendre invariant en rotation un ensemble de descripteurs 3D existants. Leurs expérimentations se sont entre autres portées sur les "EGI" [51], les fonctions définies par Vranic précédemment [136], la fonction sur les sphères concentriques [41] ou encore les histogrammes de forme [6]. Les principaux intérêts dégagés par les auteurs sont l'invariance en rotation qui améliore légèrement les résultats et la compacité accrue des descripteurs permettant ainsi une recherche plus efficace.

Descripteurs de symétrie : Ce type de descripteur est présenté par Kazhdan *et al.* [60] pour décrire un objet 3D à partir de ses symétries de réflexion. Afin de mesurer cette propriété, les auteurs décrivent le modèle voxelisé avec une fonction sphérique avant de calculer la symétrie de cette description par rapport à l'ensemble des plans passant par le centre de masse de l'objet.

Kazhdan [62][61] généralise ce concept aux symétries à "k-plis" afin de calculer à la fois les symétries de réflexion et de rotation. Les auteurs utilisent toujours une description de

l'objet voxelisé par un ensemble de fonctions sphériques, mais y ajoutent une décomposition en harmoniques sphériques qui garantie l'invariance en rotation. Les résultats obtenus montrent que ce descripteur, associés à un descripteur par harmoniques sphériques, offre de meilleures performances que toutes les autres approches basées sur ce dernier.

Finalement, Podolak *et al.* [97] améliorent cette approche en introduisant la transformée de symétrie de réflexion planaire. Grâce à celle-ci, les auteurs ne se limitent plus aux plans de symétrie passant par le centre de masse de l'objet comme dans [60] mais à l'ensemble des plans intersectant l'objet. Afin de réduire le coût de calcul, ils utilisent un algorithme de Monte-Carlo associé à une étape d'affinement par optimisation itérative locale pour calculer la transformée. Même si cette dernière approche offre des résultats satisfaisants, elle reste lourde à cause de sa complexité de calcul.

Ces approches ont l'avantage d'être peu sensibles au bruit et à la résolution de l'objet, offrant ainsi des résultats intéressants pour la description d'information 3D. Elles peuvent, de plus, être une information appréciable pour l'alignement des objets 3D en vue de leur normalisation.

Transformée de Fourier 3D discrète : Les travaux de Vranic et Saupe [135] apportent un nouveau descripteur de forme 3D invariant aux translations, changements d'échelles, rotations, réflexions et peu sensible aux changements de résolution. Celui-ci est basé sur l'application de la transformée de Fourier 3D discrète (3D DFT) sur un modèle voxelisé. Afin d'obtenir l'invariance aux transformations de l'espace, une analyse en composante principale continue est appliquée en prétraitement de l'extraction du descripteur. La transformée de Fourier 3D se calcule sur les N^3 voxels q_{ikl} de l'objet par la formule suivante :

$$g_{uvw} = \frac{1}{\sqrt{N^3}} \sum_{i=-\frac{N}{2}}^{-\frac{N}{2}+1} \sum_{k=-\frac{N}{2}}^{-\frac{N}{2}+1} \sum_{l=-\frac{N}{2}}^{-\frac{N}{2}+1} q_{ikl} e^{-j\frac{2\pi}{N}(iu+kv+lw)} \tag{3.24}$$

où $-K \leq u,\ v,\ w \leq K$ afin de ne garder que les basses fréquences qui encodent mieux la forme générale de l'objet.

Cette application de la transformée de Fourier, généralement appliquée à des images en deux dimensions, permet de décrire après un passage en 3D la forme d'objets complexes. Les résultats obtenus avec cette approche sont insuffisants comparés aux autres méthodes par transformées.

Transformée de Hough 3D : Le descripteur de Hough 3D a été proposé par Zaharia et Preteux [141][144] afin de décrire les maillages 3D. Cette transformée est fondée sur un principe d'accumulation des points sur des plans de $\mathbb{R}^3$. En échantillonnant uniformément l'espace des paramètres sphériques, un histogramme de taille $N_s * N_\theta * N_\varphi$ est défini pour stocker le descripteur de Hough 3D. La composante $h(s_{jk}, \theta_j, \varphi_k)$ est incrémentée de l'aire

des projections de chacune des facettes sur le plan d'orientation (θ_j, φ_k) passant par le centre de gravité des facettes et à une distance s_{jk} du centre de l'objet.

Étant donné que la transformée de Hough est étroitement liée au repère canonique dans lequel elle est calculée, les auteurs considèrent les 48 configurations possibles de ce repère afin de s'affranchir des problèmes d'inversion d'axes et de réflexions. Cependant, afin de réduire la taille du descripteur ils ne considèrent que 3 configurations en exploitant les relations mathématiques liant l'ensemble des configurations.

Transformée radiale angulaire : Cette description de la forme d'un objet 3D introduite par Ricard *et al.* [104][103][102] est basée sur la notion de transformation unitaire orthogonale complexe définie sur la boule unité. Celle-ci est une transformée radiale angulaire (ART) définie par trois fonctions séparables : une fonction radiale $R_n(\rho)$ et deux fonctions angulaires $A_{m_\theta}(\theta)$ et $A_{m_\phi}(\phi)$. Les cœfficients ART 3D sont calculés, sur l'objet normalisé au moyen d'une ACP, comme suit :

$$F_{nm_\theta m_\phi} = \int_0^{2\pi} \int_0^{\pi} \int_0^{1} A_{m_\theta}(\theta) * A_{m_\phi}(\phi) * R_n(\rho) f(\rho, \theta, \phi) \rho d\rho d\theta d\phi \tag{3.25}$$

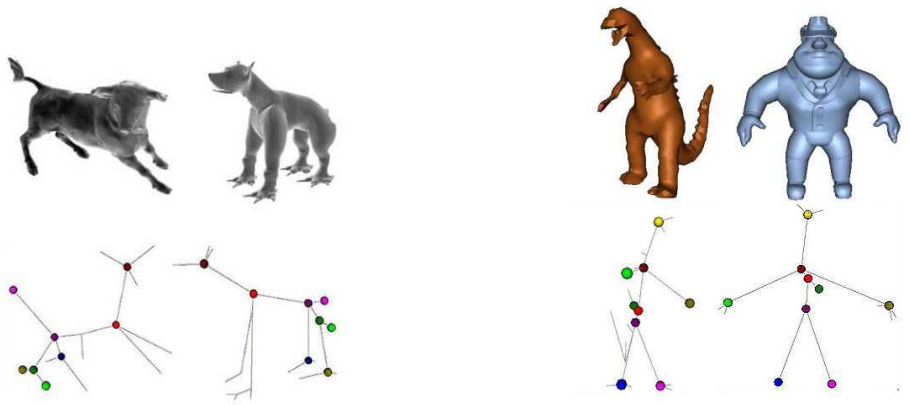
où f est la fonction sphérique représentant la surface de l'objet. Afin de garantir un bon rapport performance/vitesse, les auteurs calculent la transformée d'ordre $n = 3,\ m_\theta = 5$ et $m_\phi = 5$.

Les expérimentations menées sur les bases de "Princeton" [110] et de "Renault" [109] montrent de très bons résultats en particulier pour la seconde avec tout de même une petite supériorité des harmoniques sphériques.

3.4.5 Méthodes par squelettes

Amincissement de volume : Dans un esprit plus proche de la topologie de l'objet Sundar *et al.* [115] ont cherché à décrire la forme d'un modèle 3D par son squelette. Afin de l'obtenir, ils voxelisent celui-ci et amincissent cette représentation par une transformée de distance [42]. Les attributs d'amincissement permettent de réduire le volume en voxels par couches successives. Pour créer le squelette final de l'objet, les auteurs convertissent les points obtenus en un graphe acyclique en appliquant l'algorithme de l'arbre couvrant minimum. Finalement, pour comparer les squelettes de deux modèles 3D, et afin d'éviter un calcul trop coûteux, un algorithme basé sur la recherche de la mise en correspondance de coût minimale dans un graphe biparti est mis en place (voir figure 3.17).

L'utilité de ce type d'approche réside principalement dans la comparaison d'objets dit "articulés". En effet, dans ce cas les squelettes obtenus pour les deux objets sont très proches et l'appariement fournit des résultats robustes. Il est aussi intéressant de noter qu'avec cette méthode il est possible de faire des recherches partielles (voir section 3.7.2).



(a) *Animaux à quatre pattes* (b) *Animal debout et humain*

FIGURE 3.17 – *Exemples de mise en correspondance. (a) pour deux animaux à quatre pattes et (b) pour un animal et un humain. Seul les nœuds mis en correspondance sont montrés*

Graphe de Reeb : Une autre approche basée sur les squelettes consiste à créer le graphe de Reeb associé à un objet. Sa création repose sur la théorie de Morse [112] qui, à partir d'une fonction μ continue définie sur une surface non dégénérée, caractérise la topologie de celle-ci en ses points critiques. Le graphe s'obtient à partir d'une segmentation de la surface de l'objet obtenue selon les valeurs de la fonction μ. Les nœuds du graphe de Reeb représentent les composantes connexes d'une même partition, alors que les arrêtes permettent de relier les nœuds adjacents, selon la fonction μ. D'après cette définition, il est clair qu'il existe un lien étroit entre la fonction μ et le squelette obtenu. Un intérêt particulier doit donc être porté quant à la définition de celle-ci.

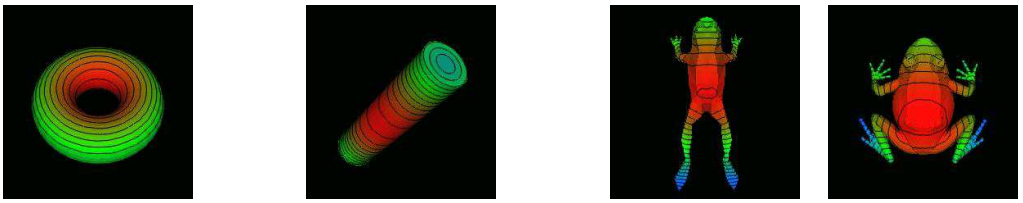
Dans leur approche Hilaga *et al.* [49] choisissent de définir la fonction μ à partir de la distance géodésique g entre deux points. La fonction μ d'un point v est définie par :

$$\mu(v) = \int_{p \in \mathcal{M}} g(v,p) d\mathcal{M} \tag{3.26}$$

où $\mathcal{M}$ représente le maillage de l'objet 3D. Grâce à cette fonction, le descripteur est invariant en rotation et aux petites perturbations du maillage (voir figure 3.18).

Dans le but d'accélérer la recherche de correspondances dans de grandes bases de données, les auteurs introduisent un descripteur qui s'appuie sur les graphes de Reeb multi-résolution (MRG) (voir figure 3.19).

Dans le même esprit de caractériser la forme d'un objet 3D à l'aide d'un squelette Tierny *et al.* [123][124][122] proposent une nouvelle fonction μ. Celle-ci est toujours basée sur la distance géodésique, mais calculée sur un ensemble de points d'intérêts extraient au préalable.



(a) *Un tore* (b) *Un cylindre* (c) *Une grenouille déformée* (d) *Une grenouille déformée*

FIGURE 3.18 – *Distribution de la fonction μ, basée sur la distance géodésique, pour un tore (a), un cylindre (b) et deux grenouilles déformées (c)(d).*

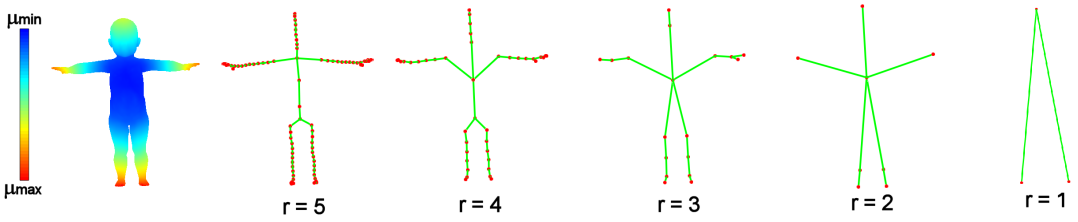
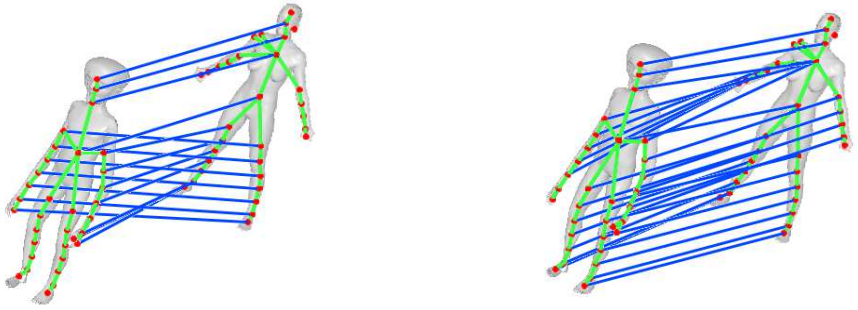


FIGURE 3.19 – *Graphe de Reeb multi-résolution d'un modèle humain aux résolutions $r = 1$ à 5 obtenu avec la fonction géodésique.*

Afin d'améliorer cette approche par squelette plusieurs améliorations ont été apportées par Tung et Schmitt [126][125] et Biasotti *et al.* [13] qui proposent d'enrichir les nœuds du graphe de Reeb d'attributs supplémentaires. Il s'agit par exemple d'attributs topologiques ou géométriques. Grâce à l'ajout de ces nouvelles informations, la mise en correspondance de graphe est nettement améliorée et permet d'obtenir de très bons résultats (voir figure 3.20).



(a) *Résultats sans information géométrique* (b) *Résultats avec information géométrique*

FIGURE 3.20 – *Comparaisons des résultats sans l'information géométrique (a), les jambes peuvent être appariées aux bras car ils sont topologiquement équivalents et en ajoutant l'information géométrique (b), tous les membres sont bien appariés.*

Finalement, les approches par construction de graphe de Reeb offrent des résultats très pertinents pour les bases de données d'objets dits "articulés". Les comparaisons faites lors du concours annuel *SHREC* avec d'autres approches placent les méthodes par graphe de Reeb à de très bonnes places (voir annexe D).

3.5 Etat de l'art sur les méthodes d'indexation "2.5D/3D"

Dans cette section, nous ne nous intéressons plus aux approches dites "3D/3D" mais aux méthodes qui projettent l'objet en trois-dimensions sur un ensemble de plans en deux dimensions. Cette famille de descripteur est appelée "2.5D/3D" car ils gardent une information tridimensionnelle sur l'objet au moyen de cartes de profondeurs par exemple. Il est en général difficile de construire une telle description sans le modèle 3D, ce qui oblige à avoir comme requête un objet de ce type. Cet état de l'art non exhaustif est organisé de la manière suivante :

1. Méthodes par cartes de courbures (3.5.1).
2. Méthodes par coupes (3.5.2).
3. Méthodes par images de profondeurs (3.5.3).

3.5.1 Méthodes par cartes de courbures

Afin de caractériser la forme de l'objet 3D, Assfalg *et al.* [9] construisent une carte de courbures de sa surface. Après une première étape de prétraitement cherchant à lisser l'objet et à réduire la complexité de son maillage, les auteurs expriment la courbure γ de chaque sommet v_i du maillage. Cette mesure est obtenue par la formule :

$$\gamma_{v_i} = \frac{1}{2} \frac{\sum_{v_j \in V^{v_i}} \|v_i^{\perp} - v_j^{\perp}\|}{|V^{v_i}|} \tag{3.27}$$

où $v_i^{\perp}$ représente la normale calculée au sommet v_i et V^{v_i} les sommets dans le voisinage de v_i. La carte de courbures en deux dimensions est obtenue en projetant, la mesure en chaque sommet, sur un cylindre. Finalement, les auteurs conservent comme descripteur la partition en région de cette carte avec l'aire et la courbure moyenne pour chacune d'entre elles.

Cette approche donne des résultats satisfaisants, mais nécessite une normalisation de l'objet pour garantir une invariance aux transformations géométriques. Afin de prendre en compte des déformations plus importantes sur la surface, il serait intéressant de mettre en place une comparaison élastique des cartes de courbures.

3.5.2 Méthodes par coupes

Jiantao *et al.* [56] proposent de décrire un objet 3D par une série de coupes de celui-ci le long de ses trois axes principaux. Afin de garantir une invariance en translation, rotation et mise à l'échelle, les auteurs sélectionnent parmi l'analyse en composantes principales

sur les sommets et sur les normales, celle qui minimise la boite englobante. Cela permet d'obtenir dans la plupart des cas un alignement visuellement correct pour un humain. Le modèle 3D est alors représenté par N coupes elles-mêmes décrites par la distribution de la distance entre deux points pris au hasard (voir figure 3.21).

(a) *Trois objets 3D* (b) *Coupes des trois objets 3D*

FIGURE 3.21 – *Trois objets 3D (a) et leurs coupes (b) le long du premier axe.*

L'un des intérêts de cette approche est son indépendance au maillage de l'objet 3D permettant de traiter des cas de surfaces dégénérées. Un travail complémentaire sur cette approche permettrait une recherche partielle de forme 3.7.2.

3.5.3 Méthodes par images de profondeurs

Images de profondeurs : Dans sa thèse Vranic [134] a développé un nouveau descripteur "DBD", pour *Depth Buffer Descriptor*, basé sur les images de profondeurs. Afin de définir une pose canonique pour l'ensemble des objets 3D, l'auteur introduit une amélioration de l'analyse en composantes principales qu'il appelle "Continue" (voir section 3.2.2). Afin de décrire la surface du modèle, celui-ci est projeté sur les six faces du cube englobant où la profondeur des images obtenues correspond à la profondeur de l'objet. Le descripteur correspond finalement aux cœfficients basses fréquences de la transformée de Fourier rapide 2D de chacune des six images de profondeurs (voir figure 3.22).

Évaluée sur la base du *Princeton Shape Benchmark* [110][19] cette méthode montre de très bons résultats. Son point faible peut venir de sa nature "2.5D/2D" qui nécessite souvent des requêtes en 3D. En effet, l'obtention d'information de profondeur à partir de photos reste délicate même avec une approche stéréoscopique.

Images de profondeurs améliorées : Passalis *et al.* [96] améliorent l'approche originale par images de profondeurs à la fois sur le plan de l'alignement que sur celui de la description réelle de l'objet 3D. Dans leur méthode, les auteurs capturent toujours un ensemble de six images de profondeurs issues des faces d'un cube. Les améliorations proposées portent sur deux aspects qui sont :

- La correction de la pose du modèle en analysant les informations de symétrie, qui apparaissent en soustrayant les images de profondeurs de deux faces opposées.
- Le calcul du descripteur sur la somme et la différence des images, associées aux faces opposées, afin d'obtenir une invariance en translation dans la direction des axes de

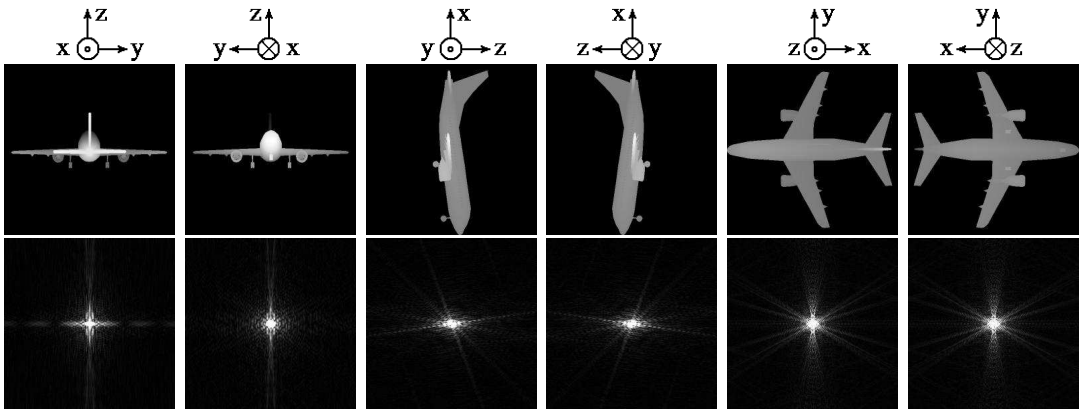


FIGURE 3.22 – *Extraction du descripteur DBD basé sur les images de profondeurs. La première ligne correspond aux six images obtenues par projection de l'objet sur les faces de la boite englobante. La seconde ligne correspond, quant à elle, à la transformée de Fourier rapide 2D de ces images.*

projections.

Comparée à l'approche originale, cette nouvelle méthode qui est une amélioration de la première accroît sensiblement la pertinence des résultats.

Elagage et alignement itératif : Dans cette nouvelle version de l'approche par images de profondeurs, Vajramushti *et al.* [128] proposent une technique visant à améliorer la pertinence des résultats. Les auteurs introduisent une mesure de proximité, entre objets, au moyen d'une différence de leur volume. Si celle-ci est inférieure à un seuil ϵ alors une étude plus approfondie est menée afin de donner une mesure de dissimilarité entre les deux modèles. Dans celle-ci, et pour palier les erreurs de l'analyse en composantes principales, un algorithme itératif d'alignement est appliqué aux objets de la base en appliquant de petites déformations et en calculant la dissimilarité de six images de profondeurs.

L'un des principaux reproches que l'on peut faire à cette approche de description d'objets 3D, outre son temps de calcul coûteux, est l'absence de réelle comparaison par rapport à la méthode originale.

Descriptions multi-vues : Afin d'enrichir la description faite par Vranic [134], Ohbuchi *et al.* [87] proposent non plus d'extraire six images de profondeurs, mais 42. Les points de vues associés à ces images sont uniformément répartis sur la sphère unité et correspondent aux 12 sommets et aux 30 milieux des arrêtes d'un icosaèdre. Les images de profondeurs obtenues sont finalement transformées du système de coordonnées cartésiennes en coordonnées polaires Zhang et Lu [146], avant de garder les cœfficients basses

fréquences de leur transformée de Fourier rapide 2D. Finalement, une invariance en rotation est obtenue par l'utilisation du système de coordonnées polaires associé à la mesure de dissimilarité $d(\mathcal{O}_P, \mathcal{O}_Q)$:

$$d(\mathcal{O}_P, \mathcal{O}_Q) = \frac{1}{42} \sum_{i=1}^{42} \left(\min_{1 \leq k \leq N} \left(\sum_{j=1}^{42} |f_{ij}^{\mathcal{O}_P} - f_{kj}^{\mathcal{O}_Q}| \right) \right) \tag{3.28}$$

avec N qui représente la taille du vecteur caractéristique et $f_{ij}^{\mathcal{O}_P}$, $f_{kj}^{\mathcal{O}_Q}$ les signatures des objets $\mathcal{O}_P$ et $\mathcal{O}_Q$. Les auteurs ([88]) améliorent leur approche en décrivant chaque image de profondeurs à l'aide des "SIFT" (*Scale-Invariant Feature Transform*) proposés par Lowe [68]. Grâce à la construction d'un alphabet de taille N sur les points d'intérêts, par un algorithme de *k-means* non supervisé, les auteurs réduisent la complexité du descripteur en codant dans un histogramme de taille N la fréquence de chaque mot de l'alphabet. Finalement, les histogrammes sont comparés en utilisant la divergence de Kullback-Leibler.

Évaluée sur la base du *Princeton Shape Benchmark* [110] la méthode proposée donne des résultats équivalents au "LFD" (voir section 3.6.1) classé à la première place.

Indices de pertinence : Afin d'améliorer la pertinence, des approches multi-vues en général, mais plus particulièrement celles qui caractérisent l'information de forme avec des images de profondeurs Chaouch et Verroust-Blondet [25][26][24] proposent, dans les méthodes qu'ils nomment "ESA" et "EDBA", de donner un poids à chaque vue. Les auteurs proposent trois indices de pertinence (voir figure 3.23) :

1. Indice de surface : Il caractérise la surface de la projection de l'objet 3D sur le plan. Il est donné par le nombre de pixels ayant des valeurs positives.
2. Indice de profondeur : C'est la somme de tous les pixels non nuls sur l'image. En d'autres termes, il correspond au volume compris entre les parties visibles de l'objet 3D et la face de projection.
3. Indice global : Il est égal à la somme des distances des points de l'objet visibles sur l'image de profondeur avec le centre de la boite englobante.

Grâce à cette pondération des différentes vues, les auteurs améliorent nettement l'approche originale et introduisent un concept applicable à toutes méthodes utilisant plusieurs vues pour la description d'un modèle 3D.

Lignes de profondeur : L'une des principales faiblesses des techniques utilisant les images de profondeurs est leur difficulté à prendre en compte la déformation des maillages. Pour accroitre la sensibilité de ce type de méthode Chaouch [24] propose de décrire chaque image avec une séquence d'états. Chaque ligne de l'image de profondeurs est alors codée avec un alphabet réduit, caractérisant la pente de celle-ci (voir figure 3.24).

Grâce à une augmentation du nombre de vues (les vingt sommets d'un dodécaèdre) et à une mise en correspondance élastique, mise en place au moyen d'une "programmation

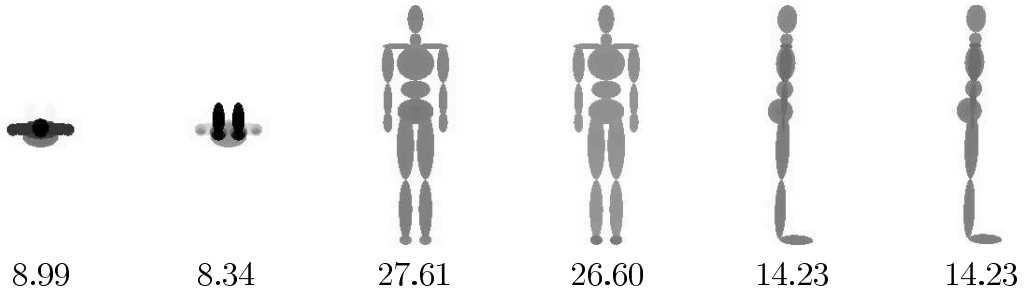
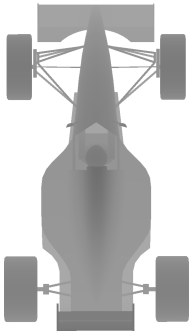
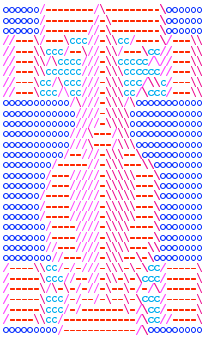


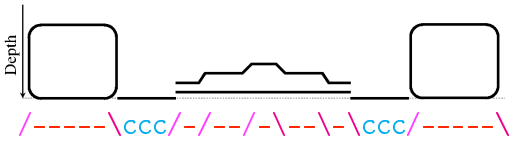
FIGURE 3.23 – *Exemple d'indices de pertinence globaux (%) pour les images de profondeurs extraites à partir d'un objet 3D représentant un humain.*



(a) *Image de profondeur*



(b) *Lignes de profondeurs*



(c) *Détail d'une ligne de profondeurs*

FIGURE 3.24 – *Image de profondeur (a), séquences d'états associées aux lignes de profondeurs (b) et transcription d'une ligne de profondeurs (c).*

dynamique", l'auteur améliore très fortement les résultats du descripteur original sur la base de données du *Princeton Shape Benchmark* [110]. L'auteur nomme ses différentes approches : "CDLA", "DLA" et "MDLA" (pour Depth Line Approach).

3.6 Etat de l'art sur les méthodes d'indexation 2D/3D

Dans cette dernière classe d'approches, nous nous intéressons aux approches "2D/3D" qui caractérisent la forme d'un modèle à partir d'informations en deux dimensions. Elles se basent toujours sur des projections de l'objet sur un ensemble de plans en deux dimensions, mais contrairement aux approches précédentes, celles-ci ne gardent aucune information tri-

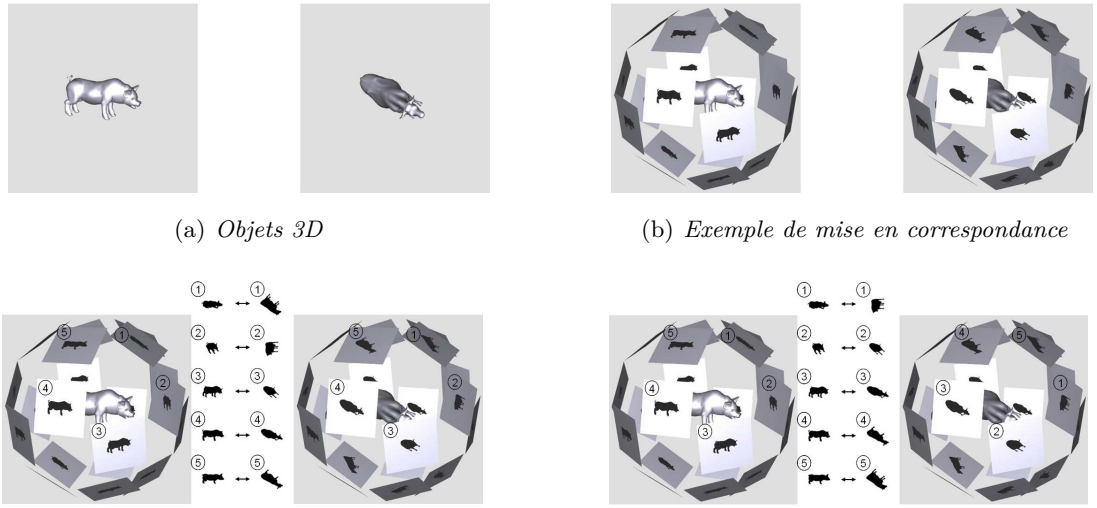
dimensionnelle. Les descripteurs sont le plus souvent basés sur la géométrie des silhouettes obtenues après la projection et s'appuient sur des approches multi-vues pour accroitre la pertinence des résultats. Nous présentons dans la suite de cette section un état de l'art non exhaustif de ces approches classées comme suit :

1. Méthodes basées région (3.6.1).
2. Méthodes basées contour (3.6.2).

3.6.1 Méthodes basées région

Moments de Zernike 2D multi-vues : Chen *et al.* [29][139][28] proposent une méthode originale nommée "LFD", pour *LightField Descriptor*, permettant d'indexer un objet 3D. Ils proposent pour chaque silhouette 2D, obtenue par projection du modèle 3D sur un plan, de calculer ses 36 moments de Zernike [145] ainsi que 10 cœfficients de Fourier. Pour caractériser au mieux l'objet 3D, dix vues sont extraites à partir des sommets d'un dodécaèdre. Il n'est pas nécessaire de capturer une silhouette à partir des vingt sommets car, en utilisant une projection orthogonale, les silhouettes des sommets opposées sont identiques à un miroir près, pris en compte par la description.

Le point novateur de cette approche est l'utilisation de la notion de "Lightfield" pour s'affranchir de l'analyse en composantes principales tout en garantissant une invariance en rotation. Les "Lightfields" sont un réagencement des silhouettes capturées à partir des sommets du dodécaèdre. Cette méthode permet de simuler une rotation de l'objet 3D en permutant uniquement les vues extraites (voir figure 3.25).



(a) *Objets 3D*

(b) *Exemple de mise en correspondance*

(c) *Exemple de mise en correspondance*

(d) *Exemple de mise en correspondance*

FIGURE 3.25 – *Comparaisons des "Lightfields" associés à un cochon et à une vache (a), afin d'obtenir une invariance en rotation, les jeux de vues sont permutés et la meilleur mise en correspondance est choisie (b)(c)(d).*

Sachant qu'il y a 60 permutations "R" possibles pour le dodécaèdre (20 (*sommets*) $*$ (3 *orientations*) $= 60$) et que les auteurs génèrent en plus 8 "Lightfields" différents "L"

(obtenus par de petites rotations d'un original), la distance entre deux objets est calculée comme suit :

$$d(\mathcal{O}_P, \mathcal{O}_Q) = \min_{1 \leq i \leq L} d_L(L_1^{\mathcal{O}_P}, L_i^{\mathcal{O}_Q}) \tag{3.29}$$

avec L qui représente un "Lightfield" et où d_L est la mesure de dissimilarité entre deux "Lightfields" définie par :

$$d_L(L_1^{\mathcal{O}_P}, L_i^{\mathcal{O}_Q}) = \min_{1 \leq j \leq R} \left(\sum_{k=1}^{V} d_R(R_{1,k}^{\mathcal{O}_P}, R_{i,j(k)}^{\mathcal{O}_Q}) \right) \tag{3.30}$$

avec R qui représente une silhouette, V leur nombre et où d_R est la mesure de dissimilarité entre deux vues qui vaut :

$$d_R(R_{1,k}^{\mathcal{O}_P}, R_{i,j(k)}^{\mathcal{O}_Q}) = \sum_{l=1}^{Z+F} d_{ZF}(ZF_{1,k,l}^{\mathcal{O}_P}, ZF_{i,j(k),l}^{\mathcal{O}_Q}) \tag{3.31}$$

avec ZF qui représente les descriptions à base de moments de Zernike et de cœfficients de Fourier et où d_{ZF} est la distance entre deux descripteurs codés par les moments de Zernike et les cœfficients de Fourier. Le coût de cette mesure est donc :

$$C = R * L * Z * F * V = 8 * 60 * 35 * 10 * 10 = 1680000 \text{ calcul de distance } L_1 \tag{3.32}$$

avec $Z = 35$ le nombre de moments de Zernike, $F = 10$ le nombre de cœfficients de Fourier et $V = 10$ le nombre de vues. La complexité de la comparaison étant très importante, les auteurs mettent en place une comparaison hiérarchique en accroissant progressivement le nombre de moments de Zernike, de cœfficients de Fourier et de "Lightfields".

Lors de la comparaison faite par le *Princeton Shape Benchmark* [110] la méthode "LFD" a obtenu les meilleurs résultats. L'un des intérêts majeurs de cette méthode, outre sa robustesse, est sa capacité à traiter des requêtes multimodales telles que par exemple : des images segmentées, des dessins au trait ou plus simplement des objets 3D.

Sélection de vues et approche Bayesienne : Dans le but d'améliorer cette méthode Filali-Ansary *et al.* [35][7] proposent un algorithme de classification adaptatif "AVC" permettant de sélectionner les vues caractéristiques d'un objet 3D [36][8]. En association avec un modèle statistique, les auteurs sont capables de choisir parmi 320 vues un maximum de 40 silhouettes caractérisant au mieux la forme de l'objet 3D. Après une description de chacune d'entre elles, au moyen des moments de Zernike 2D, la mise en correspondance est effectuée avec une approche Bayesienne permettant de pondérer l'apport de chaque silhouette dans la comparaison.

Par rapport à la méthode originale ("LFD", voir section 3.6.1), cette nouvelle approche se place juste derrière. Cette baisse de performance vient probablement de l'utilisation des cœfficients de Fourier pour les "LFD" et d'une mise en correspondance plus robuste, mais coûteuse.

3.6.2 Méthodes basées contour

Contour et indices de pertinences : Dans cette approche basée sur les contours des silhouettes, Chaouch [24] propose de caractériser la forme d'un objet 3D en étudiant la position des points qui compose le contour. Afin de normaliser la longueur du contour, l'auteur effectue un échantillonnage angulaire en sélectionnant pour chaque secteur les N points les plus éloignés. Le descripteur est alors composé des cœfficients basses fréquences de la transformée de Fourier unidimensionnelle pour les N points, en coordonnées polaires, des trois vues correspondant aux directions des axes principaux.

La simplicité de cette approche la rend peu robuste et offre donc des résultats peu intéressants. Elle peut cependant servir, grâce à sa rapidité, à une phase d'élagage des objets peu pertinents.

3.7 Etat de l'art sur les autres méthodes d'indexation 3D

En plus des différents types d'approches énumérés ci-dessus, d'autres travaux ont vu le jour dans des domaines très proches du notre. Par exemple, les travaux cherchant à combiner plusieurs descripteurs, qu'ils soit "3D/3D", "2.5D/3D" ou "2D/3D", afin d'améliorer la description de forme d'un objet 3D. Il existe aussi des méthodes qui permettent d'interroger la base avec des requêtes partielles, offrant, de ce fait, une grande flexibilité de recherche à l'utilisateur. Elles seront brièvement introduites dans l'ordre suivant :

1. Méthodes par fusion de descripteurs.
2. Méthodes de recherche partielle.

3.7.1 Méthodes par fusion de descripteurs

Daras et Axenopoulos [33] proposent de fusionner trois descripteurs 2D (cœfficients de Fourier, moments de Zernike et de Krawtchouk), calculés sur les images de profondeurs et sur les silhouettes, afin d'améliorer les performances. Associée à une extraction originale basée sur 18 vues non redondantes, cette méthode offre de très bonnes performances. Cette approche est d'autant plus intéressante que les auteurs proposent un processus de recherche multimodale permettant de faire des requêtes à partir de vues 2D ou d'un modèle 3D.

Dans le même esprit, Papadakis *et al.* [92] utilisent un descripteur de forme hybride qui combine les images de profondeurs et les harmoniques sphériques. Cette association permet de caractériser la surface de l'objet à la fois en 2D et en 3D. Cette approche allie donc les forces de ces deux types d'approches et semble être très robuste sur la plupart des bases de données classiques.

Bustos *et al.* [17][20] proposent aussi une méthode basée sur la fusion de descripteurs. Celle-ci extrait une série de signatures (harmoniques sphériques, images de profondeurs...)

et pondère l'apport de chacune afin de tirer avantages de chaque description.

Finalement, Akgul *et al.* [5][4] fusionnent plusieurs informations géométriques locales afin d'obtenir une description pertinente de la surface d'un objet 3D. Grâce à une densité de probabilité, les auteurs construisent un descripteur puissant facile à comparer.

3.7.2 Méthodes de recherche partielle

Funkhouser *et al.* [40] proposent de segmenter l'objet 3D afin de pouvoir faire des recherches partielles sur celui-ci. Pour partitionner le modèle, les auteurs ont développé un outil permettant de dessiner le chemin selon lequel la coupure va se faire. Afin d'aider l'utilisateur, le chemin cherche à suivre les "coutures" naturelles de l'objet, rendant ainsi la segmentation plus réelle. Chaque partie de l'objet peut alors être décrite au moyen d'un descripteur 3D.

Toujours dans le but de faire de la recherche à partir d'un échantillon de l'objet 3D, les mêmes auteurs proposent de caractériser un ensemble de points [121]. Après avoir sélectionné aléatoirement les points, les auteurs définissent une région à l'aide d'une sphère centrée sur chacun d'entre eux. Afin d'avoir plusieurs niveaux de descriptions, la sphère est construite avec des rayons $r \in \{0.25,\ 0.5,\ 1.0,\ 2.0\}$. Pour finir, la description de chaque région est obtenue en calculant des harmoniques sphériques (voir section 3.4.4).

Même si cette dernière méthode offre des résultats très robustes sur la base du *Princeton Shape Benchmark* [110], le temps d'extraction et de calcul est trop important pour permettre une recherche en ligne. De plus, le descripteur est trop peu synthétique pour être utilisé sur de grandes bases de données.

Gal et Cohen-Or [43] décrivent un objet 3D à l'aide d'une signature caractérisant un ensemble de points clés. Ceux-ci sont sélectionnés par rapport à l'information qu'ils véhiculent : la courbure moyenne du voisinage, la variance de la courbure dans le voisinage... L'information contenue par chaque région est alors codée au moyen d'un descripteur local (voir section 3.4.2). La mise en correspondance partielle d'objet est accélérée grâce à la mise en place d'un index permettant de retrouver rapidement les régions similaires.

Cette nouvelle approche présente les mêmes inconvénients que les méthodes de recherche partielle présentées précédemment à savoir : le temps d'extraction et de calcul pour une requête ainsi que la taille importante du descripteur en mémoire.

Une approche intéressante introduite par Biasotti *et al.* [14] permet de caractériser la similarité entre deux graphes 3D comme le plus grand sous graphe commun entre les deux modèles. Deux objets différents n'auront que peu de nœuds communs tandis que deux modèles 3D proches auront la quasi-totalité de leur graphe en commun. Grâce à cette définition qui utilise les graphes de Reeb (voir section 3.4.5) pour sa mise en place, les auteurs sont capables de faire des requêtes partielles sur une base de données 3D.

Cette approche est très utile pour rechercher des portions d'un objet 3D de manière simple et rapide. En revanche, la pertinence des approches à base de graphe dépend entièrement de la construction de celui-ci qui reste parfois difficile.

Dans le même esprit, Sundar *et al.* [115] utilisent leur graphe obtenu par amincissement de volume (voir section 3.4.5) pour faire de la recherche partielle. En effet, une fois le squelette de l'objet 3D construit, la recherche d'un sous graphe devient possible et offre une certaine flexibilité à la méthode pour les recherches dans les bases de données volumineuses.

3.8 Conclusion

L'état de l'art que nous avons présenté donne un aperçu des méthodes de normalisations existantes, mais aussi des différents descripteurs développés pour extraire l'information de forme des modèles 3D. Malgré le caractère non exhaustif de cette présentation, nous pouvons nous rendre compte que la littérature scientifique est déjà très féconde sur le sujet et offre une variété d'approches très différentes.

Étant donné que les méthodes de normalisation, permettant de rendre l'objet 3D invariant aux transformations de l'espace, sont pertinentes, nous avons décidé dans la suite de ce manuscrit de présenter un critère permettant de choisir une pose adéquate, parmi un ensemble d'alignement du modèle 3D. Ainsi, il sera possible non pas de calculer une nouvelle pose, mais de déterminer dans un sous-ensemble, laquelle est la plus appropriée pour l'extraction d'un descripteur.

D'autre part, après l'état de l'art fait sur les méthodes de description de la forme d'un modèle 3D, il nous a semblé intéressant d'approfondir les méthodes "2D/3D" et "2.5D/3D" car elles offrent une grande flexibilité. En effet, ces méthodes sont généralement plus robustes que les méthodes "3D/3D" grâce à une sélection de l'information caractéristique au moyen de projections. On peut d'ailleurs supposer que la faiblesse des méthodes "3D/3D" est leurs difficultés à extraire les caractéristiques intrinsèques de l'objet 3D, peut-être car celles-ci sont noyées dans modèle 3D complet. En plus de cette robustesse, ces approches permettent de faire des requêtes à partir de diverses sources, à savoir : un modèle 3D, un ensemble de photos ou de dessins. C'est pour ces différentes raisons que nous présentons, dans la suite de ce manuscrit, des méthodes permettant de décrire de manière pertinente la forme d'un objet 3D ainsi que des méthodes de mise en correspondance adaptées offrant les qualités adéquates pour la recherche d'objets dits "classiques" ou "articulés" (voir annexe D).

CHAPITRE 4

Normalisation et estimation de l'alignement

Résumé *: Ce chapitre introduit notre première contribution portant sur la normalisation et l'estimation de la pose des objets 3D. La première passe par un calcul de la sphère minimale englobant le modèle 3D tandis que la seconde est basée sur un critère de minimisation de l'enveloppe visuelle.*

4.1 Introduction

Dans le cadre de l'indexation d'objets 3D, une étape de prétraitement du modèle, telle que celle présentée dans la section 3.2 du chapitre 3, est souvent nécessaire. En effet, les modèles que nous souhaitons indexer sont généralement fournis avec, un centre, une échelle et une orientation quelconque, car dépendant de l'outil d'acquisition ou de numérisation avec lequel ils ont été créés (voir section 2.3.1 du chapitre 2). De ce fait, certaines méthodes, nommées : "méthodes non invariantes aux transformations de l'espace", nécessitent que les objets 3D soient donnés dans une configuration précise, identique pour tous les modèles de la base de données. Cette transformation s'obtient généralement en plaçant chaque objet 3D à l'intérieur d'une enveloppe commune, appelée "volume englobant". Grâce à cette étape, il devient possible de calculer une description pertinente et synthétique ouvrant la voie à une recherche efficace à travers le processus d'indexation.

Les descriptions que nous introduirons dans le chapitre 5 n'étant pas invariantes à toutes les transformations de l'espace, nous avons ajouté une étape de prétraitement permettant la normalisation de nos objets 3D. Afin de traiter efficacement cette tâche, nous nous sommes tournés vers une nouvelle méthode adaptée à notre procédé d'indexation, à savoir les approches "2.5D/3D" et "2D/3D". En effet, malgré la variété des traitements présentés dans la section 3.2 du chapitre 3, aucun ne permet de répondre efficacement à nos contraintes, basées sur la qualité du contour d'un modèle 3D lorsqu'il est projeté sur un plan en deux dimensions. Les travaux de ce chapitre ont fait l'objet d'une publication dans *EURASIP Journal on Image and Video Processing* [76].

Nous présentons dans ce chapitre deux nouvelles méthodes nommées "Normalisation" et "Estimation de l'alignement". La première, que nous introduisons dans la section 4.2, propose une manière simple, rapide, mais aussi efficace de trouver le centre et l'échelle d'un modèle 3D. La seconde, que nous proposons dans la section 4.3, permet, quant à elle, de définir l'orientation d'un modèle dans un espace en trois dimensions. Notons que l'élaboration de ces deux nouvelles approches s'est faite en accord avec les besoins intrinsèques

de la méthode, présentée dans la suite de ce manuscrit au chapitre 5, et apportent donc des solutions pertinentes aux contraintes de notre processus d'indexation 3D.

4.2 Centrage et mise à l'échelle

Lorsque les objets 3D sont créés, manuellement par un infographiste, ou de manière automatique par un algorithme de reconstruction, un maillage est alors créé. Celui-ci, composé de triangles en trois dimensions, est défini dans un repère 3D, déterminant de ce fait le centre, l'échelle et l'orientation des modèles 3D. Cependant, rien n'assure que l'ensemble des objets soit modélisé dans une configuration identique. Ainsi, le centre et l'échelle de ceux-ci peuvent changer d'un modèle à l'autre, créant des différences, au niveau de leur position, lors de leur rendu dans un univers 3D. Les descripteurs du processus d'indexation présentés dans le chapitre 5 n'étant pas robustes envers ces transformations, il est indispensable que cette étape permette de traiter chaque objet 3D en vue de les représenter de façon similaire dans le repère 3D (voir figure 4.1).

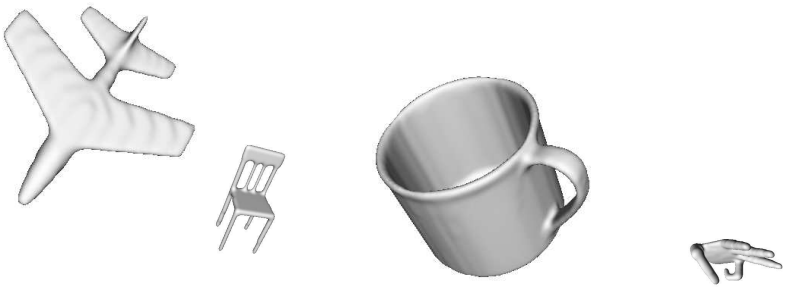
Dans notre approche d'indexation, nous nous intéressons principalement aux contours des différentes silhouettes du modèle 3D. Celles-ci sont obtenues par projection sur des plans 2D (voir figure 4.2) introduisant de ce fait des contraintes. En accord avec celles-ci, nous proposons dans cette section, la définition du centre et de l'échelle d'un modèle 3D. Le but de ce traitement est de placer tous les objets 3D dans une enveloppe commune. Cette normalisation permettra, à terme, de projeter l'intégralité d'un modèle 3D dans un espace 2D déterminé.

4.2.1 Recherche de la normalisation optimale

Lors de la numérisation ou de la création des objets 3D, le centre et l'échelle associés à ceux-ci peuvent varier, et sont donc incohérente sur l'ensemble d'une base de données que l'on souhaite indexer. Dans ce cadre, nous avons cherché à placer les modèles de façon identique, en terme de centre et d'échelle, dans le repère 3D. Ainsi, lors de leur rendu, ils apparaîtront comme étant tous à l'intérieur d'une enveloppe commune, dans notre cas, une sphère de rayon 1 centrée.

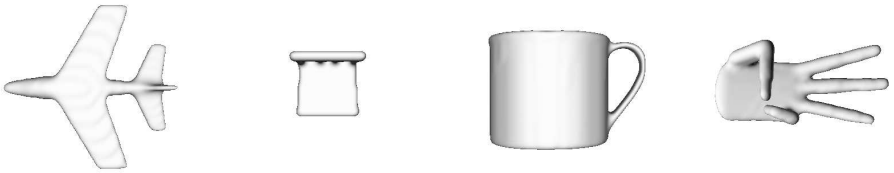
Les contraintes qui ont guidé le choix de cette normalisation viennent principalement des méthodes utilisées, a posteriori, pour l'indexation d'un objet 3D. Dans celles-ci, nous nous intéresserons aux contours des silhouettes du modèle obtenus par projection. De ce fait, deux contraintes s'imposent à nous :

- La première est l'obtention d'un contour de qualité "optimale", c'est à dire de périmètre maximum pour une résolution d'image, ou nombre de pixels, fixé. En effet, plus celui-ci est grand, plus l'information qu'il véhicule est importante réduisant ainsi le risque de perdre les caractéristiques de forme qu'il contient (voir figure 4.3).
- La seconde est l'assurance que la projection contienne l'intégralité de l'objet 3D,



(a) *Quatre objets 3D avant traitement*

(b) *Quatre objets 3D après normalisation*



(c) *Quatre objets 3D après estimation de l'alignement*

FIGURE 4.1 – *Exemple de quatre objets 3D avant traitement (a), après centrage et mise à l'échelle (b) et après estimation de l'alignement (c).*

ceci afin de ne pas rogner le contour. En effet, une occlusion artificielle nuirait à l'extraction des caractéristiques de forme du contour (voir figure 4.4).

Une approche classique pour résoudre les invariances en translation et en échelle est de placer chaque objet 3D dans un volume englobant commun. L'enveloppe couramment utilisée est la boite englobante (voir la section 3.2 du chapitre 3), centrée dans le repère 3D car simple et rapide à calculer. Cependant, même si celle-ci garantit la capture intégrale des silhouettes lorsque l'on projette l'objet sur les faces du cube, elle ne garantit aucunement une qualité "optimale" des contours. En effet, cette méthode ne cherche pas à maximiser la taille des silhouettes.

Dans le même esprit, nous nous sommes tournés vers un volume plus approprié à notre problème, à savoir la sphère minimale englobante. Tout comme pour la boite englobante, celle-ci garantit une bonne projection des silhouettes depuis n'importe quel point de vue. En effet, si le plan de projection se situe sur l'enveloppe de la sphère, alors l'objet sera

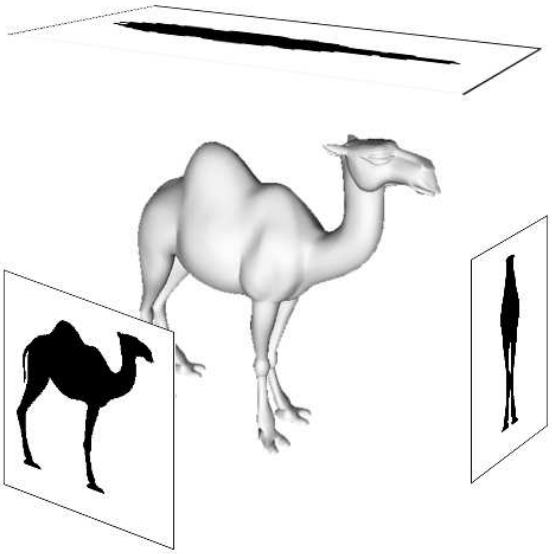
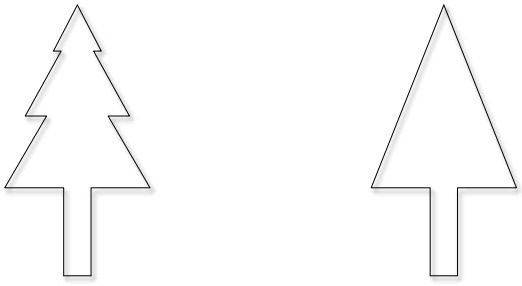


FIGURE 4.2 – *Schéma représentant la projection d'un objet 3D sur des plans 2D capturant, de ce fait, les différentes silhouettes du modèle.*

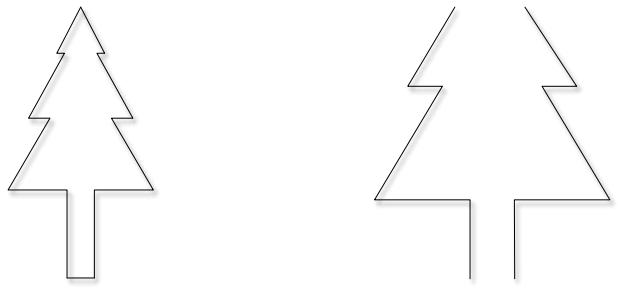


(a) *Contour bien échantillonné*

(b) *Contour sous échantillonné*

FIGURE 4.3 – *Différence entre deux contours de sapin, (a) représente un contour de bonne qualité tandis que (b) représente un contour moins précis à cause d'un sous échantillonnage. Nous nous rendons bien compte que dans le second cas, une information importante est perdue.*

visible dans son intégralité, à condition que son échelle le confine à cette enveloppe (voir aussi remarque 4.1). Mais l'intérêt majeur de cette sphère, par sa propriété de minimalité, est sa capacité à maximiser le périmètre du contour extrait. En effet, en fixant une taille de 2 au couple sphère-objet, chaque modèle aura une taille maximale à l'intérieur de la sphère unique de rayon $r = 1$. Finalement, grâce à l'utilisation de cette sphère minimale englobante, chaque objet 3D pourra être placé dans une enveloppe commune définissant ainsi un repère canonique (voir figure 4.5).

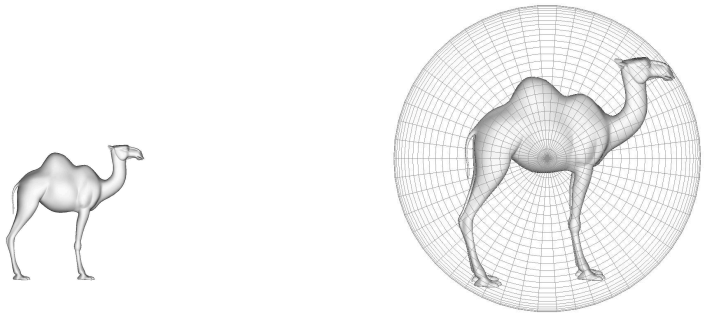


(a) *Contour complet*

(b) *Contour rogné*

FIGURE 4.4 – *Différence entre deux contours de sapin, (a) représente un contour dans son entier tandis que (b) représente un contour rogné. Nous nous rendons bien compte que dans le second cas, une normalisation non adaptée a détruit une partie de l'information contenue dans l'objet.*

> **Remarque 4.1.** *Grâce à la nature de la sphère englobante, tous les points de sa surface sont à égale distance de son centre. La capture de silhouettes depuis plusieurs points de vues, répartis uniformément autour de l'objet, est alors simple (on appellle aussi cette approche "multi-vues"). En effet, il suffit de placer un plan de projection, ou appareil photo "virtuel", n'importe où sur la sphère. Il faut cependant faire attention aux propriétés du plan, à savoir une normale dirigée vers le centre de la sphère et une focale permettant de capturer l'intégralité de la silhouette de l'objet.*



(a) *Objet initial*

(b) *Objet normalisé*

FIGURE 4.5 – *Exemple d'un objet, (a), normalisé par sa sphère minimale englobante (b). Le modèle 3D est représenté dans son ensemble et son échelle est maximale, par rapport à l'enveloppe commune.*

En accord avec l'enveloppe choisie, nous pouvons définir les paramètres des deux transformations : $\mathbf{T}_{x,y,z} = (t_x, t_y, t_z) \in \mathbb{R}^3$ pour la translation (le centre) et $\mathbf{S}_s = s \in \mathbb{R}_+$ pour le facteur d'échelle. Cela revient à rechercher les paramètres de la normalisation

$\theta = (s, t_x, t_y, t_z)$ permettant de placer un objet noté $\mathcal{O}$ dans la sphère unité notée $\mathcal{S}$ définie par un centre : $(c_x, c_y, c_z) \in \mathbb{R}^3$ et une échelle $r \in \mathbb{R}$. Dans notre cas, les paramètres sont définis comme suit : $c_x = 0, c_y = 0, c_z = 0$ *et* $r = 1$, c'est-à-dire un centre situé au centre du repère, et une enveloppe située à une distance de 1 par rapport au centre du repère.

Au moyen d'un déplacement et d'une mise à l'échelle, nous cherchons à placer de manière indépendante chaque objet 3D $\mathcal{O}$, dans la sphère unité $\mathcal{S}$. Grâce à ce confinement, dans la sphère minimale englobante de chaque modèle, nous maximisons le volume global du modèle 3D garantissant ainsi un périmètre maximal des contours projetés. De plus, cette sphère assure une projection intégrale de l'objet depuis n'importe quel point de vue. Finalement, cette sphère minimale englobante répond parfaitement aux contraintes nécessaires à notre processus d'indexation 3D définie dans le chapitre 5.

4.2.2 Sphère minimale englobante

Comme nous l'avons expliqué dans le paragraphe précédent, nous cherchons, pour chaque objet, sa sphère minimale englobante afin de le normaliser. Le problème de la recherche de celle-ci est bien connu, et remonte jusqu'en 1857 quand Sylvester [116] l'a formulé pour des points dans le plan. Depuis, de nombreux algorithmes de résolutions ont vus le jour afin de répondre de manière rapide (en temps linéaire) à ce problème complexe, voir en particulier les travaux de Megiddo [71] et de Welzl [138]. Finalement, Fischer *et al.* [38] ont proposé une nouvelle méthode très précise et efficace, permettant de résoudre le problème du calcul de la sphère minimale englobante en dimensions d quelconque. Dans ce cadre, nous avons choisi d'utiliser, pour le calcul de l'enveloppe permettant notre normalisation, l'algorithme de Gärtner [44].

Cet algorithme est basé sur une approche itérative qui cherche à réduire une sphère englobante initiale afin de la minimiser (voir annexe B). Elle permet la recherche de la sphère minimale englobante d'un ensemble de points $\mathcal{P} \in \mathbb{R}^d$ avec une complexité de $\mathcal{O}(nd)$ par itération où n est le nombre de points et d leur dimension. Finalement, grâce à la précision et à la rapidité de cet algorithme, nous avons un moyen efficace pour calculer les paramètres de notre normalisation et ainsi garantir un centre et une échelle "optimale" pour notre processus d'indexation.

4.2.3 Définition de la normalisation

Afin de normaliser chaque objet $\mathcal{O}$ de la base de données, nous cherchons à le placer dans une sphère unité $\mathcal{S}$ de rayon $r = 1$. Il s'agit donc de trouver une translation $\mathbf{T}_{x,y,z} = (t_x, t_y, t_z) \in \mathbb{R}^3$ permettant de centrer chaque modèle 3D dans la sphère, ainsi qu'une échelle $\mathbf{S}_s = s \in \mathbb{R}_+$ permettant de faire "entrer" les objets dans celle-ci. Pour trouver les différents paramètres $\Theta = (s, t_x, t_y, t_z)$ de la transformation $\mathbf{T}_{x,y,z} \circ \mathbf{S}_s$, nous utilisons l'algorithme de calcul de la sphère englobante énoncé précédemment en dimension $d = 3$

avec $\mathcal{P} = \mathcal{O}$. Ce qui nous donne :

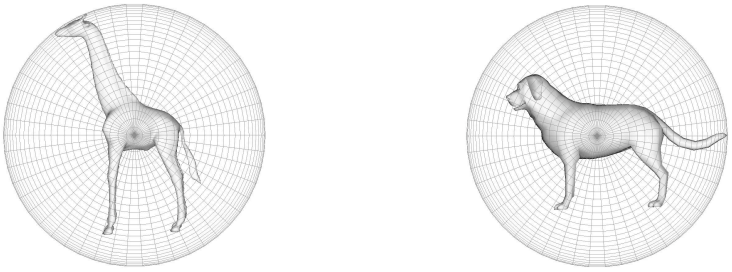
$$\Theta = (\frac{1}{r}, c_x, c_y, c_z) \tag{4.1}$$

Ainsi, nous pouvons déplacer et mettre à l'échelle chaque objet 3D de la base de données afin de le normaliser. Les coordonnées de chaque sommet de l'objet $\mathcal{O}$ deviennent alors :

$$x'_p = \frac{x_p - c_x}{r},\ y'_p = \frac{y_p - c_y}{r},\ z'_p = \frac{z_p - c_z}{r} \tag{4.2}$$

De ce fait, tous les modèles 3D de la base de données sont normalisés par un processus commun qui garantit d'une aire projetée maximale, et ceci depuis n'importe quel point de vue permettant ainsi l'extraction précise du contour des différentes silhouettes.

Une dernière justification de notre choix de normalisation doit être faite. En effet, dans celle-ci, nous nous rendons bien compte, à cause de l'utilisation de la sphère minimale englobante, que les objets ne sont pas normalisés en accord avec leurs tailles réelles. Nous entendons par là, que des objets de la même classe, des animaux à quatre pattes par exemple, tels qu'une girafe et un chien, seront normalisés avec des critères ne dépendant aucunement de leur sémantique, mais uniquement de la distribution de leurs sommets (voir figure 4.6). De ce fait, la girafe paraîtra "petite" face au chien, et ceci à cause de la longueur de son cou. Pour pallier à ce phénomène, gênant pour l'extraction d'une description pertinente de chaque objet, il serait nécessaire de prendre en compte ces informations sur la nature des objets dans la normalisation. Cependant, une telle mesure nuirait à l'extraction d'un contour de qualité optimale, contredisant ainsi nos contraintes définies dans la section 4.2.1. Nous avons donc choisi de clairement séparer la normalisation de l'objet, au moyen de la sphère minimale englobante, en vue d'extraire un contour précis, et l'éventuelle normalisation a posteriori du contour lors de la phase d'indexation à proprement parler. En effet, nous pourrons dans cette étape introduire des notions en prenant en compte l'aspect sémantique sur la taille des différents objets.



(a) *Girafe* (b) *Chien*

FIGURE 4.6 – *Exemple d'une girafe (a) et d'un chien (b) normalisés avec notre approche utilisant la sphère minimale englobante. Il est clair que cette méthode ne tient pas compte de la sémantique des objets car la taille des deux modèles 3D ne correspond pas à la réalité. En effet, la girafe est environ deux fois plus grande que le chien, ce qui ne reflète pas la réalité.*

4.3 Estimation de l'alignement

Comme pour l'étape de normalisation que nous venons de présenter, de nombreuses techniques d'indexation 3D par le contenu nécessitent un alignement des objets, en terme de rotation (voir section 3.2 du chapitre 3). Cette étape, différente du centrage et de la mise à l'échelle, a pour but d'orienter un objet par rapport aux trois axes, x, y et z d'un repère 3D. En d'autres termes, elle cherche à aligner certaines parties du modèle 3D en fonction de ces axes. Il peut, par exemple, être intéressant d'orienter le corps d'un animal selon l'axe x et ses pattes selon l'axe y. Le but final étant de positionner de façon identique les objets d'une même classe. L'intérêt majeur de cette réorientation et de pouvoir définir des vues caractéristiques contenant les informations de forme intrinsèques aux modèles.

Les méthodes que nous présenterons dans le chapitre 5 étant basées sur des approches "2.5D/3D" et "2D/3D", il est nécessaire qu'un alignement, à priori, de nos objets soit effectué. Bien que l'état de l'art proposé dans la section 3.2 du chapitre 3 introduise de nombreuses méthodes d'alignement, il n'offre pas de solutions quant à la sélection de la pose la plus adéquate pour un objet précis. Dans ce cadre, nous avons décidé d'explorer une nouvelle voie dans ce processus, en cherchant à faire une estimation de la qualité des alignements, cela en terme d'informations de forme projetées. Ainsi, il devient possible, en utilisant notre estimateur, de comparer le résultat de plusieurs approches pour aligner au mieux un objet 3D.

4.3.1 Recherche de l'alignement pertinent

Les objets 3D que l'on souhaite indexer par la suite sont modélisés au moyen d'outils divers qui ne garantissent pas un alignement correct. De ce fait, la pose initiale des objets 3D est parfois erronée, par rapport à notre repère 3D, et ne permet donc pas l'extraction de silhouettes pertinentes en terme de forme. Afin de résoudre ce problème, nous avons cherché une méthode offrant un alignement cohérent, pour l'ensemble des objets. Le but étant d'obtenir une pose identique, ou tout du moins proche, pour deux objets d'une même classe. C'est principalement cette contrainte de cohérence, intra-classe, que nous avons voulu mettre en avant avec notre nouvelle approche.

Même si les méthodes existantes permettent d'aligner correctement un grand nombre d'objets, elles ne sont pas robustes envers toutes les classes. En effet, les propriétés, sur lesquelles elles se basent, les amènent souvent à se spécialiser sur des types précis de modèles. Certaines méthodes offrent un alignement cohérent pour les objets manufacturés, d'autres pour les objets ayant une pose naturelle... Dans ce cadre, nous avons cherché à tirer parti de chacune de ces approches en estimant celle qui sera la plus appropriée à un modèle 3D précis. Cette estimation est basée sur un critère visuel mesurant la pertinence de l'orientation d'un objet 3D (voir figure 4.1). La méthode que nous présentons n'assure donc pas un alignement parfait des objets 3D mais plutôt le meilleur alignement possible avec un jeu de méthodes données.

Finalement, la force d'un critère d'estimation de l'alignement est sa capacité à sélectionner, à priori, la pose adéquate sans avoir besoin d'inclure l'invariance à la rotation dans l'étape d'indexation. Ainsi, le processus reste cohérent en dissociant les étapes de prétraitements des étapes de descriptions des objets 3D. D'autre part, cet estimateur permet de tirer parti de la force de plusieurs techniques d'alignement sans pour autant alourdir la signature en capturant une description selon différentes poses (voir remarque 4.2).

> **Remarque 4.2.** *Contrairement à notre approche, Papadakis* et al. *[92] proposent d'extraire la signature de chaque objet pour deux alignements. Lors de la comparaison d'un modèle par rapport à un autre, ils calculent une distance pour les deux poses et sélectionnent la distance minimale. Ainsi, étant donné les deux alignements, ils obtiennent de meilleurs résultats. Néanmoins, la taille du descripteur est deux fois plus importante. Notre approche cherche à tirer parti de ces observations sans pour autant alourdir le descripteur.*

4.3.2 Critère de minimisation de l'enveloppe visuelle

Comme nous venons de le présenter, nous avons cherchés à définir un critère d'estimation, permettant de mesurer la pertinence de l'alignement d'un objet 3D face à un autre. L'étape cruciale dans la construction d'un estimateur est la recherche d'un critère pertinent permettant l'évaluation des différentes approches. Pour définir un tel estimateur, nous avons donc tenté de comprendre ce que veut dire l'expression usuelle : "bien aligné", et ceci, dans le but d'extraire l'information pertinente des silhouettes, obtenues par projection, selon les trois axes du repère 3D (voir remarque 4.3). En effet, avant de pouvoir juger, de manière automatique, de la justesse d'un quelconque alignement, il faut d'abord savoir ce qu'un humain penserait. Mais surtout, comment il analyserait cette pose, et ceci, dans le but de donner, si ce n'est pas une valeur, au moins une approximation de sa pertinence. Finalement, c'est un critère d'alignement de l'objet 3D par rapport aux trois axes : $\overrightarrow{x}$, $\overrightarrow{y}$ et $\overrightarrow{z}$ que nous avons cherchés, plutôt qu'une recherche plus précise, incluant miroir et ordre de ces mêmes axes, ce que nous traiterons dans la suite de ce chapitre.

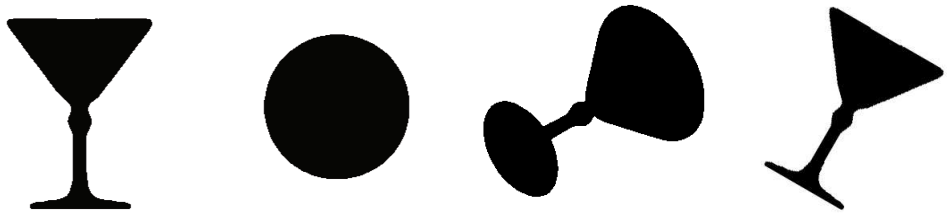
> **Remarque 4.3.** *Les méthodes d'indexation par le contenu, que nous introduirons par la suite, sont basées sur la capture de silhouettes, obtenues par projection de l'objet 3D sur un espace en deux dimensions. Pour simplifier la compréhension de l'estimation de l'alignement, nous allons supposer la capture de seulement trois silhouettes résultant de la projection d'un modèle en direction des trois axes principaux du repère 3D,* $\overrightarrow{x}$, $\overrightarrow{y}$ *et* $\overrightarrow{z}$. *Le but de cette étape est de trouver une pose cohérente de cet objet, et ceci, de manière rapide et si possible robuste.*

Étant difficile d'évaluer un alignement directement sur l'objet 3D, nous avons voulu en caractériser la justesse par rapport au repère 3D. Dans ce sens, nous avons choisi de juger une pose en fonction du rendu qu'elle offrira en deux dimensions plutôt qu'en trois. Nous avons donc cherché un critère sur les silhouettes 2D, et non directement sur le modèle 3D.

Relation à la psychologie cognitive

En psychologie cognitive, qui étudie les grandes fonctions psychologiques de l'être humain telles que le raisonnement, la résolution de problèmes ou la perception, Lemaire [67] a mis en évidence que l'être humain identifie des objets grâce à la représentation mentale qu'il s'en fait. En particulier, Marr [69] et Biederman [15] montrent que l'orientation des objets joue un rôle très important pour la reconnaissance d'un objet, but de notre indexation.

En effet, lors de la comparaison d'un objet avec son image mentale, un humain est plus rapide si l'objet est aligné de telle sorte que l'information qu'il contient soit maximale (voir section 4.2.1). Ainsi, dans la figure 4.7, l'image mentale qu'un humain se fait d'un verre est plus proche de (a) que de (b) ou (c). Ceci vient du fait que dans le cas d'une représentation binaire, noir et blanc, c'est la vue (a) qui véhicule le plus d'informations.



(a) *Verre vu de face* (b) *Verre vu de dessus* (c) *Verre vu de biais* (d) *Verre vu de face avec une rotation*

FIGURE 4.7 – *Exemple d'un verre vu de face (a), vu de dessus (b), vu de biais (c) et vu de face avec une rotation (d). Il est clair que la représentation mentale qu'un humain se fait de celui-ci se rapproche plus de la vue (a) car elle fournit un maximum d'informations et non pas des représentations (b), (c) ou (d).*

D'autre part, Leek [66] a montré que la reconnaissance d'un objet est plus facile pour un humain lorsque celui-ci est aligné selon ses axes de symétrie et qu'une rotation par rapport à ceux-ci complique la tâche d'identification. Un humain choisira donc plutôt la représentation (a) que (d).

Dans d'autres études, Marr [69] et Quinlan et Humphreys [101] montrent que nous apportons une très grande importance à la symétrie des objets pour leur représentation mentale. En effet, un humain représentera une table ou une chaise avec les pieds alignés car cette orientation tient compte des axes de symétrie et offre, de ce fait, plus d'informations (voir figure 4.8).

Finalement, la fonction de l'objet peut aider un humain à décider de la position dans laquelle il se le représente. En effet, son utilité influence parfois la représentation mentale que l'homme se fait de l'objet réel. Cependant, étant donné que cette dernière notion nécessite une information sémantique dont l'étude sort du cadre de notre thèse.

(a) Chaise avec les pieds bien alignés

(b) Chaise avec les pieds mal alignés

FIGURE 4.8 – *Exemple d'une chaise alignée de deux manières. La première (a) représente une chaise avec les pieds bien alignés tandis que (b) montre celle-ci avec les pieds décalés. La représentation mentale qu'un humain se fait de cet objet se rapproche plus de la vue (a) car elle répond à un alignement tenant compte des symétries du modèle 3D alors que la représentation (b) néglige ces informations (voir Leek Leek [66]).*

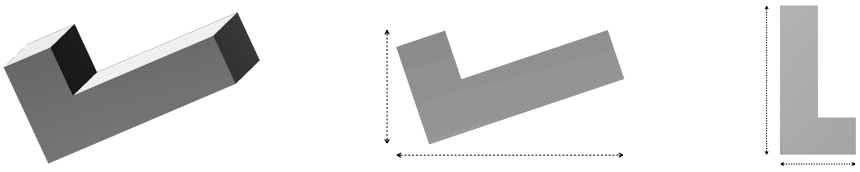
En accord avec ses différentes notions de psychologie cognitive, mais aussi avec nos propres contraintes (voir section 4.3.1) nous avons cherché un critère permettant d'estimer l'orientation d'un objet 3D, à partir de sa représentation 2D sous forme de silhouettes.

Recherche du critère d'estimation de l'alignement

La première approche qui peut paraître pertinente est : "la maximisation des aires des silhouettes extraites", et ceci, de manière itérative pour les trois silhouettes obtenues par projection, selon les axes du repère 3D. En effet, si l'on prend l'exemple d'une chaise, la première silhouette d'aire maximale correspond bien au dossier de celle-ci. Tandis que la seconde, qui doit être orthogonale à la première, correspond à l'assise. Cela semble être correct. Cependant, d'autres exemples font ressortir des contradictions.

Prenons le cas particulier d'une table. La première silhouette représentera bien le plateau de celle-ci, en revanche, la seconde, au lieu d'aligner les pieds de la table préféra faire en sorte qu'ils soient tous visibles afin de maximiser l'aire de la silhouette (voir la vue (b) de la figure 4.8). Un exemple encore plus simple (voir figure 4.9) permet de se rendre compte de l'ambiguïté qui apparaît lorsque l'on veut aligner certains objets 3D. En effet, pour ce "L" en 3D nous devrions chercher la silhouette ayant une aire maximale. Cependant, même si nous nous attendrions à choisir, pour celle-ci, la partie la plus longue, l'algorithme décrit choisira de coucher l'objet de travers, comme représenté sur la figure. En effet, d'après le théorème de Pythagore, la pose qui maximise l'aire de la projection 2D correspond à celle qui maximise l'hypoténuse du triangle formée par les deux barres du "L".

À cause de ces incohérences, nous avons décidé de choisir comme critère : "la minimisation de l'aire globale" – équivalente à la somme des aires des surfaces engendrées par les trois silhouettes – plutôt que leur maximisation. En effet, cet algorithme résout les problèmes soulevés pour la précédente approche. Même si ce critère d'estimation semble



(a) *Objet 3D "L" initial*

(b) *Pose maximisant l'aire des projections*

(c) *Objet "L" bien aligné*

FIGURE 4.9 – *Cas de l'alignement d'un objet 3D (a) en suivant l'approche de maximisation des aires. (b) représente l'alignement optimale en utilisant ce critère tandis que (c) montre l'alignement vu par un humain. Il est clair que l'alignement trouvé est incorrect car en cherchant à maximiser les aires, l'algorithme ne tient pas compte des surfaces planes et de leur alignement avec la prise de vue. En revanche (c) tient compte de ces paramètres et offre un alignement cohérent.*

à première vu, illogique, il s'avère très cohérent pour le jugement d'une pose par rapport à une autre, et ceci, pour deux raisons :

- La première est l'alignement des zones importantes, souvent caractérisées par de grandes surfaces planes, dans la direction d'un des axes du repère 3D.
- La seconde est la prise en compte des informations de symétries du modèle 3D.

En effet, la minimisation locale (sur chaque silhouette) de l'aire, ne veut pas dire minimisation globale (sur l'objet 3D). De ce fait, en cherchant à minimiser l'aire globale, l'algorithme présenté va préférer maximiser l'aire d'une zone sur une silhouette afin de la minimiser sur les autres, par exemple pour l'alignement d'un rectangle en 3D (voir 4.10). La figure 4.11 montre clairement que dans le cas d'une chaise le plateau est bien aligné et que les pieds le sont aussi. Il est intéressant de noter que dans ce cas, les critères d'orientation et de symétrie, en accord avec la psychologie cognitive, sont bien respectés répondant, de ce fait, à l'alignement qu'un humain aurait pu faire mentalement. Il faut tout de même noter que la méthode introduite propose un moyen de définir l'orientation d'un modèle 3D par rapport à un repère global, mais ne tient pas compte d'une éventuelle permutation d'axes ou d'un effet miroir.

Finalement, il est intéressant de noter que le critère choisi, basé sur la minimisation des projections tend à réduire le volume de la reconstruction de l'objet 3D. En effet, en utilisant trois silhouettes orthogonales, telles que dans la méthode, il est possible de reconstruire un modèle 3D avec des "voxels" (voir section 2.3.2 du chapitre 2). Il suffit pour cela de calculer les intersections entre les trois silhouettes est de placer un pixel 3D pour chacune d'entres elles. Ce critère de "minimisation du volume après reconstruction", visible sur la figure 4.12, aurait pu aussi être choisi comme critère de sélection de la pose.

(a) *Object 3D initial*

(b) *Pose maximisant l'aire des projections*

(c) *Pose minimisant l'aire des projections*

FIGURE 4.10 – *Cas de l'alignement d'un objet 3D (a) en suivant l'approche de minimisation des aires. (b) représente l'alignement "optimale" en utilisant le précédent critère de "maximisation" tandis que (c) montre l'alignement utilisant le critère de "minimisation". Il est clair que cette dernière pose offre de meilleurs résultats, pour cet objet, que l'approche évaluée précédemment..*

(a) *Avion*

(b) *Table*

(c) *Verre*

FIGURE 4.11 – *Trois objets 3D alignés selon le critère de "minimisation" de l'aire des projections. (a), (b) et (c) représentent respectivement un avion, une table et un verre.*

(a) *Objet 3D initial représentant une chaise*

(b) *Reconstruction de la chaise avec l'approche par "maximisation"*

(c) *Reconstruction de la chaise avec l'approche par "minimisation"*

FIGURE 4.12 – *Reconstruction d'un objet (a) (une chaise). (b) présente le résultat avec le critère de "maximisation" tandis que (c) présente le résultat avec le critère de "minimisation".*

Formalisation du critère d'estimation de l'alignement

Pour trouver cet alignement "optimal", par rapport à nos critères, nous définissons une rotation de l'objet 3D : $\mathbf{R}_{\rho,\phi,\theta}$ de paramètres : $(\rho, \phi, \theta) \in \mathbb{R}^3$. Nous recherchons donc les paramètres de cette transformation qui vont minimiser l'aire des silhouettes, de l'objet 3D $\mathcal{O}$, capturées sur les axes : $\overrightarrow{x}$, $\overrightarrow{y}$ et $\overrightarrow{z}$ du repère 3D. Nous pouvons ainsi définir l'enveloppe visuelle comme :

$$\mathcal{E}_\Theta = \sum_{v \in \{\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z}\}} f\Big((\mathbf{P}_v \circ \mathbf{T}_\Theta)(\mathcal{O})\Big) \tag{4.3}$$

avec $\mathbf{T}_\Theta = \mathbf{R}_{\rho,\phi,\theta}$ et $\mathbf{P}_v, v \in \{\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z}\}$ les axes de projections correspondants aux trois axes de notre repère 3D global, permettant le passage de trois à deux dimensions. Finalement, nous cherchons à minimiser l'enveloppe visuelle $\mathcal{E}_\Theta$:

$$\widehat{\Theta} = \arg\min_{\Theta} \mathcal{E}_\Theta \tag{4.4}$$

Nous cherchons en quelque sorte, à minimiser l'enveloppe visuelle de notre objet 3D, que nous définissons dans l'équation 4.3 comme la somme des aires des trois silhouettes, résultantes des projections sur les plans 2D. Ici, $f \in \mathbb{R}^{\mathcal{H}_v(\mathcal{O})}$ (avec $\mathcal{H}_v(\mathcal{O}) = [\mathbf{P}_v \circ \mathbf{T}_{\widehat{\Theta}}](\mathcal{O})$) mesure l'aire d'une silhouette. Grâce à la recherche de cette enveloppe visuelle minimale, chaque objet 3D d'une base de données pourra être orienté, au moyen de sa rotation associée, dans un repère 3D commun.

4.3.3 Critère simplifié

La formulation du critère d'estimation de l'alignement présentée ci-dessus permet de trouver la rotation d'un objet 3D en accord avec nos critères. En effet, en évaluant l'aire des silhouettes capturées en fonction des trois paramètres : $\rho \in \mathbb{R}, \phi \in \mathbb{R}$ et $\theta \in \mathbb{R}$ il est aisé de trouver l'enveloppe visuelle minimale. Cependant, un test exhaustif de l'espace des valeurs, que peuvent prendre ces paramètres, n'est pas envisageable, car étant de complexité $\mathcal{O}(n^3)$ où n est l'échantillonnage de chaque paramètre. De ce fait, nous avons cherché à réduire l'espace des paramètres dans le but de réduire la complexité de la recherche. Le calcul d'un sous-espace de ces valeurs étant peu pertinent, du fait du choix difficile de celui-ci, nous avons choisi d'utiliser un ensemble d'orientations calculé mathématiquement sur l'objet 3D. Ainsi, même si indépendamment, les poses choisies sont peu robustes envers toutes les classes, l'estimation de celle-ci permet de tirer parti des forces de chacune, en choisissant l'orientation la plus adaptée à chaque modèle 3D. Il faut cependant noter que la solution que nous proposons est "sous-optimale" par rapport au test exhaustif de l'espace des paramètres.

Cette nouvelle approche va donc nous permettre de faire un choix entre plusieurs alignements en fournissant à l'estimateur un ensemble d'orientations non aléatoires et si possible pertinentes. Afin de s'approcher au mieux d'une pose correcte, tout en limitant le temps de calcul, nous avons choisi, pour notre critère, de ne considérer que trois orientations possibles qui sont : la pose initiale de l'objet et deux analyses en composantes

principales "ACP".

Le choix de l'orientation initiale est très intéressant. En effet, dans de nombreux cas, l'objet est numérisé, ou modélisé, avec un alignement correcte. Ceci vient du fait que lors de l'utilisation d'un scanner, le manipulateur positionne l'objet dans l'espace de capture selon une orientation correct, car venant d'un humain. D'autre part, pour les objets modélisés, dans la majeure partie des cas, l'infographiste oriente sa création de telle sorte qu'elle soit interprétable par un homme, garantissant, de ce fait, un alignement cohérent.

Cependant, une partie des objets n'étant pas ou mal alignés, il est nécessaire de considérer d'autres orientations comme valeurs du sous-espace des paramètres de l'alignement "sous-optimal". Dans ce sens, nous avons choisi d'utiliser, en plus de la pose initiale, deux poses issues de l'"ACP" afin de diversifier notre sous-espace de paramètres. La première est basée sur les centres des facettes de l'objet 3D, et cherche à aligner un objet en fonction de la position de celles-ci [1]. La seconde, quant à elle, est basée sur les normales et tente d'orienter le modèle 3D à partir de l'orientation de ces mêmes facettes [2].

4.3.4 Ordonnancement des axes de rotations et réflexions

Dans l'approche d'estimation de la pose proposée, l'objet 3D est orienté en sélectionnant la meilleure pose parmi un ensemble d'alignements calculé. Cependant, même si, grâce à cette nouvelle approche, les trois axes permettant de placer l'objet dans un repère 3D commun sont relativement robustes, l'ordre de ceux-ci est trop peu pertinent pour que nous puissions en tenir compte. En effet, si l'objet 3D a des élongations proches sur deux ou trois axes principaux (par exemple $\lambda_1 \simeq \lambda_2$ où λ_1, λ_2 sont les valeurs propres), alors l'ordre des axes devient instable car les vecteurs propres peuvent commuter (voir figure 4.13).

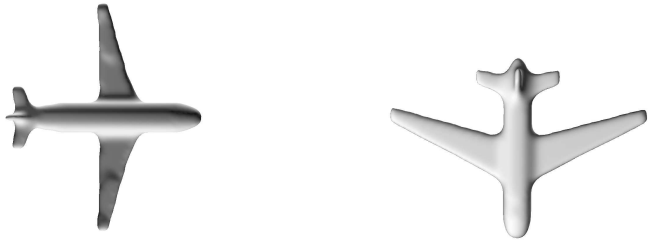


FIGURE 4.13 – *Exemple d'une permutation d'axes sur deux objets 3D de la même classe "avion".*

Plusieurs approches pourraient être envisagées pour ordonner les valeurs propres λ_1, λ_2 et λ_3 autrement que par leurs ordres $\lambda_1 > \lambda_2 > \lambda_3$. En effet, il serait par exemple possible de les classer selon l'aire des silhouettes qu'il engendre, avec la première vue, dans ce cas, $\overrightarrow{x}$ qui correspondrait à la silhouette d'aire maximum et de même pour les axes $\overrightarrow{y}$ et

1. Nous utiliserons l'abréviation *ACPC* pour cette Analyse en Composantes Principales Continue.
2. Nous utiliserons l'abréviation *ACPN* pour cette Analyse en Composantes Principales par Normales.

$\overrightarrow{z}$. Cependant, cette approche reste encore trop instable pour pouvoir l'utiliser dans la méthode d'indexation que nous proposons (voir Zaharia et Preteux [141]). De ce fait, nous suggérons, comme Zaharia et Preteux [141], d'utiliser toutes les permutations possibles de ces trois axes, afin de rendre robuste notre alignement. Ainsi, nous décrivons un objet 3D avec $3! = 6$ orientations différentes correspondant à l'alignement sélectionné, avec notre critère d'estimation, associé à toutes les configurations possibles des trois axes de notre repère 3D. De ce fait, le bon ordonnancement des axes sera présent dans notre alignement car faisant, forcément, partie des 3! configurations calculées (voir figure 4.14).

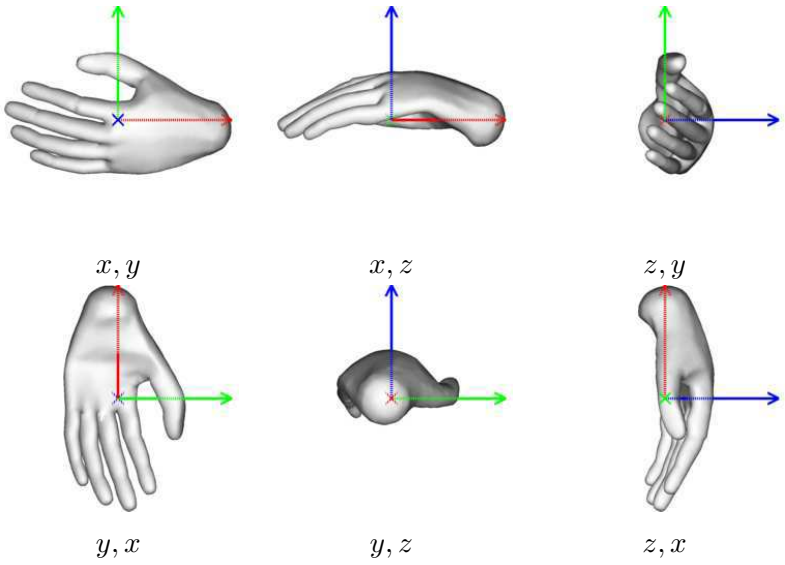


FIGURE 4.14 – *Ensemble des 6 permutations possibles des axes du repère permettant le rendu objet 3D.*

Après la résolution du problème de commutation d'axes, il nous reste à déterminer la pertinence de l'orientation, $\overrightarrow{x}$ ou $-\overrightarrow{x}$ par exemple, de ceux-ci. En effet, nous nous sommes posé la question de l'intérêt de considérer les représentations "miroir" des objets 3D de la base. Il est clair que pour un grand nombre d'objets symétriques, la question ne se pose pas, cependant, pour certains modèles, tels que, par exemple les "mains", il peut être intéressant de prendre en compte ce problème. Dans le cas ou nous ne considérerions pas ces représentations "miroir", nous pourrions dissocier les "mains gauches" des "mains droites" et ainsi répondre, peut-être, de manière plus pertinente à la requête de l'utilisateur. Dans l'autre cas, nous ne considérerions que les "mains" dans un cas général sans distinction particulière.

Pour traiter ce dernier cas, il faut tenir compte de tous les miroirs possibles pour les axes du repère 3D global. Il suffit pour cela de décrire, comme Zaharia et Preteux [141] l'objet 3D avec $2^3 = 8$ réflexions différentes qui correspondent à l'ensemble des signes possibles pour les axes du repère (voir figure 4.15). Une autre approche proposée par Vranic et Saupe [135][136][134], résout cette invariance en réflexion en calculant les distances signées des

facettes de l'objet 3D par rapport aux trois axes principaux obtenus au moyen de l'"ACP Continue". Cependant, cette approche n'est pas assez robuste pour certains types d'objets, la rendant, de ce fait, peu adaptée à notre méthode d'indexation. Nous avons donc choisi, afin de tenir compte de tous les miroirs, d'utiliser la méthode de Zaharia et Preteux [141] malgré son aspect peu synthétique (voir remarque 4.4). Finalement, le processus global de normalisation et d'estimation de l'alignement utilisé est représenté dans la figure 4.16.

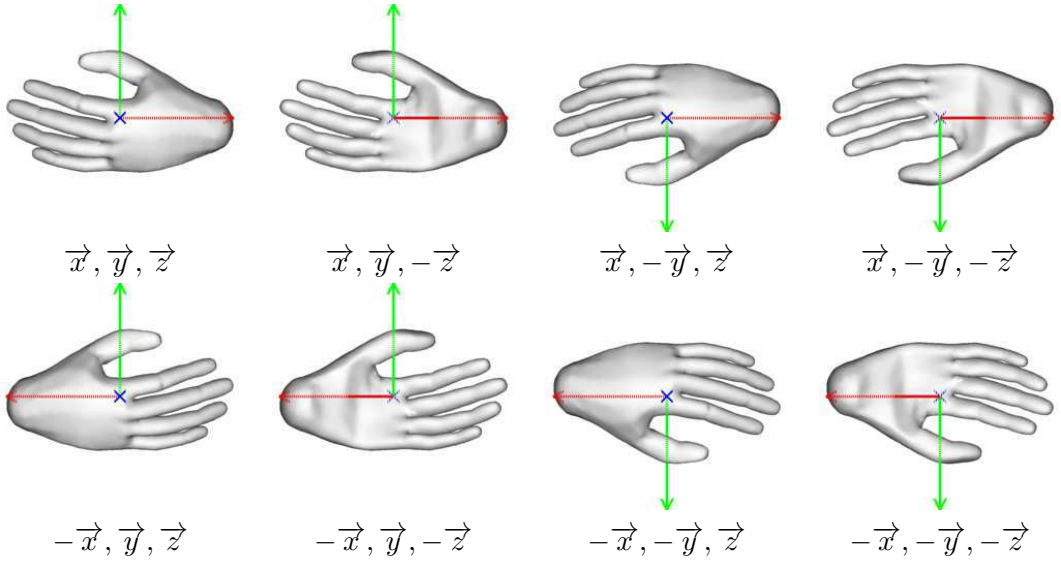


FIGURE 4.15 – *Ensemble des 8 réflexions possibles des axes du repère permettant le rendu objet 3D.*

> **Remarque 4.4.** *En apparence l'approche choisie, pour traiter les permutations d'axes et les réflexions, augmente considérablement la taille du descripteur –* $3! * 2^3 = 48$ *au lieu de 1 –. Cependant, ce traitement n'est nécessaire que pour la requête de l'utilisateur permettant d'interroger la base de données. En effet, il est inutile de stocker l'ensemble de ces configurations à la fois pour chacun des modèles 3D de la bibliothèque et pour les requêtes. Ainsi, l'espace de stockage n'est pas affecté et permet de garder une description synthétique des objets.*

4.4 Conclusion

Une étape de prétraitements étant nécessaire pour l'extraction pertinente des caractéristiques de nos objets 3D en vue d'une indexation. Dans ce chapitre, nous avons proposé une nouvelle approche de normalisation ainsi qu'une méthode permettant l'estimation de l'orientation des modèles dans le but de les aligner. Ces deux étapes assurent une extraction optimale, en terme de qualité des contours. Ceci passe par une maximisation de leurs tailles et un alignement des objets 3D grâce aux points forts d'autres approches permettant d'orienter un modèle. Il est intéressant de noter que nos méthodes ont été développées en relation avec les points clés de l'analyse humaine. C'est finalement un cadre complet que nous avons développé, assurant un prétraitement efficace à la suite des opérations, à

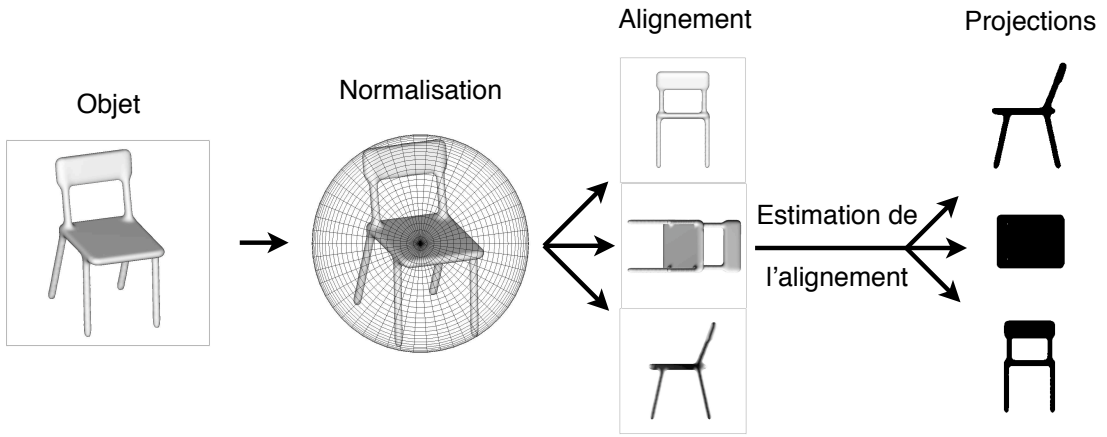


FIGURE 4.16 – *Cette figure présente le processus de normalisation et d'alignement d'un objet.3D de la base de données. Dans un premier temps, nous calculons la sphère minimale englobante du modèle que nous combinons ensuite à une estimation de la pose par minimisation de l'enveloppe visuelle dans le but d'aligner au mieux l'objet 3D. Finalement, cette étape permet l'extraction des silhouettes du modèle en vue d'une indexation 3D.*

savoir l'indexation 3D.

Afin de simplifier l'étape, visant à normaliser les objets avant leur indexation, nous aurions pu chercher à définir le centre et l'échelle directement sur les silhouettes 2D extraites. Cependant, cette approche, bien que fonctionnelle, s'éloigne de l'aspect 3D du modèle. En effet, le fait d'appliquer des rotations, translations ou changements d'échelles sur les silhouettes, de façon indépendante, rompt la cohérence spatiale 3D. Il est simple de se rendre compte de ce phénomène si l'on tente de reconstruire le modèle 3D, a posteriori, en utilisant les silhouettes normalisées et alignées indépendamment. Dans notre cas, nos processus de prétraitements garantissent une cohérence spatiale en liant l'extraction des silhouettes avec la pose canonique des objets 3D, garantissant, de ce fait, le bon déroulement de la phase d'indexation introduite dans le chapitre 5.

FIGURE 4.17 – *Résultats de notre processus de normalisation et d'estimation de la pose sur la base de SHREC'07 Watertight (voir annexe D) – partie 1 –.*

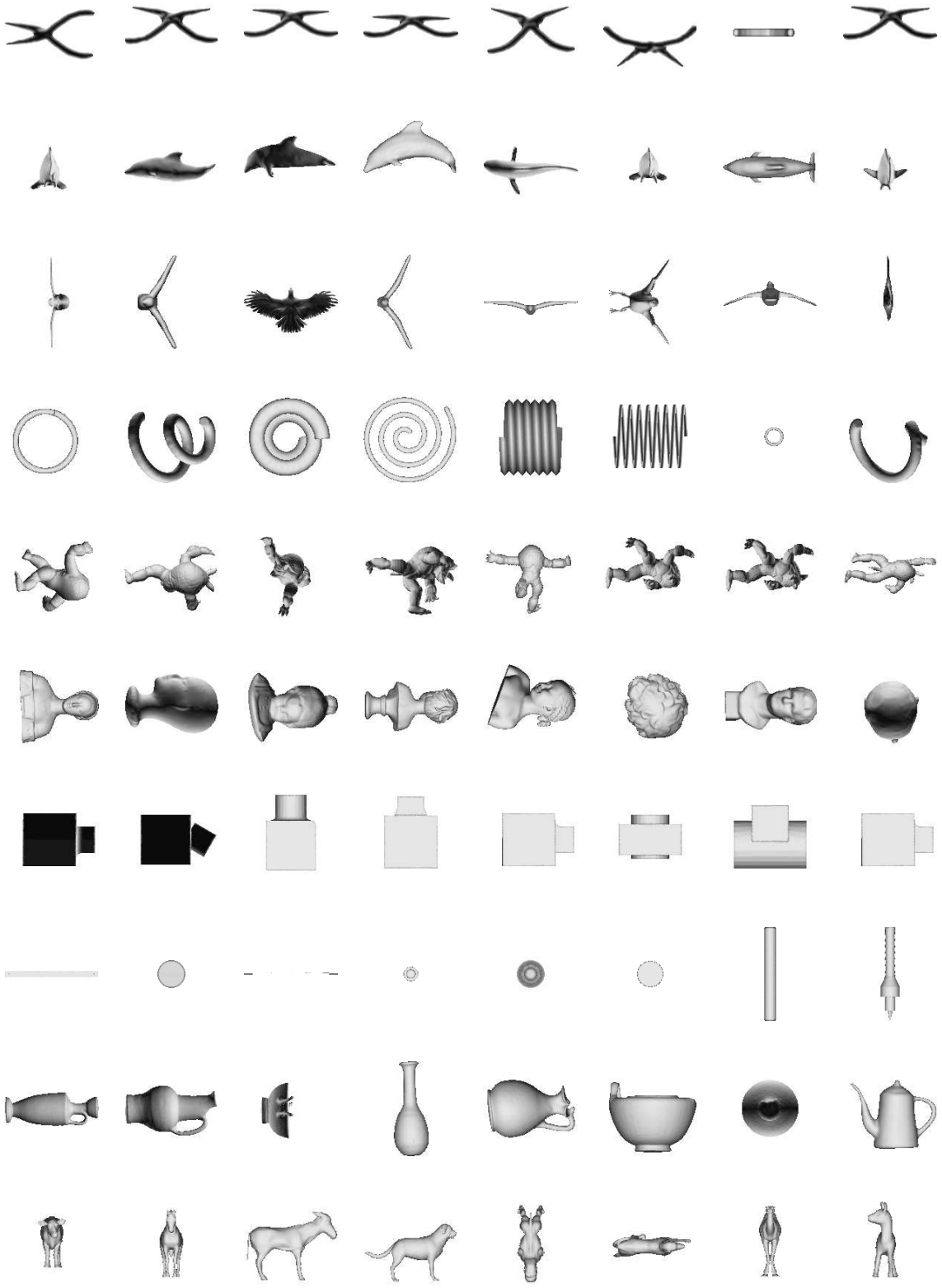


FIGURE 4.18 – *Résultats de notre processus de normalisation et d'estimation de la pose sur la base de SHREC'07 Watertight (voir annexe D) – partie 2 –.*

CHAPITRE 5

Indexation d'objets 3D

Résumé *: Dans ce chapitre nous développons notre seconde contribution portant sur la caractérisation de la forme d'un objet 3D. Nous proposons en particulier trois nouveaux descripteurs basés sur des ensembles de pixels, sur des convexités et concavités des contours et sur des cartes de normales.*

5.1 Introduction

La recherche d'informations, dans les grandes bases de données, nécessite de mettre en place des techniques adaptées permettant un accès rapide et facile aux documents. Pour cela, l'indexation de contenus offre une solution efficace en sélectionnant les données caractérisant le modèle 3D et en les synthétisant dans une représentation, couramment appelée descripteur ou signature. Dans le cas particulier des objets 3D, les métadonnées, ou mots-clés, qui sont généralement associés aux données, et qui permettent de les caractériser, ne sont pas présentes et nous obligent donc à analyser leur forme.

Lors de la conception d'un descripteur permettant de stocker, de façon synthétique, les caractéristiques d'un modèle 3D, une attention particulière doit être apportée à certains points. Le premier est la conception d'une description au fort pouvoir discriminant garantissant une recherche rapide et efficace. Le second est la proximité de la représentation avec l'utilisateur, permettant ainsi de lui offrir une interrogation multimodale, de la bibliothèque. Finalement, il faut prendre en compte les contraintes liées aux données – les objets 3D – à la machine et enfin aux utilisateurs (voir dans le chapitre 2 section 2.2.5).

Afin de répondre à ces contraintes tout en restant proche de l'utilisateur, nous avons pris le parti de pouvoir rechercher un objet 3D de deux manières : avec un modèle en trois dimensions ou directement à partir de silhouettes en deux dimensions (voir chapitre 2). Pour cela, nous décrivons chaque modèle 3D avec un ensemble de descripteurs "2.5D/3D" et "2D/3D". La principale tâche du processus d'indexation consiste donc à extraire un ensemble de silhouettes 2D puis de sélectionner les informations pertinentes sur leur forme.

Nous débutons ce chapitre avec la section 5.2 introduisant les moyens techniques permettant de transformer un objet 3D en vues 2D. Nous introduisons ensuite trois descriptions différentes. La première est directement basée sur l'étude de la forme générale des silhouettes 5.3. La seconde décrit le modèle au moyen d'une étude approfondie des silhouettes. Elle capture en particulier les convexités et les concavités des contours de celles-ci 5.4. Finalement, la dernière décrit le modèle 3D en utilisant une information de

surface par la notion de normales en tout point du maillage 5.5. Finalement, notons que l'invariance aux transformations géométriques est obtenue par les étapes de normalisation et d'estimation de l'alignement présentées dans le chapitre 4.

5.2 Extraction multi-vues de silhouettes

L'indexation d'objets n'étant pas nécessaire si elle n'est pas associée à un processus de recherche, il nous a semblé pertinent de choisir des descriptions permettant une recherche aisée, basée en particulier sur des silhouettes (voir chapitre 2). Dans cette optique, nous cherchons à extraire une description de forme, de l'objet 3D, à partir d'un ensemble de vues 2D de celui-ci. Ces descriptions s'obtiennent par une analyse précise de la forme après un processus d'extraction de silhouettes autour du modèle en question. Il s'agit en quelque sorte, de décrire un objet 3D en analysant les ombres qu'il projette sur des murs, en supposant chaque source de lumière comme étant unidirectionnelle (deux "lampes" ne sont pas allumées en même temps) et qu'il n'y a pas de notion de perspective.

Dans notre cas, il est donc nécessaire de passer d'un modèle en trois dimensions à une représentation en deux dimensions. Pour réaliser cette opération, nous avons souhaité prendre en compte deux contraintes. La première tend à extraire les vues caractéristiques afin de décrire au mieux l'objet (voir 5.2.1). La seconde, quant à elle, oblige à choisir un nombre restreint de vues permettant, grâce à l'aspect synthétique de cette nouvelle représentation 2D, de faire une analyse plus pertinente du contenu en se focalisant sur les vues présentant les informations essentielles. Ce passage d'un espace 3D vers 2D est associé à un ensemble de techniques efficaces assurant une capture optimale des silhouettes avec les outils décrits (voir 5.2.2).

5.2.1 Choix des vues

Pour indexer efficacement nos objets 3D à partir de silhouettes en deux dimensions, l'extraction de celles-ci doit répondre à nos deux contraintes, qui sont : l'extraction de vues optimales et leur nombre restreint. Afin de répondre efficacement à celles-ci, nous avons choisi de capturer un ensemble de silhouettes ou projections, autour de l'objet 3D, à une distance égale du centre du modèle. Cette opération est réalisée en se plaçant sur la sphère minimale englobant l'objet 3D à indexer (voir section 4.2). La question qui se pose alors est de connaître la position et le nombre de "murs" sur lesquels nous allons projeter l'objet 3D. Ceci en sachant que ces silhouettes doivent garantir la pérennité des informations cruciales contenues dans le modèle. L'état de l'art du chapitre 3 a montré que des méthodes permettant la sélection des vues caractéristiques d'un objet 3D existent. Cependant, l'analyse offrant cette sélection étant difficile et peu robuste, nous avons choisi d'extraire un nombre fixe de vues représentées par un ensemble de plans de projection.

Remarque 5.1. *Une des questions qui pourrait se poser est : Est-il indispensable d'extraire plusieurs vues ou projections ? La réponse est oui. En effet, on remarque qu'une unique projection est identique au principe des ombres chinoises. Dans celui-ci, l'artiste joue avec les ombres pour représenter à l'aide de ses mains, ou de tout autre objet, une ombre particulière. Comme on peut le voir dans la figure 5.1, cette ombre peut être trompeuse et représenter dans une direction bien choisie, un objet autre que l'original. Afin d'éviter cette situation, il est donc indispensable d'extraire plusieurs silhouettes dans des directions différentes.*

(a) *Oeuvre de l'artiste anglais Tim Noble nommé "Dirty White Trash (With Gulls)" réalisée en 1998 qui montre une ombre chinoise représentant deux personnes assises*

(b) *Affiche du colloque de l'association pour la recherche cognitive 2009 représentant une ombre chinoise.*

FIGURE 5.1 – *L'intérêt, pour nous, outre l'aspect artistique, est le jeu de projection. En effet, seul un axe de projection bien précis permet d'obtenir l'œuvre artistique, d'où le besoin d'indexer plusieurs vues dans nos travaux d'indexation*

La sélection du nombre de silhouettes à extraire est un point critique pour indexer un objet 3D avec des informations 2D. En effet, deux situations problématiques peuvent apparaître et ainsi rendre difficile l'analyse et l'extraction des caractéristiques visibles sur les vues :

1. La première est la capture d'un nombre trop réduit de silhouettes ne permettant pas, de ce fait, une pérennité des propriétés indispensables à la bonne analyse du contenu de celui-ci. Cela a pour effet de rendre difficile, voir impossible, l'indexation, (sur la plan de la caractérisation de la forme) et donc la recherche, de l'objet 3D (voir remarque 5.1). Par exemple, la représentation d'un modèle 3D a de fortes chances d'être erronée si elle n'est considérée que dans une seule direction. C'est la "sous-indexation".
2. La seconde situation problématique qui peut apparaître est le cas inverse, qui par peur de perdre les informations nécessaires à la bonne description d'un objet 3D, capture un nombre de silhouettes trop important. Dans ce cas, l'information est

présente, mais difficile à extraire car noyée parmi la multitude d'informations non pertinentes. Ceci se rapproche des problèmes liés au approches "3D/3D" face aux approches "2D/3D" présentées en conclusion du chapitre 3. C'est la "sur-indexation".

Notre but est donc d'extraire un nombre de silhouettes adéquat, permettant une analyse pertinente, en se focalisant sur les silhouettes comportant un maximum d'informations et en éliminant celles qui sont inutiles.

La question que nous nous sommes posée à donc été : Comment extraire les vues pour qu'elles contiennent l'information décrivant au mieux l'objet 3D ? Nous avons répondu à cette question par deux réponses distinctes :

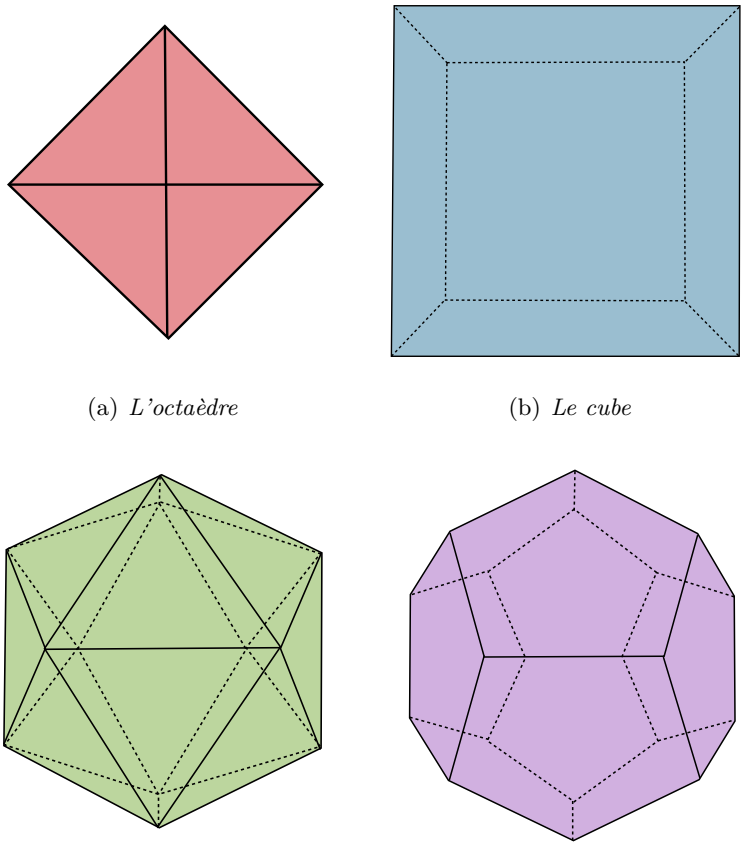
1. Les silhouettes à extraire doivent s'appuyer sur l'alignement préalable de l'objet afin de tirer parti des caractéristiques de celui-ci.
2. Les silhouettes doivent être réparties uniformément autour de l'objet. Dans notre cas sur la sphère unité englobant le modèle 3D (voir section 4.3).

En effet, sachant qu'une étape de prétraitement nous a permis d'aligner les plans de symétrie de l'objet avec les axes du repère, il nous parait intéressant de tenir compte de cela dans le processus d'extraction des silhouettes. De plus, l'approche choisie pour aligner l'objet 3D se rapproche de l'esprit humain. On rappelle que celui-ci caractérise généralement un objet avec des vues qui sont basées sur les plans de symétrie et l'alignement des grandes surfaces planes (voir remarque section 4.3.2 du chapitre 4). D'autre part, nous avons vu (voir remarque 5.1) qu'un ensemble de vues réparties uniformément autour de l'objet permet de s'affranchir des problèmes de représentation liés aux projections.

Afin de valider nos propos et de comparer différents jeux de silhouettes, nous proposons 10 ensembles de vues, à extraire pour chaque objet 3D. Notre choix s'est porté sur les solides de Platon, ou polyèdres réguliers convexes, qui ont des sommets uniformément répartis sur la sphère unité (voir figure 5.2). Afin de prendre en compte notre remarque sur l'importance de lier l'alignement avec l'extraction des vues, nous combinons plusieurs de ces solides avec les axes principaux du modèle (voir remarque 5.2). Voici la liste des dix jeux de points de vues que nous utilisons :

1. Les **6** sommets de l'octaèdre (figure 5.2 (a)).
2. Les **8** sommets de l'hexaèdre (cube) (figure 5.2 (b)).
3. Les **12** sommets de l'icosaèdre (figure 5.2 (c)).
4. Les **20** sommets du dodécaèdre (figure 5.2 (d)).
5. Les $\mathbf{6 + 8 = 14}$ sommets de l'octaèdre et de l'hexaèdre (figure 5.2 (a) et (b)).
6. Les $\mathbf{6 + 12 = 18}$ sommets de l'octaèdre et de l'icosaèdre (figure 5.2 (a) et (c)).

7. Les $\mathbf{6 + 20 = 26}$ sommets de l'octaèdre et du dodécaèdre (figure 5.2 (a) et (d)).
8. Les $\mathbf{6 + 6 = 12}$ sommets de l'octaèdre et de 6 bissectrices (figure 5.3 (a)).
9. Les $\mathbf{6 + 12 = 18}$ sommets de l'octaèdre et de 12 bissectrices (figure 5.3 (b)).
10. Les $\mathbf{6 + 12 = 18}$ sommets de l'octaèdre et de 12 trisectrices (figure 5.3 (c)).



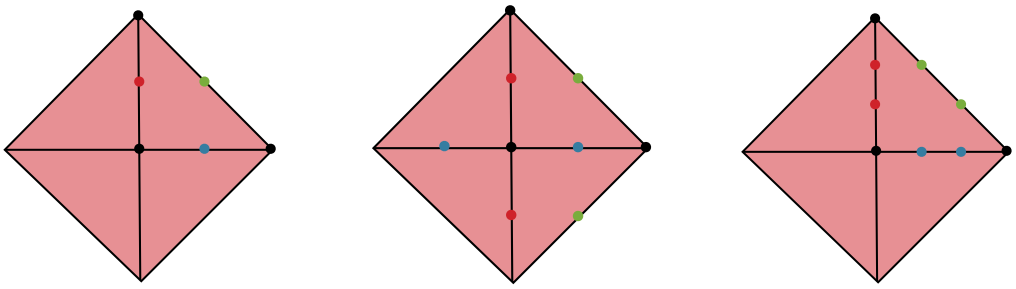
(a) *L'octaèdre* (b) *Le cube*

(c) *L'icosaèdre* (d) *Le dodécaèdre*

FIGURE 5.2 – *Les quatre solides de Platon utilisés pour placer nos points de vues. L'octaèdre (a), l'hexaèdre ou cube (b), l'icosaèdre (c) et le dodécaèdre (d).*

> **Remarque 5.2.** *Notons que les axes principaux de l'objet correspondent aux sommets de l'octaèdre. En effet, si celui-ci est orienté comme sur la figure 5.3 alors on peut tirer parti de l'alignement du chapitre 4. Le choix de six vues positionnées sur les axes du repère 3D permettent finalement de décrire le modèle 3D avec des projections naturelles proches de la vision humaine (voir section 4.3.2 du chapitre 4).*

Pour capturer chaque silhouette d'un ensemble, nous définissons pour chacun des points de vues de celui-ci un centre c_i, un point de visé v et un vecteur "nord" $\overrightarrow{n_i}$ qui définit l'orientation de la prise de vue (voir figure 5.4). Ces paramètres sont définis de la manière



(a) *L'octaèdre + 3 bissectrices* (b) *L'octaèdre + 6 bissectrices* (c) *L'octaèdre + 6 trissectrices*

FIGURE 5.3 – *Représentations d'un octaèdre avec des points de vues positionnées sur les sommets plus 3 bissectrices (a), 6 bissectrices (b) et 6 trisectrices (c).*

suivante :

- c_i correspond à la position dans $\mathbb{R}^3$ de chaque sommet des polyèdres de Platon listés précédemment. Un exemple de ces points de vues est visible sur la figure 5.3.
- v est toujours égal au centre du repère $\mathbb{R}^3$ de l'objet, à savoir, $(0, 0, 0)$.
- Le vecteur $\overrightarrow{n_i}$ pointe vers l'axe $\overrightarrow{y}$ dans tout les cas, excepté quand la position c_i de la caméra vaut (..., ± 1, ...), dans quel cas $\overrightarrow{n_i}$ pointe vers l'axe $\overrightarrow{z}$.

Ce qui donne pour le premier jeu de vues (octaèdre) de la figure 5.2 :

$$\begin{aligned} s_1 &= \{c_1,\ v,\ \overrightarrow{n_1}\} |\ c_1 = (1,\ 0,\ 0),\ v = (0\ 0\ 0),\ \overrightarrow{n_1} = (0,\ 1,\ 0) \\ s_2 &= \{c_2,\ v,\ \overrightarrow{n_2}\} |\ c_2 = (0,\ 1,\ 0),\ v = (0\ 0\ 0),\ \overrightarrow{n_2} = (0,\ 0,\ 1) \\ s_3 &= \{c_3,\ v,\ \overrightarrow{n_3}\} |\ c_3 = (1,\ 0,\ 0),\ v = (0\ 0\ 0),\ \overrightarrow{n_3} = (0,\ 1,\ 0) \end{aligned} \tag{5.1}$$

Nous pensons que l'extraction des vues est un point essentiel pour garantir la pertinence de l'information, contenue dans chacune des silhouettes 2D représentant le modèle 3D. Dans ce cadre, nous avons alors introduit un ensemble de points de vues basé sur les sommets des polyèdres de Platon, en particulier pour leurs propriétés de répartition uniforme et leurs liens avec l'alignement présenté au chapitre 4. Il reste cependant à passer de trois à deux dimensions. Pour cela nous introduisons dans la section suivante une technique permettant de projeter efficacement un objet 3D sur différents plans 2D.

5.2.2 Principe d'extraction

Après la définition des différents points de vues, il est nécessaire d'extraire chacune des silhouettes qui leur sont associées. Afin de réaliser cette tâche, nous allons utiliser une application permettant de représenter dans un environnement tridimensionnel l'objet 3D étudié. Celle-ci, en plus d'afficher le maillage de chacun des modèles 3D, doit permettre l'extraction efficace des différentes projections introduites dans la section précédente.

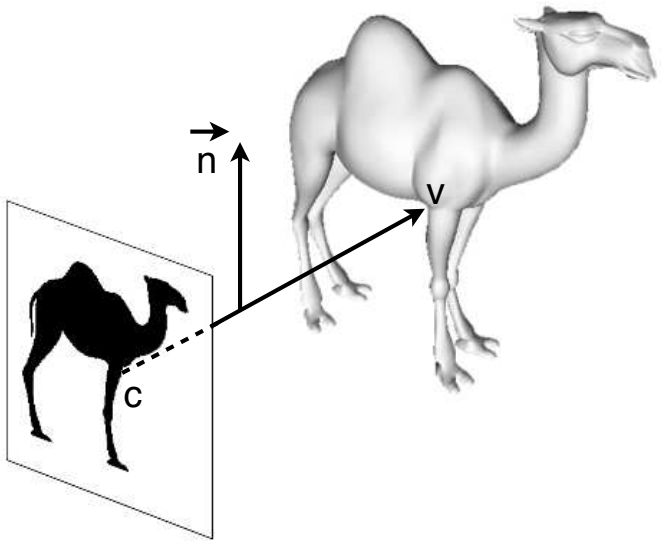


FIGURE 5.4 – *Représentation du centre c de la projection, du point de visée v et du nord* $\overrightarrow{n}$.

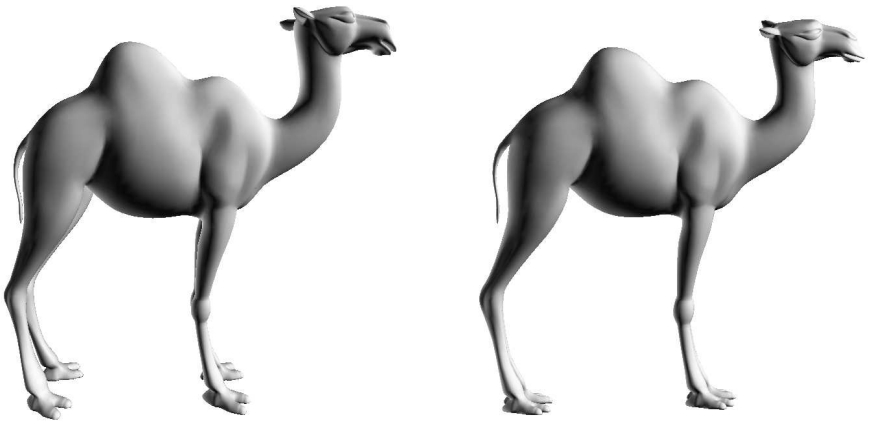
Pour faire face à ces contraintes techniques, nous utilisons la librairie graphique 3D *OpenGL* [1]. Celle-ci permet d'afficher un modèle en trois dimensions à partir d'un fichier contenant faces et sommets avec des options de couleurs, de textures et de rendus. Dans notre cas plusieurs paramètres sont à régler, à savoir : la caméra pour la projection, l'éclairage, la couleur et le rendu de l'objet 3D.

Les librairies graphiques, telles que celle utilisée, offrent une grande flexibilité pour traiter des données tridimensionnelles. En effet, en plus de permettre la représentation virtuelle d'un modèle 3D, elles offrent un jeu d'outil pour interagir avec ces données. Il est, par exemple, possible de définir une caméra permettant de se placer à un endroit précis de la scène, ceci afin de regarder l'objet 3D dans une direction précise. Il s'agit en fait d'une projection des données 3D dans une direction particulière, contrôlée par un ensemble de paramètres.

C'est donc naturellement par cette caméra que nous capturons les différentes silhouettes qui nous intéressent. Pour cela, après avoir affiché l'objet 3D nous plaçons la caméra fictive grâce aux paramètres de points de vues c_i, v et $\overrightarrow{n_i}$ définies dans la section précédente. Il reste cependant à choisir le type de projection de cette caméra. Quelle projection choisir ? Perspective ou orthogonale ? Dans notre cas, cette deuxième projection, malgré une représentation plus "plate" du modèle 3D offre des propriétés intéressantes (voir

1. http://www.opengl.org/

figure 5.5). En effet, nous pouvons remarquer qu'avec ce type de projection, le contour obtenu par deux caméras situées à l'opposé l'une de l'autre, par rapport au centre du repère, sera le même à un miroir près (voir figure 5.6). Grâce à cette observation, il nous est possible de réduire de moitié le nombre de vues, dans le cas ou seul le contour ou la silhouette de l'objet 3D nous intéresse. En revanche, dans le cas où nous souhaitons capturer une information de profondeur ou d'orientation des surfaces, cette optimisation ne peut pas s'appliquer. Nous utiliserons donc l'intégralité des vues définies dans la section précédente pour notre descripteur à base de cartes de normales. Finalement, les résolutions que nous avons choisies – pour le capteur de la caméra virtuelle – pour chaque projection sont $64 * 64$ et $256 * 256$ pixels.

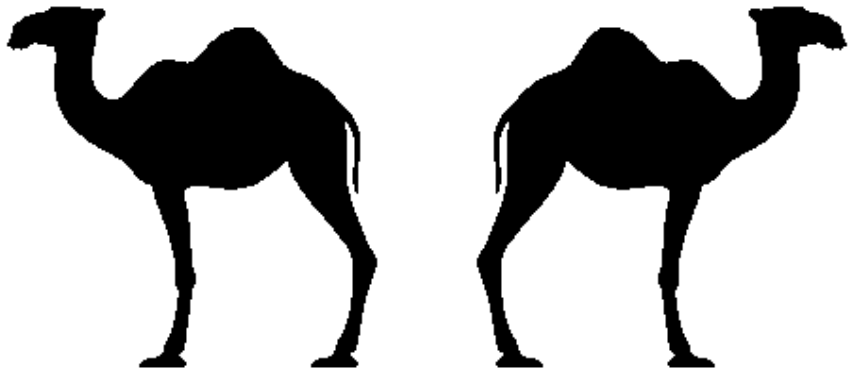


(a) *Projection perspective* (b) *Projection orthogonale*

FIGURE 5.5 – *Exemple d'un dromadaire affiché dans un environnement tridimensionnel. (a) représente un rendu en perspective tandis que (b) représente un rendu orthogonal.*

Nous venons de présenter la méthode permettant l'extraction des vues, il est maintenant nécessaire d'introduire différents paramètres de rendus permettant de projeter uniquement les informations qui nous intéressent. Les descripteurs que nous allons proposer et utiliser dans la suite de ce manuscrit sont aux nombres de trois et sont basés sur différentes informations de forme. Afin de capturer ces informations, nous avons dû paramétrer le rendu différemment pour le dernier que pour les deux premiers. En effet, ces derniers utilisent les informations de contour et de surface de la projection tandis que le troisième utilise une notion de variation de l'orientation du maillage dans l'espace. Les paramètres choisis sont les suivants :

– **Descripteurs 1 et 2 :** Étant basés sur des informations de surface et de contour ils ne nécessitent donc que d'un rendu monochrome tel que sur la figure 5.6. Ce type de projection est obtenu en désactivant l'éclairage de l'objet affiché avec une couleur noire et en donnant une couleur blanche au fond. Ainsi, seule la silhouette de l'objet est visible et toute information de profondeur ou de maillage est effacée (voir figure



(a) *Silhouette originale*

(b) *Silhouette capturée sur la caméra symétrique par rapport au centre du repère*

FIGURE 5.6 – *Exemple d'un dromadaire capturé avec une projection orthogonale selon un premier point de vue (a) et son opposé par rapport au centre du repère (b).*

5.7).

– **Descripteur 3 :** Étant donné qu'il utilise des informations d'orientations de la surface du modèle 3D, ce descripteur a besoin d'une information plus riche, à savoir la normale du maillage en chaque point. Pour capturer cette nouvelle donnée, outre la nécessité de calculer la normale en chaque sommet, que nous présenterons dans la partie consacrée à cette description, il est nécessaire d'obtenir une valeur pour chaque point du maillage 3D. Pour optimiser ce calcul, nous utilisons l'interpolation offerte par *OpenGL* pour le traitement de la couleur des faces. Celle-ci permet de calculer la couleur de chaque point d'une face à partir des couleurs de ses sommets. Afin d'utiliser cette accélération matérielle pour le traitement des normales, nous codons les trois composantes x, y et z des normales dans les composantes couleur r, g et b de l'objet. Associées à l'interpolation des couleurs on obtient alors un rendu tel que présenté sur la figure 5.8 où l'information d'orientation est contenue dans la couleur de chaque pixel. On appellera cette image "carte de normales" dans la suite de ce manuscrit.

Finalement, un traitement graphique a été mis en place afin de limiter le nombre de composantes connexes de chaque silhouette. En effet, l'idéal serait, pour pouvoir traiter l'information de forme du contour, de n'avoir pour chaque projection qu'une et une seule composante. Cependant, nous nous sommes rendu compte que le rendu classique *OpenGL* avait tendance à générer beaucoup de composantes connexes, surtout pour des silhouettes en basses résolutions ($64 * 64$ pixels par exemple). Ceci vient du fait que le rendu d'une face n'est fait que si celle-ci occupe une surface suffisante dans un pixel. Afin de réduire cet artefact, nous utilisons une technique de double rendu associant les faces pleines et les

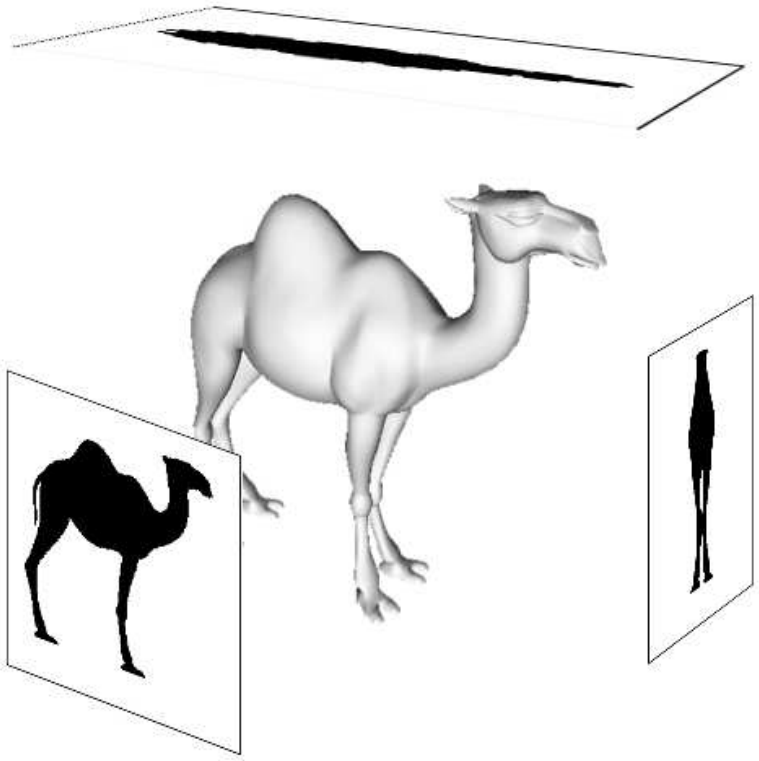
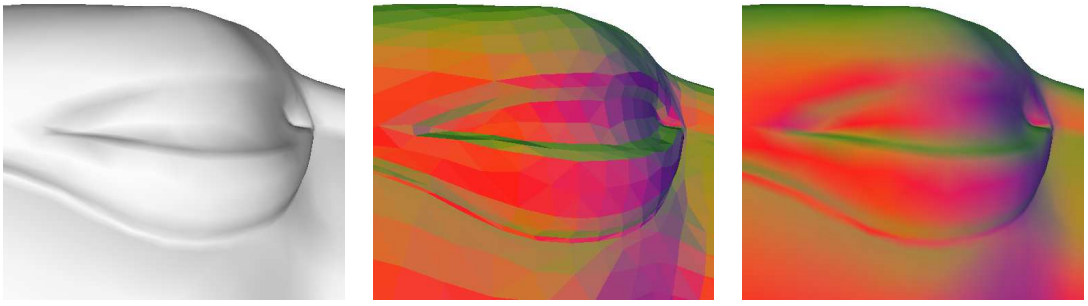


FIGURE 5.7 – *Extraction de trois vues du dromadaire précédent. Les silhouettes extraites sont alignées avec les axes principaux de l'objet 3D.*



(a) *Objet original* (b) *Carte de normales sans interpolation* (c) *Carte de normales avec interpolation*

FIGURE 5.8 – *Rendu d'un détail du dromadaire précédent. (a) représente le rendu classique, (b) le rendu avec la carte de normales sans interpolation et (c) le rendu final avec interpolation.*

arrêtes. Il s'agit en fait de superposer à l'affichage classique de l'objet 3D (faces pleines) chacune des arrêtes qui le compose. Ainsi, comme on peut le voir sur la figure 5.9, le nombre de composantes connexes est réduit laissant place à un meilleur rendu du modèle

3D. Des résultats plus approfondis sont présentés en annexe C de ce manuscrit.

(a) *Projection avec un rendu simple*

(b) *Projection avec un double rendu*

FIGURE 5.9 – *(a) montre le rendu d'un objet 3D en mode faces pleines uniquement, avec une résolution de* $64 * 64$ *pixels. (b) montre le rendu en mode faces pleines et fil de fer. On voit sur (b) que le nombre de composantes connexes est réduit grâce à cette nouvelle technique.*

5.2.3 Conclusion

Nous avons présenté dans cette partie les moyens techniques permettant la capture efficace des silhouettes d'un objet 3D et le rendu des informations de forme indispensables à l'indexation. La capture a été guidée par l'alignement effectué au chapitre 4, mais aussi par les propriétés intéressantes de répartition des sommets des polyèdres de Platon. Le rendu, quant à lui, s'appuie sur la richesse de la librairie graphique *OpenGL*. Les jeux de silhouettes 2D que nous obtenons après cette étape d'extraction sont aux nombres de :

- $48 * n$ pour une requête (voir section 4.3.4 du chapitre 4 à cause de l'ordonnancement des axes d'alignements).
- n pour les objets 3D de la base de données.

où n correspond au nombre de vues du polyèdre considéré. Il reste finalement à extraire l'information de forme contenue dans chacune de ces représentations 2D afin de décrire le modèle 3D.

5.3 Descripteur par ensemble de pixels (EP)

Afin d'indexer efficacement un objet 3D, il est nécessaire de caractériser sa forme avec une description adaptée. Dans cette thèse, nous avons choisi de baser celui-ci sur l'information contenue dans les projections en deux dimensions, du modèle, obtenues à l'étape précédente (voir section 5.2). Le premier descripteur que nous proposons cherche à décrire de manière simple la forme globale de la silhouette. Nous présentons donc dans cette partie, la nouvelle signature ainsi que sa transcription dans une représentation synthétique.

5.3.1 Descripteur de forme

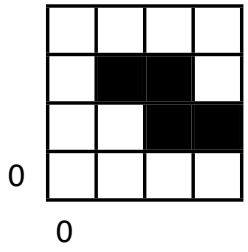
L'état de l'art présenté dans le chapitre 3 nous a montré que les méthodes "2D/3D" basées sur les régions telles que celles de Chen *et al.* [29][139][28] donnaient des résultats plus qu'intéressants et permettaient des requêtes en deux dimensions. Pour rester dans le même esprit, nous avons cherché à construire un nouveau descripteur avec un pouvoir discriminant raisonnable tout en garantissant un espace de stockage faible et un temps de comparaison qui l'est tout autant.

La description que nous avons choisie se base sur la notion de surface de la projection 2D. En effet, nous avons choisi pour décrire la forme de l'objet 2D une représentation ensembliste où les éléments $e = \{(x, y)\}$ de l'ensemble E sont les pixels noirs de la silhouette 2D monochrome (noir et blanche). L'ensemble E est alors défini comme suit :

$$E = \{(x, y) \in \mathbb{N}^2 \mid 0 \leq x < r,\ 0 \leq y < r, c(e) = Noir\} \tag{5.2}$$

avec $c(e)$ la couleur du pixel e définie par ses coordonnées x et y dans un espace de résolution de r^2 pixels. Une image vide, intégralement blanche dans notre cas, peut alors être considérée comme l'ensemble vide représenté par $E = \{\}$. Cette description offre l'avantage de pouvoir effectuer des opérations ensemblistes telles que l'intersection, l'union ou l'inclusion de deux ensembles $E_1 \in E$ et $E_2 \in E$ représentant deux silhouettes. Nous capturons donc la forme générale de l'objet par un ensemble de pixels dans E.

Cette représentation ensembliste permet de déduire très rapidement à partir d'une image 2D, de l'objet 3D, la forme de celui-ci. En effet, cette description stocke simplement l'information contenue par la juxtaposition des pixels formant la silhouette. La figure 5.10 montre une image décrite au moyen du descripteur proposé.



$$\Longrightarrow E_1 = \{(1, 2), (2, 1), (2, 2), (3, 1)\} \in E$$

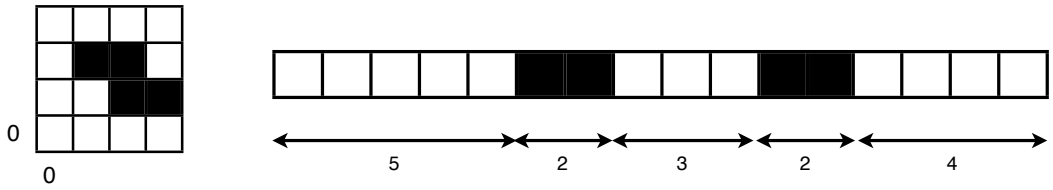
FIGURE 5.10 – *Image et sa traduction en terme ensembliste. Ici $x = 0, y = 0$ correspond au pixel en bas à gauche.*

Grâce à ce codage, nous avons vu que la forme de l'objet 3D est capturée dans le but de la comparer ultérieurement, au moyen d'opérations ensemblistes, avec d'autres objets. Cependant, il est nécessaire de synthétiser cette description mathématique pour réduire l'espace de stockage qu'elle nécessite.

5.3.2 Principe du codage synthétique

Afin de suivre les différentes étapes du processus d'indexation, nous devons transcrire la description choisie pour l'adapter à la machine. Le but étant de synthétiser la signature afin de garantir un espace de stockage faible favorisant une comparaison rapide. Avec la description choisie, une image intégralement noire aura $256 * 256 = 2^{16}$ éléments, ce qui n'est pas raisonnable, aussi bien pour le stockage que pour la comparaison, dans le cas de l'indexation d'une seule silhouette. Partant de la description ensembliste, basée sur la juxtaposition de pixels, nous avons donc cherché à synthétiser cette signature.

Les images que nous cherchons à encoder comportent, de par leur nature, un nombre restreint de composantes connexes représentées avec uniquement deux couleurs (plein ou vide). En représentant l'image, non plus avec deux, mais une dimension, telle que sur la figure 5.11, il nous est alors possible de décrire chaque projection 2D en caractérisant la succession des bandes noirs et blanches. Ce codage très simple permet de synthétiser la description, de la rendre adaptée à la machine et offre des moyens de comparaisons rapides (voir chapitre 6). Il nous suffit finalement de représenter chaque bande par un entier (16 bits si la résolution de la projection est $\leq 256 * 256$) codant sa longueur (voir figure 5.11 et algorithme 5.1).



(a) *Image originale* (b) *Codage synthétique du descripteur par ensemble de pixels*

FIGURE 5.11 – *Image originale (a) et sa traduction synthétique (b). Avec celui-ci, 5 valeurs suffisent à coder l'image au lieu de 16.*

L'intérêt majeur de cette transcription, outre son aspect synthétique, est sa propriété de compression sans perte offrant la pérennité des informations sélectionnées.

Pour conclure, les points forts du descripteur proposé sont :

- La formulation ensembliste permettant par la suite de comparer deux silhouettes avec des opérations telles que l'intersection, l'union ou l'inclusion.
- Le faible coût de stockage grâce à la transcription synthétique introduite.
- Le codage sans perte de la forme de la silhouette 2D.
- La vitesse d'extraction.

Algorithme 5.1 *Transcription synthétique d'une image monochrome*

```
1:  couleur = blanc
2:  nombre_pixels = 0
3:  Pour i = 0 à taille_image Faire
4:     Si pixel (i) = blanc Alors
5:        couleur_pixel = blanc
6:     Sinon
7:        couleur_pixel = noir
8:     Fin Si
9:     Si couleur_pixel == couleur Alors
10:       nombre_pixels = nombre_pixels + 1
11:    Sinon
12:       couleur = couleur_pixel
13:       descripteur_objet.ajouter (nombre_pixels)
14:       nombre_pixels = 1
15:    Fin Si
16: Fin Pour
```

5.4 Descripteurs multi-échelles de convexités/concavités (MCC)

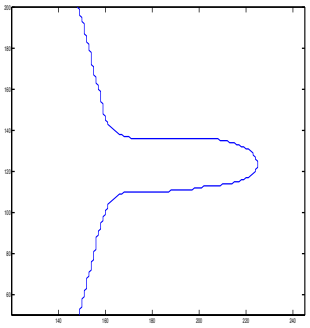
Toujours dans une optique "2D/3D", nous avons cherché à développer un nouveau descripteur basé sur les contours permettant une analyse pertinente de la forme. Nous utilisons pour cela une description proposée par Adamek et O'Connor [1][2][3] qui s'appuie, d'une part, sur les convexités/concavités et, d'autre part, sur une approche multi-échelles. Cette section introduit donc le calcul des convexités/concavités sur les contours 2D, l'approche multi-échelles et enfin le codage synthétique approprié à la machine.

5.4.1 Informations de convexités/concavités des contours

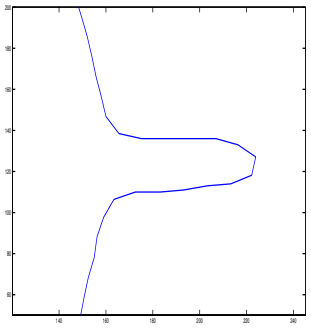
Comme pour la méthode précédente, ce sont les conclusions sur l'état de l'art présenté dans le chapitre 3 qui nous ont dirigé vers les approches dites "2D/3D". Plus que leur fort pouvoir discriminant, c'est leur grande flexibilité coté requêtes qui nous a particulièrement intéressé. Pour ce nouveau descripteur, nous avons cherché à garantir une bonne robustesse tout en offrant une grande variété des moyens d'interrogations de la base. La description utilisée, basée sur les convexités et les concavités, permet à la fois de rechercher un modèle avec un objet 3D mais aussi avec une photo ou une esquisse au trait.

La première étape nécessaire à l'analyse du contour est son extraction. Pour cela nous utilisons un algorithme simple de suivi de contour sur la plus grande partie connexe, s'il en existe plusieurs (voir section 5.2.2).

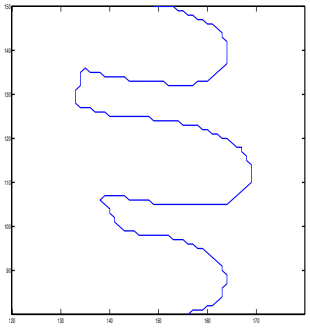
Après cette première étape, nous devons rééchantillonner le contour C en vue de cal-



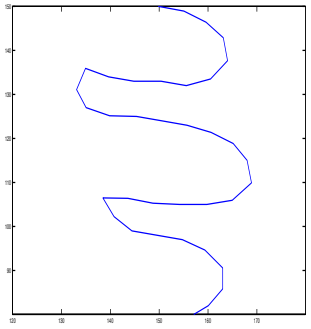
(a) *Détail d'un contour d'avion original*



(b) *Détail d'un contour d'avion discrétisé*



(c) *Détail d'un contour de main original*



(d) *Détail d'un contour de main discrétisé*

FIGURE 5.12 – *(a) et (b) représentent respectivement un détail d'un contour original d'avion et sa version discrétisée en $N = 100$ points, tandis que (c) et (d) représentent respectivement un détail d'un contour original de main et sa version discrétisée.*

culer les informations sur un nombre identique de points pour chaque silhouette. Afin de rester le plus proche possible de la projection initiale, nous calculons un ensemble de N points en suivant l'abscisse curviligne. Nous obtenons alors une représentation composée de N points discrets pour chaque contour (voir figure 5.12). Cette discrétisation permet, d'autre part, de lisser sommairement le contour C si le nombre de points est réduit. Dans ce sens, nous avons choisi $N = 100$ pour notre implémentation. Les mesures de convexités et de concavités pour l'ensemble des représentations 2D devant être dans un même ordre de grandeur, nous normalisons les N points en taille et en position. Cette normalisation utilise les moments géométriques et un facteur d'échelle β transformant un point p en un point p' comme suit (dans notre implémentation nous fixons $\beta = 500$) :

$$p'_x = \underbrace{\frac{\beta}{m_{00}}}_{\text{échelle}} \underbrace{(p_x - \frac{m_{10}}{m_{00}})}_{\text{position}} \text{ et } p'_y = \frac{\beta}{m_{00}}(p_y - \frac{m_{01}}{m_{00}}) \tag{5.3}$$

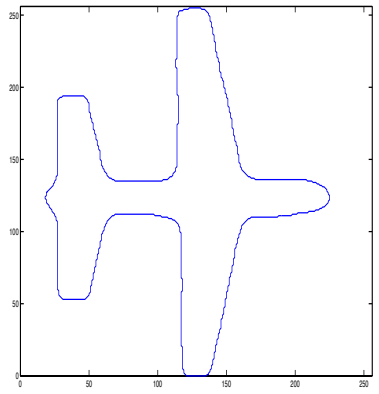
avec :

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \tag{5.4}$$

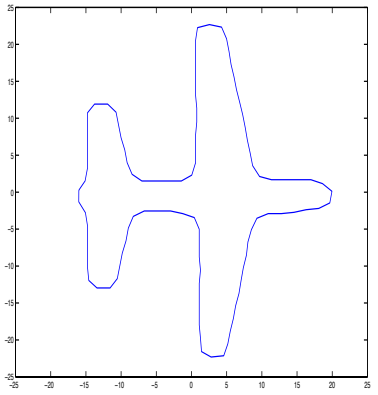
où :

$$f(x,y) = \begin{cases} 1 & \text{si le pixel est noir} \\ 0 & \text{sinon} \end{cases} \tag{5.5}$$

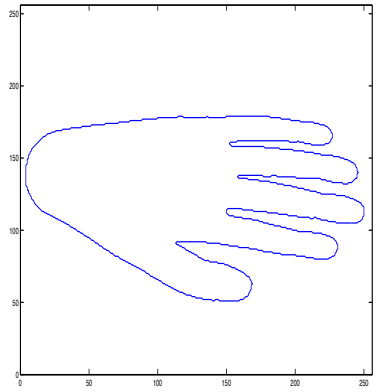
La figure 5.13 montre la transformation du contour C original en un ensemble de $N = 100$ points normalisés en position et en échelle.



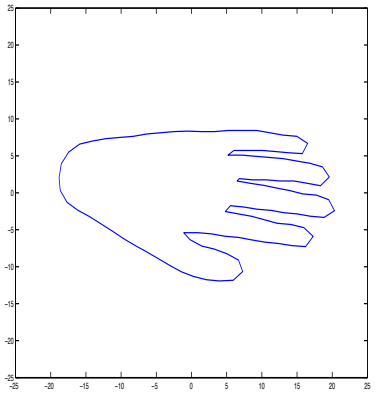
(a) *Contour d'avion original*



(b) *Contour d'avion normalisé*



(c) *Contour de main original*



(d) *Contour de main normalisé*

FIGURE 5.13 – *(a) et (b) représentent respectivement un contour original d'avion et sa version normalisée en position et en échelle, tandis que (c) et (d) représentent respectivement un contour original de dromadaire et sa version normalisée.*

Les prétraitements indispensables, au bon déroulement de l'algorithme d'analyse du

contour, ayant été présentés, nous pouvons introduire la méthode de calcul des informations de convexités et de concavités. Cette mesure a pour but de caractériser un contour par une valeur de convexité/concavité calculée en chaque point p du contour C, tel que visible sur la figure 5.14, où la couleur et la longueur des vecteurs donnent cette information pour un sous ensemble de points. Afin d'extraire cette information, nous avons choisi la méthode d'Adamek et O'Connor [1][2][3] basée sur celle de Mokhtarian et Mackworth [72] utilisant un noyau Gaussien pour filtrer le contour.

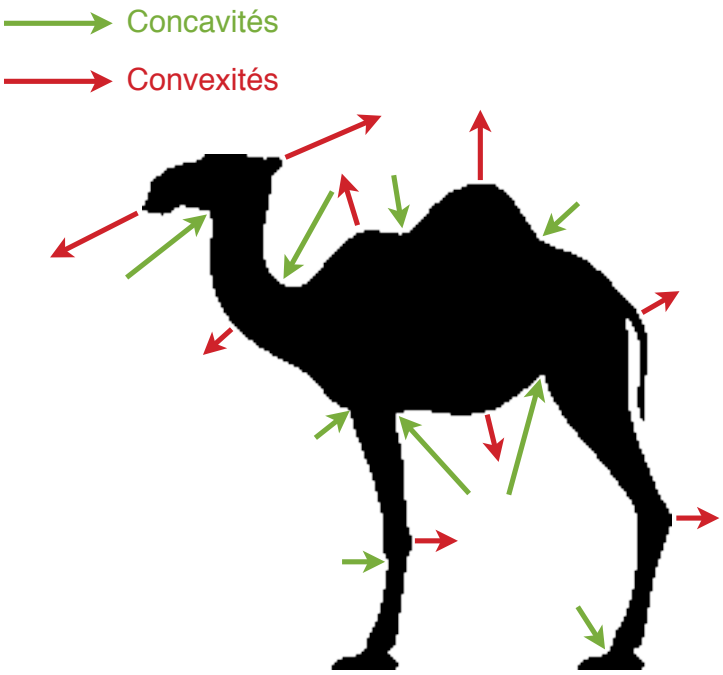


FIGURE 5.14 – *Représentation des informations de concavités (en vert) et de convexités (en rouge) que l'on souhaite extraire pour chaque silhouette de l'objet 3D.*

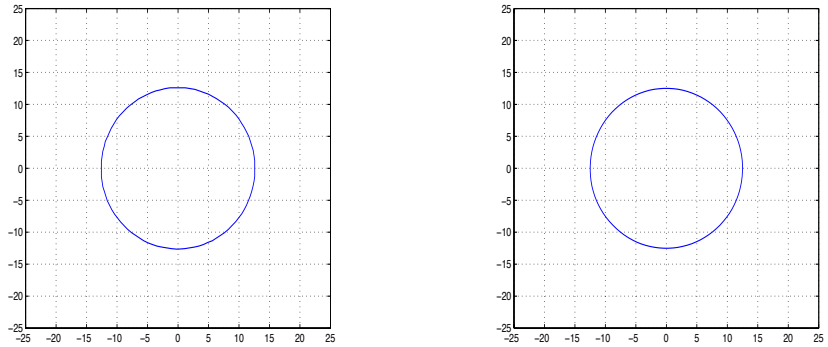
Étant donné un contour C discrétisé en N points, définissons alors la fonction suivante :

$$p' : C(p') = (x(p'), y(p')) \text{ où } p' \in [0..100] \tag{5.6}$$

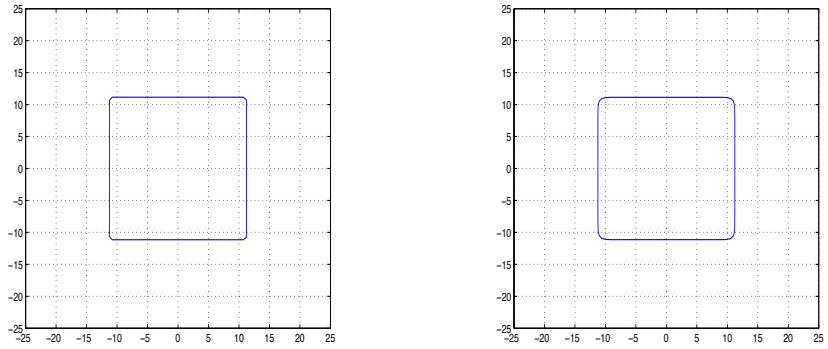
Le filtrage du contour revient alors à effectuer un produit de convolution entre cette fonction et un noyau Gaussien ϕ comme suit :

$$\begin{aligned} x_1(p') &= \int x(p')\phi(t-p')dt \\ y_1(p') &= \int y(p')\phi(t-p')dt \\ \phi(t) &= \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2) \end{aligned} \tag{5.7}$$

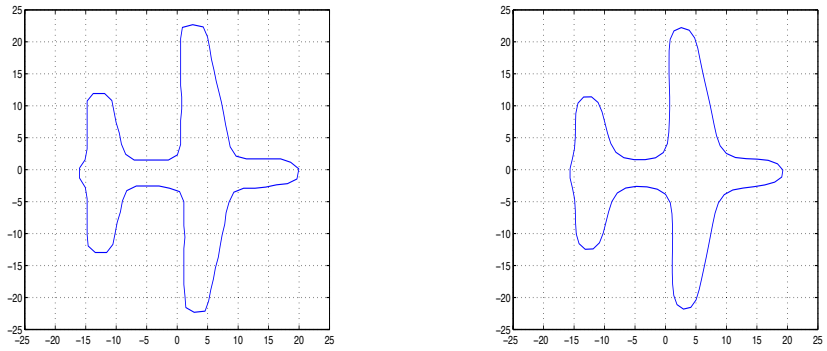
avec $t \in [-3..3]$. Ainsi, une version plus lissée du contour original C est obtenue, elle est notée C_1 avec $C_1(p') = (x_1(p'), y_1(p'))$ (voir figure 5.15). Il reste cependant à comparer ces deux représentations afin d'en extraire l'information de forme.



(a) *Opération de filtrage sur un cercle*



(b) *Opération de filtrage sur un cube*



(c) *Opération de filtrage sur un avion*

FIGURE 5.15 – *Résultat de l'opération de filtrage sur trois contours. (a), (b) et (c) représentent respectivement un cercle, un cube et un avion.*

La mesure de convexités/concavités proposée par l'auteur s'obtient simplement en calculant le déplacement du contour entre C et C_1. Alors, la mesure de convexités/concavités en un point revient à calculer la distance Euclidienne d entre les coordonnées du point $C_1(p')$ et $C(p')$, soit :

$$d(p') = k\sqrt{(x_1(p')) - x_0(p'))^2 + (y_1(p')) - y_0(p'))^2} \tag{5.8}$$

avec :

$$k = \begin{cases} 1 & \text{si le point } C_1(p') \text{ est à l'intérieur de } C \\ -1 & \text{sinon} \end{cases} \tag{5.9}$$

La mesure correspond alors au déplacement du point avant et après filtrage. Plus la distance est importante plus la convexité ou concavité est importante. On peut noter que la différence entre ces deux notions est apportée par l'équation 5.9 où un signe est donné à cette distance, en fonction du sens du déplacement. Le contour au point p' est considéré comme convexe si le point filtré $C_1(p')$ est à l'intérieur du contour fermé C, dans ce cas la mesure est positive. En revanche, le contour au point p' est considéré comme concave si le point filtré $C_1(p')$ est à l'extérieur du contour fermé C, dans ce cas la mesure est négative. La figure 5.17 montre différents exemples de mesures de convexités et de concavités obtenus sur des silhouettes 2D.

> **Remarque 5.3.** *Pour un ensemble restreint de projections, l'implémentation directe de l'équation 5.8 peut entraîner une discontinuité des valeurs à la frontière entre convexité et concavité. En effet, pour des contours très lisses, le déplacement entre deux filtrages peut se faire dans une direction parallèle plus que perpendiculaire (voir figure 5.16). Dans ce cas la mesure mise en place est erronée et nécessite une amélioration. En conséquence, il faudrait obtenir le déplacement entre deux filtrages consécutifs en calculant la distance entre $C_1(p')$ et C. Ce calcul étant coûteux, nous l'effectuons uniquement entre $C_1(p')$ et la portion de contour $\langle C(p'^{-1}),\ C(p'^{+1})\rangle$ où p'^{-1} et p'^{+1} représente respectivement le prédécesseur et le successeur de p'.*

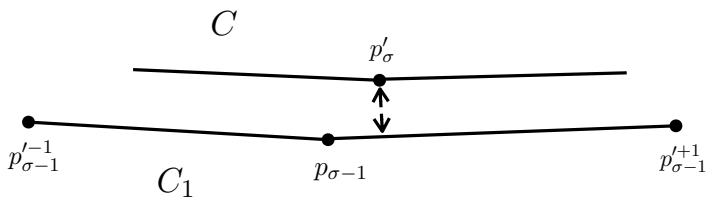
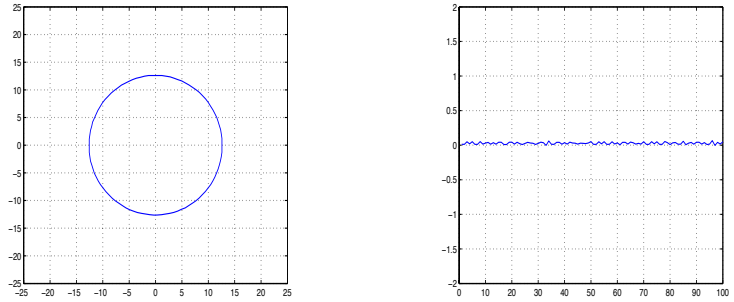
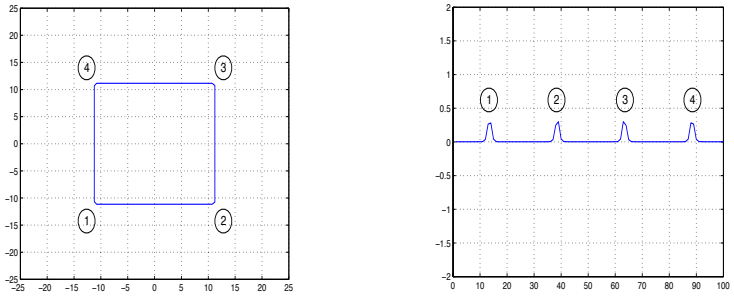


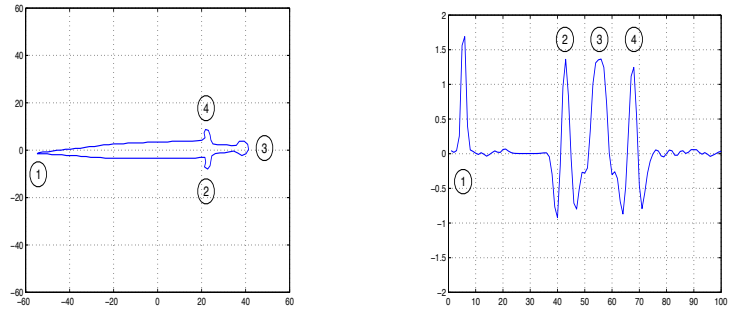
FIGURE 5.16 – *Exemple d'un filtrage sur un contour à la frontière entre convexité et concavité entrainant un déplacement parallèle et non perpendiculaire.*



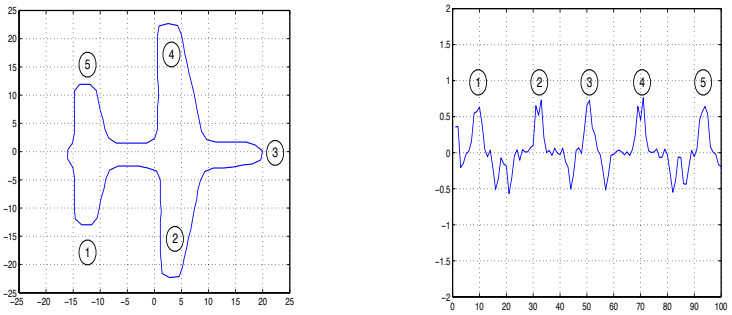
(a) *Convexités/concavités d'un cercle*



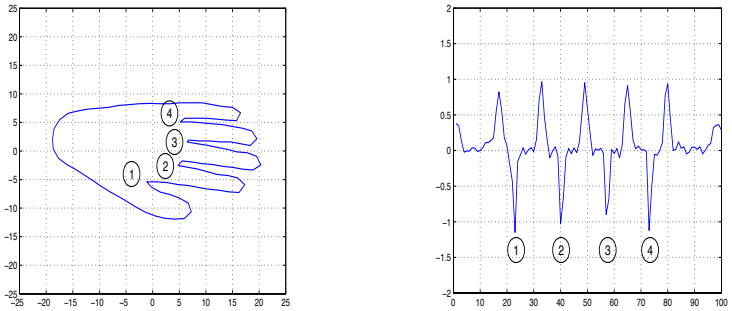
(b) *Convexités/concavités d'un carré*



(c) *Convexités/concavités d'une épée*



(d) *Convexités/concavités d'un avion*



(e) *Convexités/concavités d'une main*

FIGURE 5.17 – *Mesures de convexités et de concavités obtenues pour cinq contours. (a), (b), (c), (d) et (e) représentent respectivement un cercle, un cube, une épée, un avion et une main*

5.4.2 Approche multi-échelles

Nous avons introduit dans la partie précédente les outils permettant le calcul d'une information de convexité/concavité sur un contour 2D. Afin d'améliorer la pertinence de cette notion, les auteurs originaux, Adamek et O'Connor [1][2][3], proposent une approche multi-échelles permettant de calculer ces mesures après un filtrage plus ou moins fort.

Pour permettre un traitement sur plusieurs échelles, plusieurs filtrages successifs doivent être appliqués au contour original C_0. Pour cela, il suffit de transformer le produit de convolution de l'équation 5.7 en utilisant un noyau Gaussien ϕ_σ de taille croissante avec $\sigma \in \{1,\ 2,\ ...,\ \sigma_{max}\}$. Ce qui nous donne :

$$\begin{aligned} x'_\sigma(p') &= \int x(p')\phi_\sigma(t-p')dt \\ y'_\sigma(p') &= \int y(p')\phi_\sigma(t-p')dt \\ \phi_\sigma(t) &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{1}{2\sigma^2}t^2) \end{aligned} \tag{5.10}$$

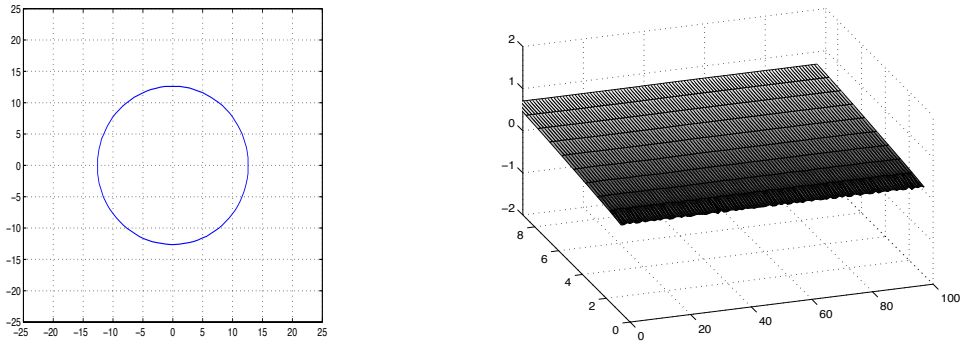
où $tin[-3\sigma..3\sigma]$. Avec cette nouvelle notation, nous considérons C_0 le contour original C. Ainsi, le contour obtenu est de plus en plus lissé tandis que σ croît pour parvenir finalement à un contour entièrement convexe – un cercle – lorsque σ est grand. Dans notre implémentation, nous fixons $\sigma_{max} = 10$ afin d'avoir pour chacun des N points du contour 10 mesures de convexités/concavités. Celles-ci sont calculées en reprenant les équations 5.8 et 5.9 comme suit :

$$d_\sigma(p'_\sigma) = k\sqrt{(x_\sigma(p')) - x_{\sigma-1}(p'))^2 + (y_\sigma(p')) - y_{\sigma-1}(p'))^2} \tag{5.11}$$

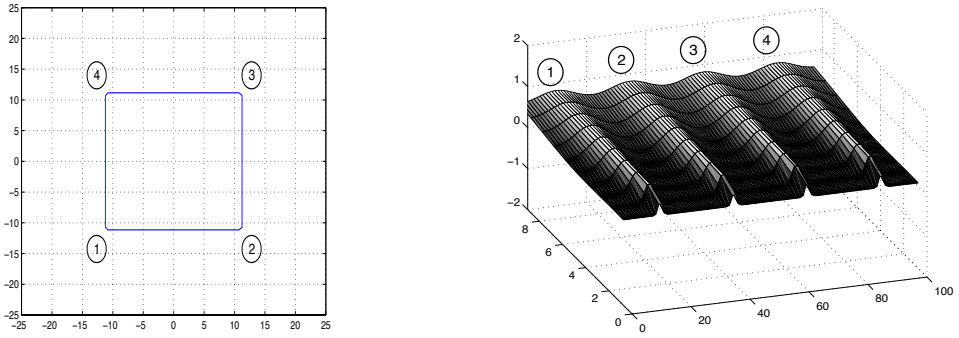
avec :

$$k = \begin{cases} 1 & \text{si le point } C_\sigma(p') \text{ est à l'intérieur de } C_{\sigma-1} \\ -1 & \text{sinon} \end{cases} \tag{5.12}$$

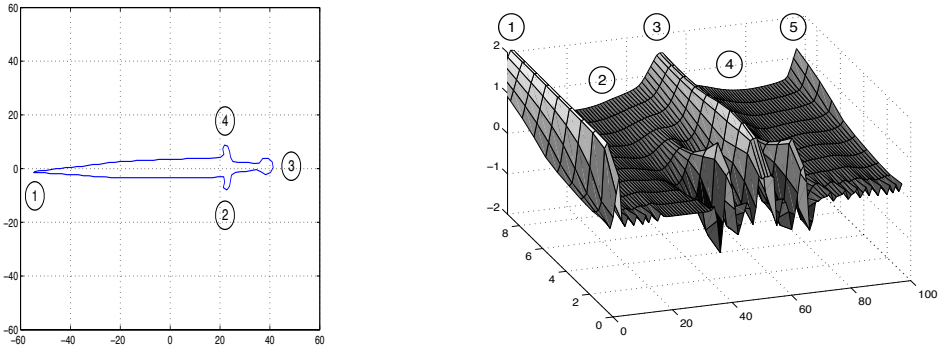
La figure 5.18 montre les informations calculées pour 10 échelles tandis que la figure 5.19 compare les mesures de convexités/concavités pour quatre silhouettes différentes d'avion.



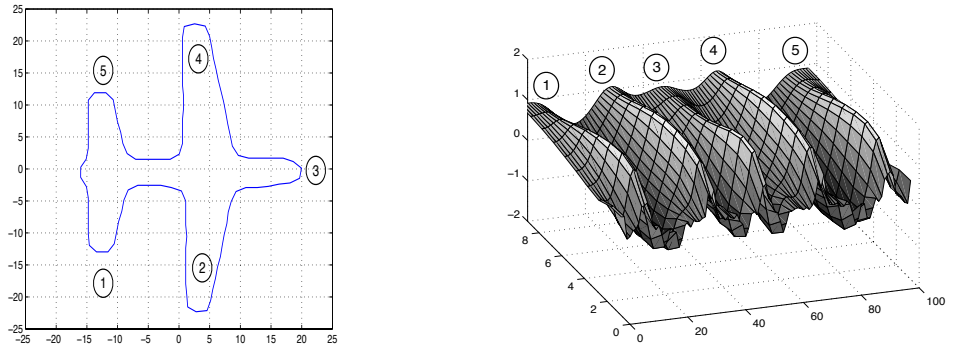
(a) *Convexités/concavités multi-échelles d'un cercle*



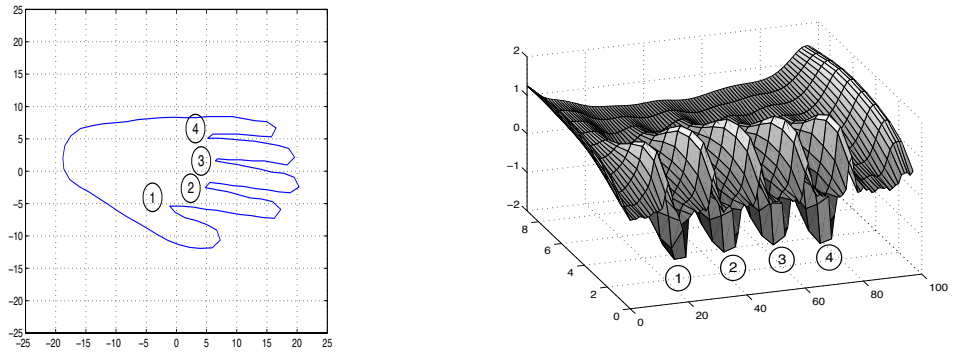
(b) *Convexités/concavités multi-échelles d'un carré*



(c) *Convexités/concavités multi-échelles d'une épée*



(d) *Convexités/concavités multi-échelles d'un avion*



(e) *Convexités/concavités multi-échelles d'une main*

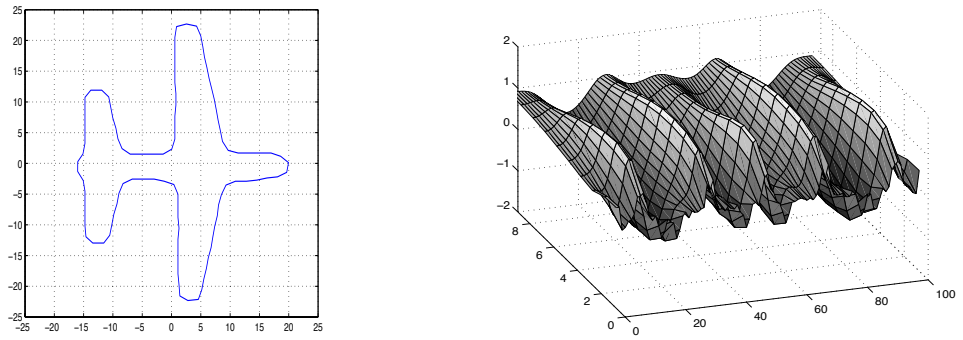
FIGURE 5.18 – *Mesure de convexités et de concavités obtenues pour cinq contours avec 10 échelles. (a), (b), (c), (d) et (e) représentent respectivement un cercle, un cube, une épée, un avion et une main*

Finalement, la description choisie, basée sur les informations de convexités/concavités, nous permet d'obtenir pour chaque silhouette un ensemble de $N * \sigma_{max} = 100 * 10 = 1000$ valeurs. Ces informations permettent de caractériser la forme du contour dans le but d'indexer nos objets 3D et d'effectuer ensuite une recherche rapide parmi eux. Cependant, le nombre de mesures de convexités/concavités calculées et peu synthétiques. Il est donc nécessaire de les compresser afin de limiter l'espace de stockage nécessaire.

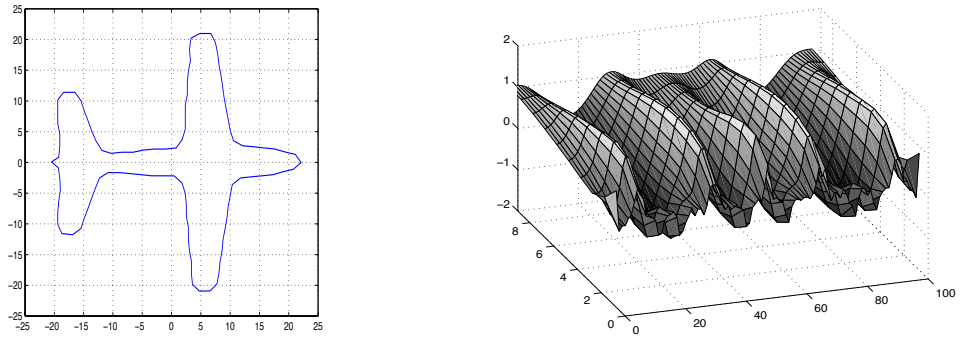
5.4.3 Codage synthétique de l'information

Afin de suivre les étapes indispensables au bon déroulement du processus d'indexation, nous devons synthétiser la signature. Dans le cas des mesures de convexités/concavités, même si la description est bien adaptée à la machine, elle doit cependant subir une compression afin de limiter l'espace de stockage et éventuellement le temps de recherche.

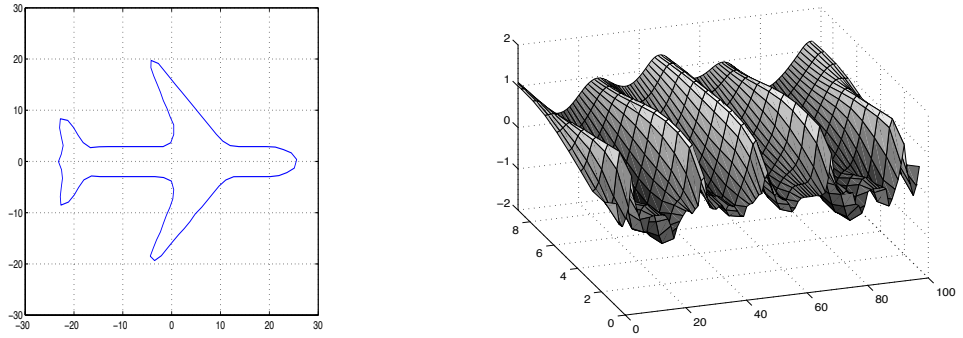
Étant donné que cette compression doit garantir une perte minimum des informations de forme, Adamek et O'Connor [1][2][3] proposent d'utiliser une transformée en cosinus



(a) *Convexités/concavités multi-échelles de l'avion 1*



(b) *Convexités/concavités multi-échelles de l'avion 2*



(c) *Convexités/concavités multi-échelles de l'avion 3*

FIGURE 5.19 – *Comparaisons des informations de convexités/concavités calculées pour différentes silhouettes d'avion. On voit sur cette figure que l'on retrouve bien les points caractéristiques du contour, à savoir, le nez, les ailes et la queue des avions.*

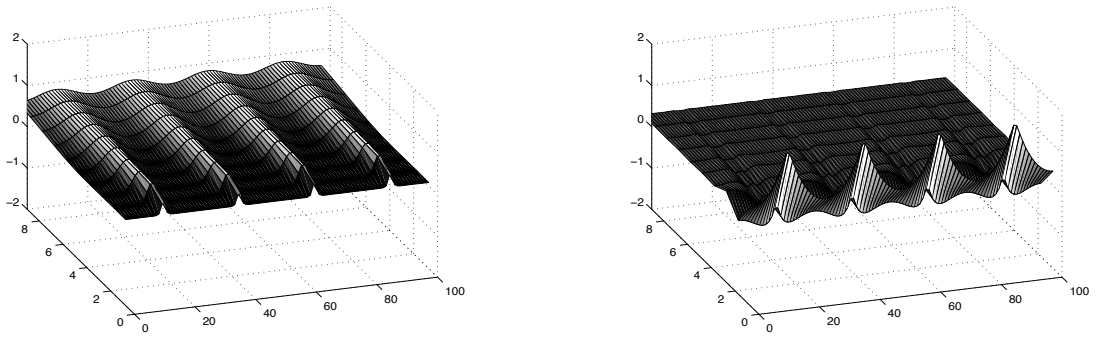
discrète "DCT" appliquée sur la dimension de filtrage σ :

$$DCT_k = \alpha \sum_{\sigma=0}^{\sigma_{max}-1} d_\sigma \cos\left(\frac{\pi}{2\sigma_{max}}(2\sigma+1)k\right) \tag{5.13}$$

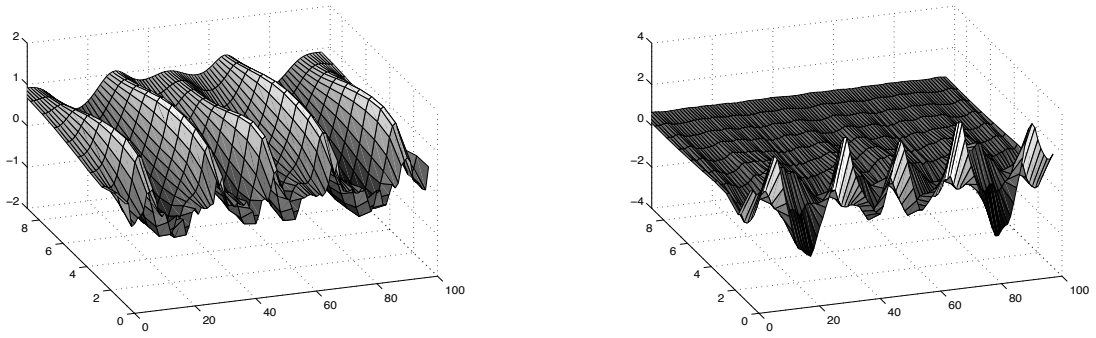
où $DCT_k,\ 0 \leq k < K$ représente la compression de la donnée d, σ la taille du filtre, $\sigma_{max} = 10$ et avec :

$$\alpha = \begin{cases} \sqrt{\frac{1}{N}} & \text{si } k = 0 \\ \sqrt{\frac{2}{N}} & \text{sinon} \end{cases} \tag{5.14}$$

L'intérêt majeur de cette transformée est sa capacité de "regroupement" qui place l'essentiel de l'information dans les cœfficients basses fréquences. De plus, elle permet de traiter des données discrètes et génère des cœfficients réels, donc faciles à stocker. Ce sont donc ceux-ci que nous allons stocker, offrant maintenant une taille de $N*K = 100*4 = 400$ valeurs si on fixe $K = 4$. La figure 5.20 montre deux exemples de cette compression par "DCT".



(a) *Convexités/concavités multi-échelles avec et sans "DCT" d'un carré*



(b) *Convexités/concavités multi-échelles avec et sans "DCT" d'un avion*

FIGURE 5.20 – *Exemples d'une compression par "DCT" avec $K = 10$ pour un carré (a) et un avion (b).*

Afin de réduire encore la taille de la description, nous proposons un second codage basé sur les histogrammes (nous nommons ce descripteur MCC-H). Dans celui-ci, les informations de convexités/concavités d'une échelle donnée sont synthétisées en un histogramme de $H = 10$ composantes. Ainsi, cette signature contient $K * H = 10 * 10 = 100$ valeurs.

Pour conclure, la nouvelle description proposée permet de décrire efficacement un contour 2D et donc un objet 3D à travers l'ensemble de ses projections. Les points forts de cette signature sont :

- L'utilisation des informations présentes uniquement sur le contour permettant, de ce fait, une approche par requêtes 2D.
- L'approche par convexités et concavités offrant une description pertinente de la forme (voir chapitre 7).
- La mesure multi-échelles qui permet d'obtenir une description plus fine des silhouettes.
- Les compressions des signatures par "DCT" et par histogrammes.

5.5 Descripteur par cartes de normales (CN)

Pour cette troisième description, nous avons voulu explorer la piste des approches dites "2.5D/3D". Le nom de cette famille de méthodes vient du fait qu'elles exploitent des informations 3D, obtenues par projection, qui ne sont pas présentes dans une image 2D, telle que la profondeur ou des informations de surface par exemple. C'est en particulier les approches par cartes de profondeurs proposées par Vranic [134] et améliorées par Chaouch et Verroust-Blondet [25][26][24] qui nous ont intéressées. Dans cette nouvelle description, nous proposons de décrire la forme d'un objet 3D par l'orientation de son maillage. Pour cela nous présentons l'étape d'extraction de ces informations ainsi que leur codage synthétique.

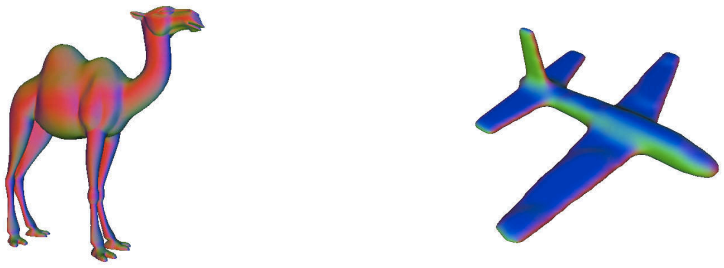
5.5.1 Extraction des normales

Afin de décrire la forme du modèle 3D étudié, nous proposons de choisir comme description l'orientation de la surface. Il s'agit en quelque sorte de caractériser la direction du maillage de l'objet 3D de manière continue. Étant donné que nous avons choisi une approche "2.5D/3D", nous devons exporter cette information dans une image 2D. Dans ce sens, nous devons projeter non plus la silhouette de l'objet 3D mais les normales de sa surface. La première étape pour extraire ce nouveau descripteur est donc le calcul des normales en chacun des points du maillage.

Afin d'obtenir une information de direction pour chaque point du maillage nous utilisons l'interpolation offerte par la bibliothèque de rendu *OpenGL* comme présenté dans la section 5.2.2. Ainsi, il nous suffit de calculer les normales uniquement pour les sommets du maillage. La normale d'un sommet est donnée par la somme des normales de ses faces adjacentes. La normale $\mathcal{N}_i$ d'une de ces faces, $\mathcal{F}_i$ composée de trois sommets A_i, B_i et C_i, est donnée par :

$$\mathcal{N}_i = \frac{\overrightarrow{A_iC_i} \wedge \overrightarrow{A_iB_i}}{||\overrightarrow{A_iC_i} \wedge \overrightarrow{A_iB_i}||} \tag{5.15}$$

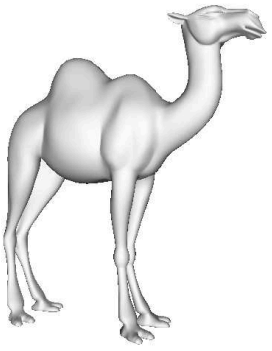
Ainsi, nous obtenons un maillage coloré en fonction des orientations des faces telles que sur la figure 5.21. La projection sur un plan 2D nous donne alors une carte de normales où chaque pixel donne l'information de forme de la surface 3D en un point précis. La figure 5.22 présente un exemple de cartes de normales obtenues à partir d'un objet 3D.



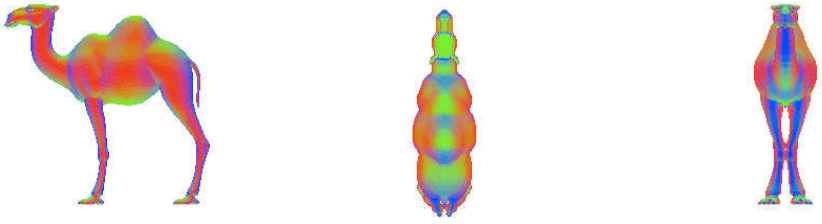
(a) *Un dromadaire coloré selon l'orientation de ses normales*

(b) *Un avion coloré selon l'orientation de ses normales*

FIGURE 5.21 – *Exemple de deux objets 3D colorés selon l'orientation de leur maillage. (a) et (b) représentent respectivement un dromadaire et un avion.*



(a) *Objet original*



(b) *Première carte de normales*

(c) *Seconde carte de normales*

(d) *Troisième carte de normales*

FIGURE 5.22 – *Cartes de normales de l'objet (a) capturées dans la direction des trois axes du repère* $\mathbb{R}^3$. *(b) représente le point de vue* x^+, *(c)* y^+ *et (d)* z^+.

Même si cette description permet de caractériser la forme globale d'un objet 3D, elle est peu synthétique. En effet, il est indispensable de réduire l'espace de stockage qui est ici de $r^2 = 256 * 256 = 2^{16}$, où r est la résolution de la carte de normales. En effet, c'est en général l'aspect synthétique d'une représentation qui permet un archivage ainsi qu'une recherche efficace.

5.5.2 Codage synthétique des cartes de normales

Les descripteurs extraits lors du processus d'indexation nécessitent, de par leur aspect persistant, d'être compact. Dans ce sens, il est nécessaire de compresser la caractérisation de la forme, présentée ci-dessus et basée sur une description utilisant l'orientation du maillage. Pour réaliser cette tâche nous avons choisi d'utiliser une transformée de Fourier discrète "DFT", donnée en une dimension par :

$$V(k) = \sum_{n=0}^{N-1} v(n) \exp(-\frac{2i\pi}{N} kn) \tag{5.16}$$

où, N représente la taille des données et $v(n)$ la donnée d'indice n. Dans notre cas, les données à compresser sont trois images. Une pour la composante rouge, une pour la verte et une dernière pour la bleue. Pour cela, nous utilisons une "DFT 2D" qui se calcule comme suit :

$$V_2(k) = V(V(k)') \tag{5.17}$$

où $V(k)'$ correspond à la transposée de $V(k)$. Dans notre implémentation, nous utilisons une variante appelée Transformée de Fourier Rapide "FFT" à travers la librairie *fftw*[2], ce qui permet d'améliorer grandement la rapidité de calcul. La figure 5.23 montre des exemples de transformées sur différentes cartes de normales. Afin de compresser les données, nous calculons le module de la transformée et nous plaçons les basses fréquences au centre, comme représenté sur la figure 5.23. Enfin, en se basant sur la propriété de symétrie axiale de la transformée de Fourier, vérifiée lorsque les données sont réelles, nous sélectionnons un sous-ensemble de valeurs situées dans un demi-cercle de rayon r comme caractérisation de la carte de normales. En fixant $r = 20$, nous obtenons un codage synthétique de la description tout en garantissant une caractérisation suffisante de la carte de normales, car basée sur les basses fréquences.

5.5.3 Prise en compte de l'ordonnancement des axes

Dans la phase de prétraitement que nous présentons dans le chapitre 4, nous utilisons un réordonnancement des axes afin de contourner des erreurs lors de l'estimation de l'alignement. Cependant, à cause de ce traitement, les cartes de normales d'un même objet 3D après un réordonnancement d'axes ne sont pas comparables aux cartes de normales de l'objet original. En effet, comme on peut le voir sur la figure 5.24, les couleurs des cartes de normales changent après un réordonnancement. Cela vient du fait que le calcul

2. http://www.fftw.org/

(a) *Transformé de Fourier de la composante rouge d'une carte de normale d'un avion*

(b) *Transformé de Fourier de la composante rouge d'une carte de normale d'une main*

(c) *Transformé de Fourier de la composante rouge d'une carte de normale d'un dromadaire*

FIGURE 5.23 – *Transformée de Fourier discrète d'un ensemble d'images, en niveaux de gris, représentant des cartes de normales. Nous représentons ici le module avec les basses fréquences situées au centre de l'image. (a), (b) et (c) représentent respectivement les composantes rouge d'un avion, d'une main et d'un dromadaire.*

des normales n'est effectué qu'une seule fois lors du chargement du modèle 3D, accélérant l'extraction du descripteur, mais introduisant cette erreur.

Afin de résoudre ce problème, nous introduisons une table de correspondance des différentes composantes Rouge, Verte et Bleue pour chacun des réordonnancements d'axes. Le tableau 5.1 présente cette table de correspondance pour les six possibilités (l'ordonnancement numéro 1 correspond à la pose originale).

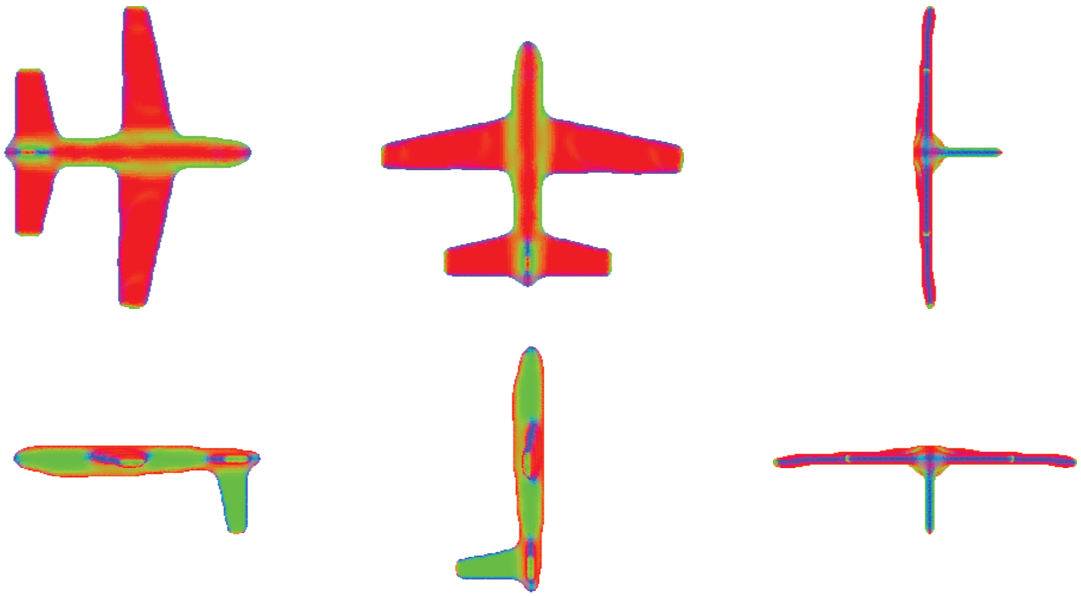


FIGURE 5.24 – *Cartes de normales de la même vue d'un objet 3D après réordonnancement des axes du repère* $\mathbb{R}^3$*. Chaque image représente un ordonnancement différent. Il est clairement visible que ces cartes de normales ne sont pas comparables tels quels car les composantes Rouge, Verte et Bleue ne représentent pas le même axe.*

Ordonnancement	*Rouge (R)*	*Vert (V)*	*Bleu (B)*
1	**R**	**V**	**B**
2	**R**	**B**	**V**
3	**B**	**V**	**R**
4	**V**	**R**	**B**
5	**V**	**B**	**R**
6	**B**	**R**	**V**

TABLE 5.1 – *Table de correspondance des composantes Rouge, Verte et Bleue en fonction du réordonnancement des axes du repère* $\mathbb{R}^3$*.*

Pour conclure, la nouvelle description basée sur une approche "2.5D/3D" caractérise la forme grâce à l'orientation du maillage. Les points forts de cette signature sont :

- L'utilisation des normales qui offre une information de forme riche.
- La sélection des informations basses fréquence grâce à la "DFT".
- L'aspect synthétique du codage.

5.6 Conclusion

Dans ce chapitre, nous avons traité l'un des points majeurs de l'indexation, à savoir la description des données. Dans notre cas, afin de caractériser efficacement les objets 3D tout en garantissant une recherche facile pour l'utilisateur, nous avons choisi les approches basées sur des projections en deux dimensions. En effet, en choisissant ce type de méthode, nous permettons à l'utilisateur de faire des requêtes à partir d'objets 3D, mais aussi de photos ou de dessins au trait.

À cause de la nature des données que nous avons choisi d'indexer, à savoir des objets 3D, nous avons dû mettre en place des techniques particulières pour extraire les caractéristiques de forme de chacun d'eux. Dans ce sens, nous avons introduit dans les sections 5.2 et 5.2.2 des méthodes nouvelles pour extraire des informations de forme en deux dimensions.

Le processus d'indexation étant basé sur une description pertinente des données, nous avons ensuite introduit trois nouvelles descriptions basées sur les projections d'un modèle 3D. Le premier est basé sur la notion de pixel tandis que la seconde caractérise les contours des silhouettes et que la dernière décrit l'orientation de la surface de l'objet. L'ensemble de ces représentations est associé à un codage synthétique et pertinent adapté au stockage ainsi qu'à la recherche. Ce chapitre clôture donc la fin de la partie "hors-ligne" du processus d'indexation et ouvre la voie à la partie "en-ligne" indispensable à l'utilisateur pour rechercher un objet 3D spécifique.

CHAPITRE 6

Recherche d'objets 3D

Résumé *: Ce chapitre introduit la dernière contribution de notre thèse portant sur la recherche d'objets 3D. Nous y proposons un processus complet de recherche basé sur trois types de requêtes. Nous introduisons aussi deux optimisations permettant d'allier robustesse et vitesse. Finalement, nous proposons un prototype fonctionnant sur terminal mobile.*

6.1 Introduction

La conception du moteur de recherche est l'une des étapes cruciales dans la création de tout processus d'interrogation d'une base de données. En effet, c'est grâce à celui-ci qu'il est possible de répondre au besoin de l'utilisateur, à savoir la recherche d'un objet 3D précis. Contrairement au processus d'indexation, dont le fonctionnement est invisible bien qu'indispensable, le moteur de recherche est le seul moyen donné à l'utilisateur pour dialoguer avec la bibliothèque d'objets 3D. À cause de ce rôle d'interface homme-machine, il est nécessaire de faciliter la prise en main de ce processus d'interrogation en le rendant rapide, robuste en terme de résultats et si possible ergonomique.

Lors de la conception d'un tel système de recherche, basé sur la forme, il est nécessaire de comparer la requête de l'utilisateur avec l'ensemble des modèles 3D stocké au préalable dans la bibliothèque. Afin d'accélérer cette tâche complexe, nous utilisons le processus d'indexation proposé dans le chapitre 5. Ainsi, après une description de la requête de l'utilisateur par un processus similaire, il n'est plus nécessaire de la comparer avec chaque objet 3D de la base de données, mais seulement avec leur description synthétique. Cependant, même si cette étape offre des solutions efficaces pour extraire les caractéristiques importantes de chaque objet, en terme de forme, elle ne propose pas de solution quant à leur comparaison.

La tâche principale du moteur de recherche est donc de proposer des mesures de dissimilarité pertinentes permettant d'évaluer la dissemblance entre deux descripteurs. Cependant, l'interrogation de la bibliothèque se faisant "en ligne", elle doit s'appuyer sur des optimisations permettant l'accélération de la recherche ainsi qu'une meilleure robustesse, en terme de précision des résultats. Finalement, il faut tenir compte de la place centrale de l'utilisateur, dans ce processus, en lui offrant des outils facilitant sa recherche d'objets 3D. Les contributions apportées dans ce chapitre ont fait l'objet de publications dans [75][77].

Nous débutons ce chapitre par la section 6.2 présentant le fonctionnement global du

processus de recherche d'objets 3D. Nous introduisons ensuite, dans la section 6.3, les différents types de requêtes prisent en compte par notre moteur de recherche qui vont des "objets 3D" aux "dessins au trait" en passant par les "photos". Nous proposons, dans un même temps, leur analyse et description à l'aide de l'une des approches proposées dans le chapitre 5. La comparaison de deux signatures n'ayant pas encore été abordée, nous définissons dans la section 6.4 les mesures de dissimilarité associées à nos différents descripteurs ainsi que la démarche employée pour comparer une requête à l'ensemble de la base de données. Nous enchainons, dans la section 6.5, avec les différentes approches mises en place pour optimiser la recherche dans une base de données 3D. Nous nous attardons en particulier sur deux approches permettant d'optimiser la pertinence grâce à une approche multi-descripteurs, pour la première, et d'accélérer le processus de recherche au moyen d'un élagage précoce, pour la seconde. Finalement, la section 6.6 présente les prototypes qui permettent l'indexation d'objets 3D, mais aussi leur recherche, aux moyens d'interfaces ergonomiques et confortables tournées vers l'utilisateur.

6.2 Fonctionnement du processus de recherche

L'élaboration d'un processus de recherche d'objet 3D est une tâche complexe. En effet, celui-ci doit répondre aux attentes de l'utilisateur en terme de cohérence des résultats, de rapidité et de prise en main de l'interface associée. La difficulté principale est de jouer avec les deux principaux aspects du moteur de recherche, à savoir : l'obtention et l'analyse de la requête utilisateur d'une part, et l'interrogation de la bibliothèque à travers le processus d'indexation d'autre part. Pour le premier point, il est indispensable d'offrir à l'utilisateur des types de requêtes divers et variés ainsi que leur description automatique dans l'une des représentations prises en compte par le processus d'indexation. Pour le second point, il est nécessaire de comparer, grâce à des mesures de dissimilarité adaptées, la requête de l'utilisateur avec l'ensemble des descriptions de forme stockées au préalable dans la bibliothèque.

Afin de bien comprendre le fonctionnement global d'un tel processus de recherche, nous allons brièvement expliquer, dans la suite de cette section, les trois étapes qui lui sont indispensables, à savoir :

- La définition des types de requêtes possibles, ainsi que la description de leur forme.
- La comparaison de la signature de la requête avec l'ensemble des signatures stocké dans la base de données.
- La mise en place d'une interface homme-machine orientée vers l'utilisateur.

6.2.1 La requête de l'utilisateur

Lorsque l'utilisateur souhaite rechercher un objet 3D précis dans la base de données, celui-ci doit formuler sa recherche au moyen d'une requête. Celle-ci peut prendre différentes formes qui vont des mots clés aux objets 3D en passant par des photos. Afin d'aider l'utilisateur dans cette formulation, le moteur de recherche doit accepter un nombre res-

treint de types de requêtes. Cependant, même s'il est nécessaire de limiter leur nombre, il est indispensable de garantir une variété suffisante pour offrir une certaine flexibilité à l'utilisateur. Il est, par exemple, difficilement concevable de n'autoriser que des requêtes de type "objet 3D". En effet, l'utilisateur qui recherche un modèle 3D n'est, en général, pas en possession d'un exemple de celui-ci.

Une fois la requête de l'utilisateur obtenue, il est nécessaire, pour la comparer à la base de données, de décrire sa forme. En effet, l'efficacité du moteur de recherche tient dans l'utilisation du processus d'indexation défini au chapitre 5. Dans ce cadre, il est indispensable d'analyser la requête afin d'obtenir une signature similaire à celles proposées pour les objets 3D. Cependant, il faut faire attention à adapter la description de forme au type de la requête (voir section 6.3).

6.2.2 La comparaison avec la base de données

L'étape précédente a permis de décrire la requête, formulée par l'utilisateur, dans l'une des représentations prises en compte par le processus d'indexation (voir section 6.2.1). Cependant, même si cette indexation permet la description de forme, indispensable à la recherche, elle n'offre pas de solutions quant à la comparaison de deux signatures. Pour cela, il est nécessaire de définir une mesure de dissimilarité, ou distance, entre deux descriptions synthétiques ainsi qu'une procédure permettant la comparaison de la requête avec l'intégralité de la base de données.

À cause de la variété des représentations choisies (histogrammes, ensemble de pixels...), il est nécessaire de définir des mesures de dissimilarités adaptées à chaque descripteur. Ainsi, il devient possible de calculer une distance entre deux objets. En effet, puisque la mesure de dissimilarité permet de comparer deux signatures, il est alors possible de comparer deux à deux chaque silhouette du modèle 3D et ainsi de donner une distance globale entre deux objets 3D. Finalement, pour renvoyer à l'utilisateur une liste de résultats pertinents, il suffit de calculer cette distance entre la requête et chaque objet de la base de données. Ainsi, il sera possible de lui renvoyer comme résultat la liste d'objets de la bibliothèque ordonnés selon leur distance à la requête (voir sections 6.4 et 6.5). Cependant, même si nous sommes désormais en présence d'une liste de résultats, il faut bien se rendre compte que tout ceci est impossible sans une interface homme-machine permettant d'obtenir la requête de l'utilisateur et de lui renvoyer les résultats de sa recherche.

6.2.3 Interface homme-machine

Contrairement à ce que l'on peut voir dans l'état de l'art, proposé dans le chapitre 3, la conception d'une interface homme-machine, permettant le dialogue entre l'utilisateur et le processus d'interrogation, est une étape cruciale pour la recherche efficace d'objets 3D. En effet, même si le processus mis en place pour indexer chaque objet et pour comparer chaque signature est pertinent, il est inutile s'il n'est pas associé à un outil d'interrogation.

Même s'il existe actuellement de nombreux moteurs de recherche d'objets 3D, tels que l'outil *Google 3D Warehouse*[1], ils demeurent inappropriés à la recherche d'objets 3D. En effet, dans la plupart des cas, soit ils ne permettent qu'une recherche par "objet 3D", inadaptée à l'utilisateur, soit ils offrent des résultats peu satisfaisants. De plus, leur ergonomie ne permet pas, le plus souvent, une prise en main facile pour un utilisateur lambda. Ainsi, il est nécessaire d'offrir à l'utilisateur des outils conçus pour répondre à ses attentes en terme de types de requête, de pertinence des résultats et de rapidité (voir section 6.6).

6.2.4 Schéma global

Comme nous venons de le présenter, la conception d'un processus complet d'interrogation est complexe et doit prendre en compte des contraintes diverses et variées. Celles-ci vont des attentes de l'utilisateur à l'interfaçage avec le processus d'indexation en passant par la conception d'un outil simple et ergonomique. La figure 6.1 présente le fonctionnement global d'un processus d'interrogation orienté vers l'utilisateur final.

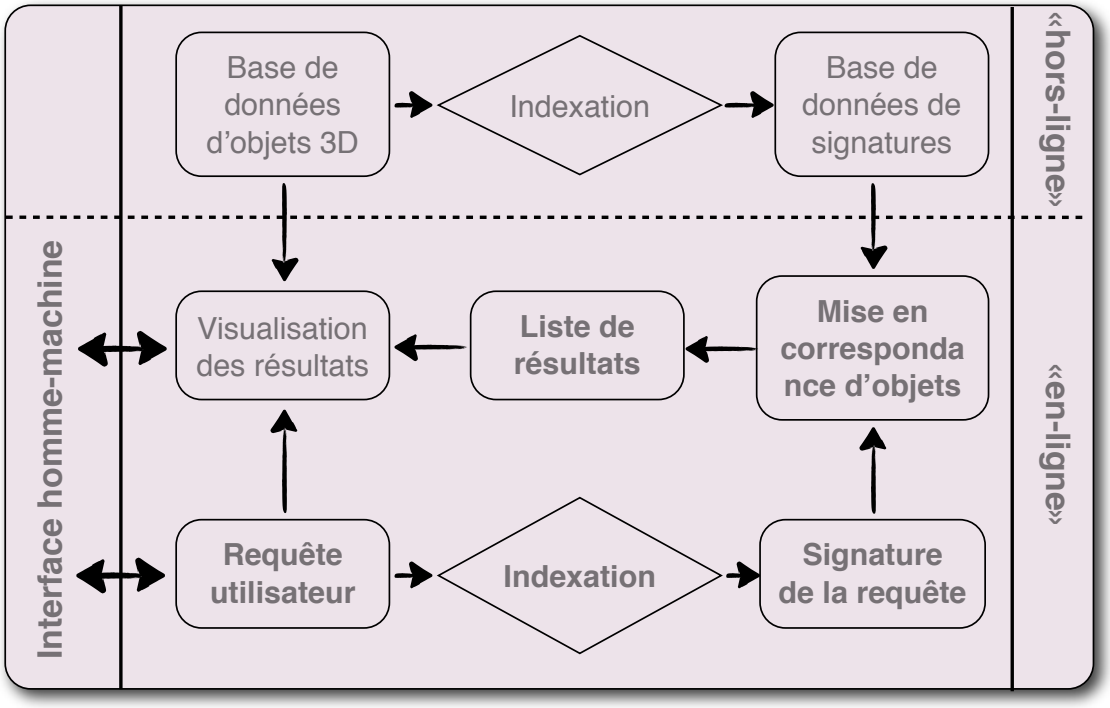


FIGURE 6.1 – *Fonctionnement global d'un processus de recherche allant de la description de la requête utilisateur, à sa comparaison avec la base de données d'objets 3D en passant par l'interface homme-machine permettant le dialogue entre l'utilisateur et le moteur de recherche.*

1. http://www.google.fr/3dwarehouse/

6.3 Définition, analyse et description des requêtes

Comme nous l'avons présenté dans la section précédente, la construction d'un processus de recherche se base avant tout sur la requête de l'utilisateur. Afin de le guider vers une recherche efficace, nous avons choisi de lui proposer trois types de requêtes permettant, de ce fait, de faire une recherche multimodale. Il s'agit d'un objet 3D, de photos ou de dessins au trait. Le choix de ces trois types s'est fait après une étude des besoins de l'utilisateur en terme de moyens d'interrogation. En effet, la représentation d'un modèle 3D étant complexe, comment formuler une requête mentale ? Le type de requête qui se rapproche le plus de la représentation mentale qu'un humain se fait d'un objet 3D est, selon nous, une requête en deux dimensions de type photos ou dessins au trait. C'est dans ce sens que nous avons aussi défini les descripteurs du chapitre 5.

Cependant, afin de garantir rapidité et pertinence des résultats, le moteur de recherche doit s'appuyer sur la puissance du processus d'indexation proposé dans le chapitre 5. Dans ce sens, il est nécessaire de décrire synthétiquement la forme de la requête de l'utilisateur. Étant donné la variété de types de celle-ci, nous adaptons dans la suite de cette section le processus à chaque type de requête possible. Finalement, même s'il devient possible de comparer, à terme, deux objets 3D, il faut cependant bien se rendre compte que le type de requête va orienter, de façon non négligeable, le choix de telle ou telle signature.

6.3.1 Requête de type "objet 3D"

Le premier type de requête que nous proposons, pour interroger la base de données, est l'objet 3D. Même si celui-ci est peu adapté à l'utilisateur, il permet d'obtenir des résultats très pertinents lorsque l'on a un exemplaire de l'objet que l'on souhaite rechercher. Cela vient principalement de la proximité entre les modèles 3D et la requête.

Lorsque l'utilisateur fournit au processus de recherche une requête de ce type, l'analyse et la description de sa forme sont simples. En effet, dans ce cas, étant donné que les descripteurs proposés dans le chapitre 5 s'obtiennent tous à partir d'un objet en trois dimensions, l'indexation de la requête peut suivre le même procédé. Cependant, le processus d'alignement proposé dans le chapitre 4 ne garantissant pas la robustesse de l'ordre des axes de rotation ainsi que l'invariance en réflexions, il est nécessaire de prendre en compte ces contraintes lors de l'indexation de la requête. Ainsi, comme présenté dans la section 4.3.4, nous calculons, uniquement pour la requête, les 48 orientations possibles du modèle permettant de tenir compte de ces contraintes lors du processus de comparaison de la section 6.4 (voir figure 6.2).

Finalement, le point fort des requêtes de type "objets 3D" est la possibilité d'utiliser l'ensemble des descripteurs proposés dans le chapitre 5. En revanche, le principal point faible est la difficulté, pour l'utilisateur, de formuler une requête de ce type. En effet, celui-ci ne détient généralement aucun exemple tridimensionnel de ce qu'il recherche.

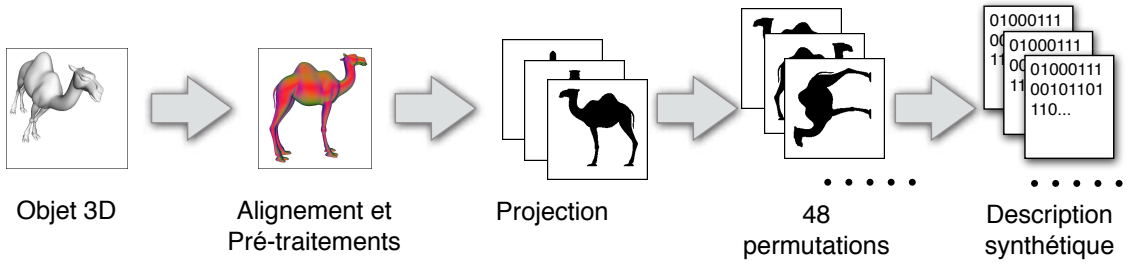


FIGURE 6.2 – *Exemple de l'indexation d'une requête de type "objet 3D". Dans ce cas, l'objet 3D est prétraité pour l'aligner et le préparer pour la phase d'indexation (voir chapitre 5), projeté sur des plans 2D, décrit synthétiquement en prenant en compte les 48 permutations possibles.*

6.3.2 Requête de type "photos"

Lorsque l'utilisateur ne fournit plus un objet 3D mais une ou plusieurs photos de l'objet qu'il recherche, la phase d'indexation nécessite une adaptation. La première étape est la séparation de l'objet et de son fond grâce à une approche similaire à celle proposée par McGuinness et O'Connor [70]. En effet, étant donné que les descripteurs de forme proposés dans le chapitre 5 traitent des objets 3D représentés sans fond, il est impossible de décrire des photos ayant un arrière-plan.

Afin de prendre en compte les erreurs d'alignement qui peuvent survenir sur chaque photo, il peut être utile d'aligner l'objet qui y est présenté à l'aide d'une analyse en composante principale "ACP". Ensuite, il est nécessaire de traiter chaque photo pour prendre en compte les erreurs qui peuvent survenir après l'alignement des objets 3D (voir section 4.3.4). En effet, l'ordonnancement des axes du modèle 3D n'étant pas robuste, il est nécessaire d'appliquer sur chaque photo une rotation 2D de 0°, 90°, 180° et 270° afin de corriger cet éventuel problème. D'autre part, il est nécessaire de traiter l'invariance en réflexion, qui n'est pas non plus prise en compte par le processus d'alignement, en appliquant un miroir à chaque photo utilisée comme requête par l'utilisateur. On obtient ainsi un ensemble de $4 * 2 = 8$ configurations.

En dernier lieu, pour caractériser la forme de chaque photo, il est nécessaire d'adapter le processus d'indexation. En effet, même si la description de la requête avec les descripteurs par ensembles de pixels ou par convexités/concavités est aisée (voir figure 6.3) celle-ci est plus complexe pour les cartes de normales. Cela vient du fait que l'information d'orientation de la surface est absente sur une photo en deux dimensions. Ainsi, il est nécessaire de la reconstruire en utilisant une approche stéréoscopique sur plusieurs photos de la même requête prises dans des angles de vues différents.

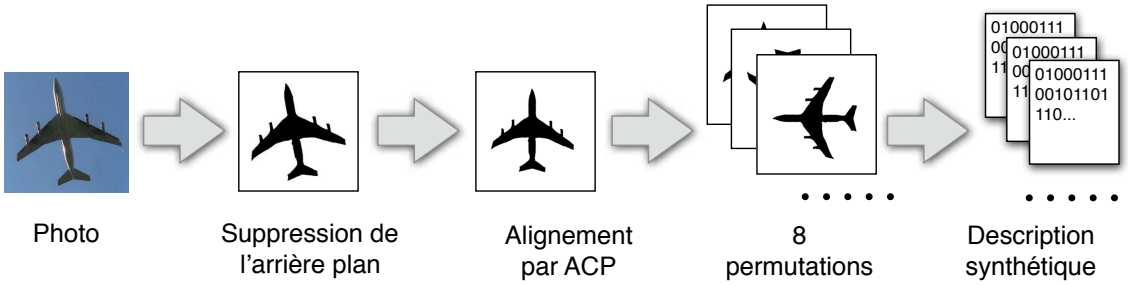


FIGURE 6.3 – *Exemple de l'indexation d'une requête de type "photo" pour les descripteurs par ensembles de pixels et par convexités/concavités. Dans ce cas, nous supprimons l'arrière plan de la photo à l'aide d'un outil de segmentation, nous alignons l'objet à l'aide d'une ACP, finalement nous le décrivons synthétiquement en prenant en compte les 8 permutations possibles.*

Finalement, les requêtes de type "photo" offrent un moyen simple, pour l'utilisateur, de rechercher un objet 3D. De plus, l'indexation en arrière-plan peut utiliser les descripteurs par ensembles de pixels ou par convexités/concavités car basée sur la forme des silhouettes en deux dimensions. En revanche, la séparation de l'arrière-plan à l'aide d'un outil de segmentation avancé est une étape qui peut, dans de nombreux cas, rendre difficile, voir impossible, l'utilisation de photo pour cette recherche. En effet, ces outils sont encore peu développés et alourdissent le processus d'interrogation. Pour finir, l'absence d'informations sur l'orientation de la surface rend quasiment impossible la description de l'approche par cartes de normales.

6.3.3 Requête de type "dessins au trait"

Dans le cas d'une requête de type "dessin au trait", l'utilisateur peut proposer non pas un, mais plusieurs dessins. Quel que soit leur nombre, il est nécessaire de les indexer séparément. En effet, aucune information ne nous permet de savoir si une cohérence spatiale existe entre eux. D'autre part, un traitement des permutations d'axes et des réflexions doit aussi être mis en place car ces contraintes ne sont pas prises en compte dans l'alignement des objets 3D.

Afin d'assurer le traitement de ces contraintes lors du processus de comparaison, nous devons décrire chaque dessin dans plusieurs poses. Comme pour les requêtes de type "photos", nous calculons les permutations d'axes possibles, soit quatre positions obtenues par rotation 2D de 0°, 90°, 180° et 270° du dessin. Pour traiter les réflexions, nous appliquons un miroir à chaque dessin. Ainsi, nous obtenons un ensemble de $4 * 2 = 8$ poses. Finalement, nous ne prenons pas en compte le réalignement du dessin lui-même (voir remarque 6.1).

> **Remarque 6.1.** *Contrairement aux deux types de requêtes précédents, dans le cas de dessin au trait nous n'appliquons pas d'alignement de ceux-ci. Cela vient des choix effectués pour le traitement de la pose des objets 3D, proposé dans la section 4.3 du chapitre 4. En effet, les études en psychologie cognitive proposées par Marr [69] et Biederman [15] montrent que la représentation mentale, par un être humain, d'un objet 3D a une pose qui n'est pas arbitraire. En particulier, Leek [66] montre que pour une représentation monochrome, l'humain cherche à aligner l'objet avec les axes horizontaux et verticaux. Nous pouvons donc penser que les dessins proposés par l'utilisateur seront, eux aussi, alignés avec les axes du repère. Finalement, étant donné que notre processus d'estimation de la pose cherche à positionner les modèles 3D de cette même manière, l'alignement des dessins n'est pas nécessaire.*

Par son aspect contour, ce type de requête n'est utilisable qu'avec deux des trois descripteurs proposés au chapitre précédent (5). En effet, si les descriptions par ensemble de pixels ainsi que par convexités/concavités du contour sont bien adaptées à ce type de requête, il n'en est pas de même pour les cartes de normales (voir figure 6.4). En effet, il est impossible de reconstruire les normales à la surface de l'objet avec ce type de requête.

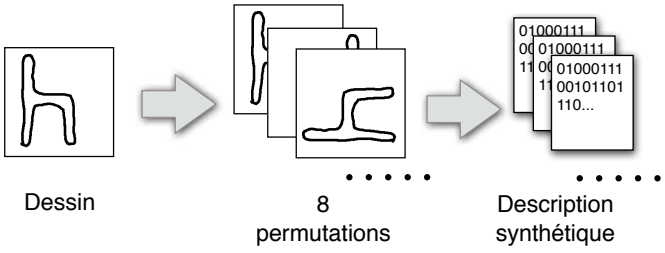


FIGURE 6.4 – *Exemple de l'indexation d'une requête de type "dessin au trait" pour les descripteurs par ensembles de pixels et par convexités/concavités. Dans ce cas, nous décrivons la forme de l'esquisse de façon synthétique en prenant en compte les 8 permutations possibles.*

Finalement, le grand avantage des requêtes de type "dessin au trait" est leur simplicité. En effet, les dessins associent un fort pouvoir expressif avec une facilité à représenter une requête mentale. Ainsi, l'utilisateur n'a plus besoin de détenir un objet 3D, il lui suffit simplement de décrire, à l'aide d'une ou plusieurs esquisses, le modèle qu'il recherche. En revanche, il est impossible de caractériser la forme des dessins au trait à l'aide du descripteur par cartes de normales.

6.3.4 Conclusion

Pour conclure, il est clair que l'approche par requête de type "objet 3D" est la plus simple à mettre en place. Inversement, les interrogations avec des photos sont difficiles

à traiter car nécessitant une suppression de l'arrière-plan. En ce qui concerne le dernier type de requête, à savoir les dessins au trait, ils offrent une formulation très facile avec un pouvoir expressif puissant permettant d'esquisser rapidement une requête mentale. Finalement, il est à noter que le descripteur par cartes de normales est peu adapté aux requêtes en deux dimensions (voir tableau 6.1).

	Ensemble de pixels	*Convexités/Concavités*	*Carte de normales*
Objet 3D	✓	✓	✓
Photo	✓	✓	✓
Dessin au trait	✓	✓	✗

TABLE 6.1 – *Tableau récapitulatif de la compatibilité entre les trois descripteurs proposés au chapitre 5 et les trois types de requêtes possibles pour interroger une base de données d'objets 3D.*

Finalement, nous avons présenté les trois types de requêtes permettant à l'utilisateur d'interroger une base de données. La phase d'indexation proposée au chapitre 5 a été adaptée pour prendre en compte chaque formulation. Cependant, même si nous sommes désormais capables d'exprimer la forme de la requête avec un descripteur, aucune solution n'a été proposée quant à la comparaison de celui-ci avec la bibliothèque d'objets 3D.

6.4 Principe de comparaison

Pour renvoyer des résultats à l'utilisateur, il est nécessaire de parcourir la base de données de signatures en vue de trouver les objets ressemblants à sa requête. La manière la plus courante, que nous utiliserons dans ce chapitre, est le calcul d'une dissimilarité, ou distance, entre la description de la requête, extraite dans la section précédente et chaque description d'objet 3D contenue dans la bibliothèque. Ainsi, il devient possible de renvoyer, à l'utilisateur, la liste des objets qu'il recherche.

Dans ce sens, nous proposons dans cette section des mesures de dissimilarités permettant de comparer deux descriptions de forme. Puis, un processus de comparaison de la requête par rapport à l'ensemble des objets de la base de données. Finalement, nous présentons les diverses façons de renvoyer les résultats à l'utilisateur.

6.4.1 Mesures de dissimilarité

Le but de l'utilisateur est de retrouver les objets qui ont une forme similaire à sa requête. Étant donné que nous sommes en possession des descriptions de forme de chaque objet 3D, de la base de données et de la requête, nous pouvons les comparer. Pour cela,

nous définissons une mesure de dissimilarité permettant de donner un degré de dissemblance entre la requête et un modèle 3D.

Étant donné que nous avons défini trois descripteurs permettant de caractériser la forme d'un objet 3D, il est nécessaire de définir une mesure de dissimilarité adaptée à chacun d'entre eux. Dans ce sens, nous définissons dans cette partie, chacune de ces mesures, en mettant en avant leurs avantages et inconvénients.

Mesure associée au descripteur par ensemble de pixels (EP)

Pour comparer deux descriptions basées sur la notion d'ensemble de pixels (voir section 5.3 du chapitre 5) nous avons choisi de définir la mesure de dissimilarité comme étant le cardinal de la différence symétrique, notée Δ, des deux ensembles E_1 et E_2 (voir figure 6.5 et remarque 6.2). En d'autres termes, il s'agit des pixels appartenant à E_1 ou à E_2 mais pas au deux ou encore à l'union de E_1 et E_2 à laquelle on soustrait leur intersection. Sa définition ensembliste est la suivante :

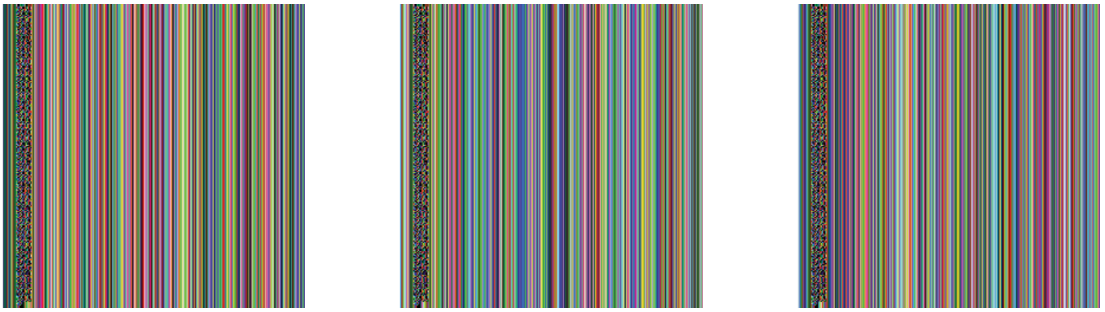
$$E_1 \Delta E_2 = \{p | (p \in E_1 \cup E_2) \wedge (p \notin E_1 \cap E_2)\} \tag{6.1}$$

> **Remarque 6.2.** *En voyant la figure 6.5, on peut se demander s'il ne serait pas judicieux de chercher à minimiser la différence symétrique (voir équation 6.1) indépendamment pour chaque silhouette. Par exemple, si l'on regarde la figure 6.5(c) on voit que la mesure de dissimilarité augmente à cause de petites translations. Cependant, un tel traitement ne doit pas être appliqué indépendamment sur chaque silhouette d'un objet 3D. En effet, cela romprait la cohérence spatiale qui existe entre chaque silhouette.*

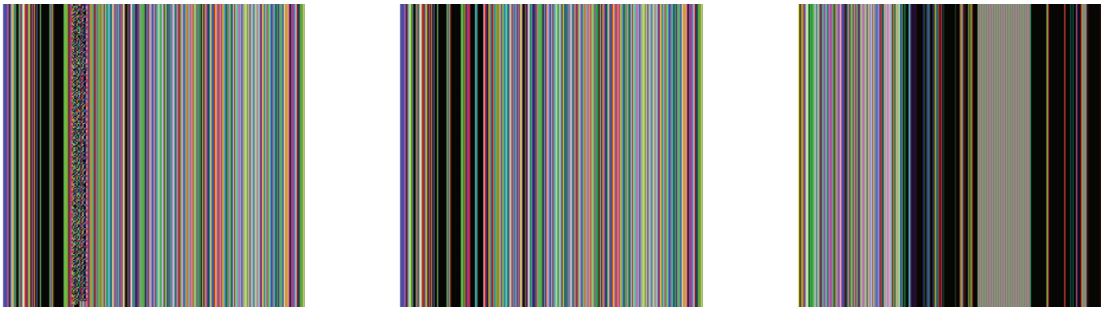
Afin de rendre la description ensembliste compréhensible par la machine, nous avons choisi, dans le chapitre précédent, de la transcrire en un tableau d'entiers (voir section 5.3.2). Dans celui-ci, chaque entier code le nombre de pixels d'une bande de couleur unique (blanche ou noir). À cause de cette compression, il n'est pas possible d'appliquer directement le calcul de différence symétrique sur ces descripteurs, qui ne représentent pas des silhouettes. Afin de calculer notre mesure de dissimilarité, nous pourrions envisager de reconstruire la silhouette initiale en décompressant la signature synthétique. Cependant, cette étape alourdirait le temps de comparaison à la fois par la phase de décompression que par celle de calcul de la différence symétrique. En effet, celui-ci nécessiterait alors $2 * 256 * 256 = 2^{17}$ opérations (test de la couleur du pixel) pour des silhouettes de $256 * 256$ pixels.

Afin d'optimiser notre mesure de dissimilarité, nous avons choisi de comparer directement les signatures compressées sans passer par une phase de décompression. Il s'agit alors de comparer les bandes de couleur contenues dans la description. Le principe de comparaison de deux bandes est le suivant : si la taille de la bande 1 est inférieure à la taille de la bande 2 et que leurs couleurs sont différentes alors on ajoute la taille de la bande 1

(a) *Silhouettes de l'objet* $\mathcal{O}_1$.



(b) *Silhouettes de l'objet* $\mathcal{O}_2$.



(c) *Différence symétrique des silhouettes des objets* $\mathcal{O}_1$ *et* $\mathcal{O}_2$.

FIGURE 6.5 – *Représentation de la mesure de dissimilarité choisie basée sur la différence symétrique (c) des silhouettes de deux objets (a) et (b).*

à la somme des différences symétriques. Puis on enlève les pixels qui ont été comparés (la taille de la bande 2 moins la taille de la bande 1). Enfin, on réitère jusqu'à atteindre la fin de l'image. Dans le cas où la taille de la bande 1 est supérieure ou égale à la taille de la bande 2, on effectue un traitement similaire décrit dans l'algorithme 6.1.

Pour conclure, nous avons proposé une méthode permettant de comparer deux descripteurs basés sur les ensembles de pixels et ceci sans passer par leur décompression. Les points forts de cette mesure de dissimilarité sont :

- L'algorithme basé sur la version compressée des descripteurs.
- L'opération de différence symétrique permettant de rester dans la formulation en-

Algorithme 6.1 *Algorithme de comparaison de deux descripteurs basé sur les ensembles de pixels.*

```
1: couleur_objet1 = blanc
2: couleur_objet2 = blanc
3: difference_symetrique = 0
4:
5: Pour i = 0 à taille_image Faire
6:   Si descripteur_objet1.tete > descripteur_objet2.tete Alors
7:     Si couleur_objet1 ≠ couleur_objet2 Alors
8:       difference_symetrique += descripteur_objet2.tete
9:     Fin Si
10:    couleur_objet2 =!couleur_objet2
11:    descripteur_objet1.tete -= descripteur_objet2.tete
12:    descripteur_objet2.EnleverTete ()
13:    continuer boucle
14:  Fin Si
15:
16:  Si descripteur_objet1.tete < descripteur_objet2.tete Alors
17:    Si couleur_objet1 ≠ couleur_objet2 Alors
18:      difference_symetrique += descripteur_objet1.tete
19:    Fin Si
20:    couleur_objet1 =!couleur_objet1
21:    descripteur_objet2.tete -= descripteur_objet1.tete
22:    descripteur_objet1.EnleverTete ()
23:    continuer boucle
24:  Fin Si
25:
26:  Si descripteur_objet1.tete == descripteur_objet2.tete Alors
27:    Si couleur_objet1 ≠ couleur_objet2 Alors
28:      difference_symetrique += descripteur_objet1.tete
29:    Fin Si
30:    couleur_objet1 =!couleur_objet1
31:    couleur_objet2 =!couleur_objet2
32:    descripteur_objet1.EnleverTete ()
33:    descripteur_objet2.EnleverTete ()
34:    continuer boucle
35:  Fin Si
36: Fin Pour
37:
38: Md = difference_symetrique
```

sembliste.

Tandis que son principal point faible est sa complexité qui varie d'une comparaison à une autre car dépendant du nombre de valeurs contenues dans la compression en bandes.

Mesure associée au descripteur par convexités/concavités

Pour le descripteur basé sur les convexités et concavités des contours, deux mesures de dissimilarité sont à définir. En effet, nous avons proposé dans le chapitre 5 deux codages différents, basés sur des histogrammes d'une part, et sur l'information complète d'autre part.

- **Cas des histogrammes (MCC-H)**

Nous avons présenté dans le chapitre précédent un codage synthétique permettant de réduire l'espace de stockage des valeurs de convexités/concavités. Celui-ci est basé sur une répartition des données à l'aide de K histogrammes de H composantes chacun. Dans cette signature, K représente le nombre d'échelles, utilisées pour calculer les informations de convexités et concavités, tandis que H traduit la granularité voulue pour la représentation. Afin de comparer ces histogrammes et ainsi de définir une mesure de dissimilarité entre eux, nous utilisons une distance L_1 entre leurs composantes (voir annexe A). De plus, afin de prendre en compte la pertinence de chaque échelle de la description, à sa juste valeur, nous donnons un degré d'importance à chaque composante, ceci au moyen d'un facteur de pondération α. La mesure de dissimilarité entre deux histogrammes s'écrit alors :

$$Md(\mathcal{D}_{\mathcal{O}_r}, \mathcal{D}_{\mathcal{O}}) = \sum_{i=0}^{K} \left(\alpha \sum_{j=0}^{H} \|\mathcal{D}_{\mathcal{O}_r}^{i,j} - \mathcal{D}_{\mathcal{O}}^{i,j}\|_1 \right) \tag{6.2}$$

ici, $\mathcal{D}_{\mathcal{O}}^{i,j}$ représente la composante j de l'histogramme d'échelle i, pour le descripteur $\mathcal{D}_{\mathcal{O}}$ de l'objet $\mathcal{O}$. Dans nos tests, nous utilisons comme pondération :

$$\alpha = \frac{1}{\sigma\sqrt{2 * \pi}} \exp(-\frac{i^2}{2\sigma^2}) \tag{6.3}$$

avec $\sigma = 3$. Celle-ci a été choisie car elle permet de donner plus de poids aux premières échelles qu'aux dernières qui tendent à représenter un cercle.

- **Cas des vecteurs complets (MCC)**

Nous avons proposé dans le chapitre 5 un descripteur de forme basé sur les convexités et concavités d'un contour. Afin de garantir une perte minimale de ces informations, nous avons choisi de stocker, dans la signature, l'ensemble des mesures de convexités/concavités calculé sur un contour. La question qui se pose alors, pour le calcul de la mesure de dissimilarité, est résumée dans la figure 6.6 et peut se formuler de la manière suivante : comment

évaluer la ressemblance entre les informations de concavités/convexités de deux contours ?

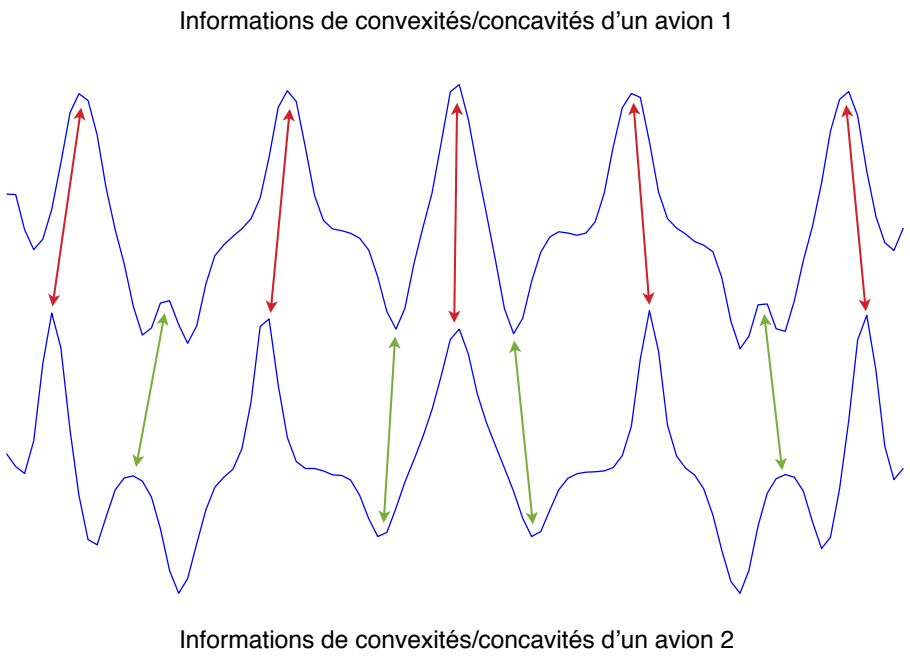


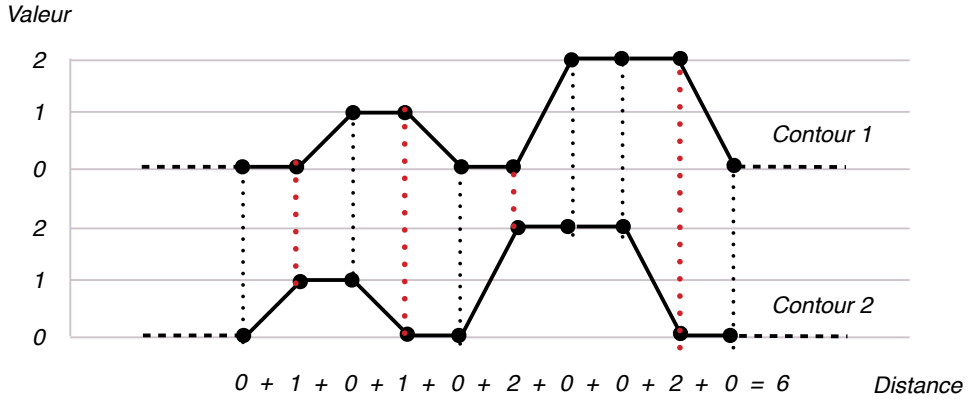
FIGURE 6.6 – *Illustration du problème de la mesure de dissimilarité des informations de concavités/convexités de deux contours d'avion.*

La représentation choisie peut être vue comme une courbe indiquant le degré de convexité où de concavité (ordonnées) en chaque point (abscisses). Ainsi, le problème se traduit par la recherche d'un appariement optimal entre les deux courbes. Cette recherche de mise en correspondance est un problème complexe nécessitant la mise en place d'algorithmes spécialisés. En effet, comme le montre la figure 6.7, un appariement naïf des informations de concavités/convexités est peu pertinent alors que l'introduction d'élasticité dans la mise en correspondance permet d'offrir des résultats plus intuitifs.

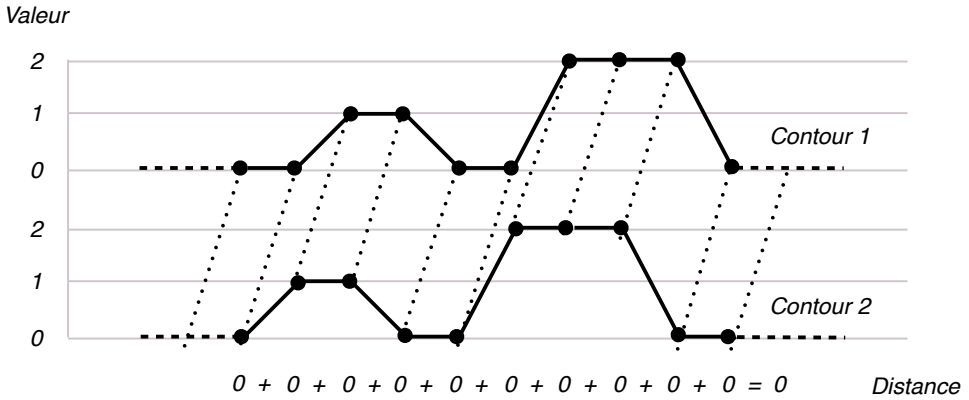
Afin de résoudre efficacement ce problème, nous construisons une matrice de correspondance M de taille $N * N$, où N représente le nombre de points de chaque contour (voir 5.4), où chaque cellule $M_{i,j}$ contient la distance d entre les points a_i et b_i des contours A et B. Cette mesure se calcule comme suit :

$$d_{a_i,b_j} = \frac{1}{K} \sum_{\sigma=1}^{K} ||a_{i,\sigma} - b_{j,\sigma}||_1 \tag{6.4}$$

où K représente le nombre d'échelles utilisées pour calculer les informations de convexités et de concavités des contours A et B (voir section 5.4). Finalement, a partir de cette matrice M, trouver la mise en correspondance "optimale" entre les deux contours revient



(a) *Mise en correspondance naïve de deux contours*



(b) *Mise en correspondance élastique de deux contours*

FIGURE 6.7 – *Exemple de deux mesures de dissimilarité permettant de comparer deux séquences d'informations de concavités/convexités. (a) illustre une mesure naïve donnant une distance globale de* 6 *tandis que (b) illustre une mise en correspondance élastique des informations aboutissant à une distance de* 0.

à calculer le chemin de coût minimal pour relier la cellule $M_{0,0}$ à la cellule $M_{N-1,N-1}$.

Afin de tendre vers un coût minimum, de nombreuses méthodes ont été proposées. En particulier les algorithmes de "programmation dynamique" permettent de résoudre ce problème en se basant sur le principe de Bellman [11] formulé comme suit : « Toute sous-séquence d'une séquence optimale doit elle-même être optimale ». Ainsi, il est possible de déduire la solution "optimale" d'un problème à partir de la solution "optimale" des sous-problèmes. Ces méthodes sont largement utilisées dans la littérature, sous des formes diverses et variées, pour résoudre des problèmes qui vont de la mise en correspondance de mots à la reconnaissance vocale en passant par l'appariement de molécules. En particulier, une approche de "programmation dynamique" basée sur les algorithmes de Needleman et Wunsch [78] et de Smith et Waterman [113] a été utilisée par Chaouch [24] pour comparer des lignes de profondeurs issues d'objets 3D.

Dans notre cas, il ne s'agit pas de séquences d'états à comparer, mais de courbes à mettre en correspondance, tel que sur la figure 6.7. Pour résoudre ce type de problème, il existe une variante de la "programmation dynamique" appelée "déformation temporelle dynamique". Cette méthode permet de mettre en correspondance de façon élastique des courbes qui peuvent varier au cours du temps. Nous utilisons donc, pour aligner localement nos deux courbes A et B, une adaptation de la méthode employée par les auteurs des informations de convexités/concavités [1] basée sur l'approche de Myers *et al.* [74]. Celle-ci a été modifiée au niveau de l'alignement des informations qui s'effectue, pour nous, par les prétraitements du chapitre 4.

Dans notre algorithme d'appariement, une première contrainte doit être mise en place pour garantir la pérennité de la séquentialité des contours. Celle-ci est obtenue en imposant que le chemin "optimal" ne puisse pas faire de retour en arrière sur les lignes où les colonnes. Cette contrainte, appelée "contrainte d'ordre", oblige le chemin à progresser vers le "haut" ou vers la "droite". Une seconde contrainte doit être imposée afin de garantir que chaque point du contour A soit mis en correspondance avec un point du contour B. Celle-ci se traduit par l'obligation d'utiliser chaque colonne pour le calcul du chemin de coût minimal.

Le caractère itératif de la "déformation temporelle dynamique" basé sur la résolution "optimale", en terme de coût, d'un sous-problème permet de résoudre simplement le problème complet. Notons D la matrice de "déformation temporelle dynamique" et $D(i,j)$ le coût du chemin "optimal" permettant de relier la cellule $M_{0,0}$ à la cellule $M_{i,j}$. Alors, la valeur de $D(i,j)$ de toute cellule de D se calcule comme suit :

$$D(i,j) = M(i,j) + \min \begin{cases} D(i-1,j) \\ D(i-1,j-1) \\ D(i,j-1) \end{cases} \tag{6.5}$$

Pour finir, la mesure de dissimilarité entre les deux contours A et B est égale au coût du chemin "optimal" complet, soit la valeur dans la cellule $D(N-1,N-1)$. Le processus de calcul complet est présenté dans l'algorithme 6.2 et la figure 6.8. Des exemples de mise en correspondance sont visibles sur la figure 6.9.

Afin d'optimiser le temps de calcul du chemin de coût "optimal", nous ne calculons $D(i,j)$ que pour les cellules situées dans une bande diagonale de taille T. Dans notre implémentation, nous utilisons une valeur de $T = 0.08 * N$ avec $N = 100$, ce qui permet de réduire le temps de calcul sans trop de perte sur la qualité de l'appariement. À cause de cette contrainte, le calcul de toutes les valeurs de la matrice de distance M n'est pas nécessaire. Ainsi, nous ne calculons la valeur des cellules $M(i,j)$ que lors de leur utilisation.

Finalement, nous interdisons qu'un point d'une courbe soit mis en correspondance avec plus de deux points de l'autre courbe. Cela se traduit par l'absence de longs segments

Algorithme 6.2 *Algorithme de "déformation temporelle dynamique" permettant de mettre en correspondance deux contours A et B.*

1: **Pour** i = 1 à N **Faire**
2: **Pour** j = 1 à N **Faire**
3: $M(i,j) = d_{a_i,b_j}$ (Equation 6.4)
4: **Fin Pour**
5: **Fin Pour**
6: **Pour** i = 1 à N **Faire**
7: **Pour** j = 1 à N **Faire**
8:

$$D(i,j) = M(i,j) + \min \begin{cases} D(i-1,j) & (insertion) \\ D(i-1,j-1) & (appariement) \\ D(i,j-1) & (suppression) \end{cases}$$

9: **Fin Pour**
10: **Fin Pour**
11: $Md = D(N-1, N-1)$

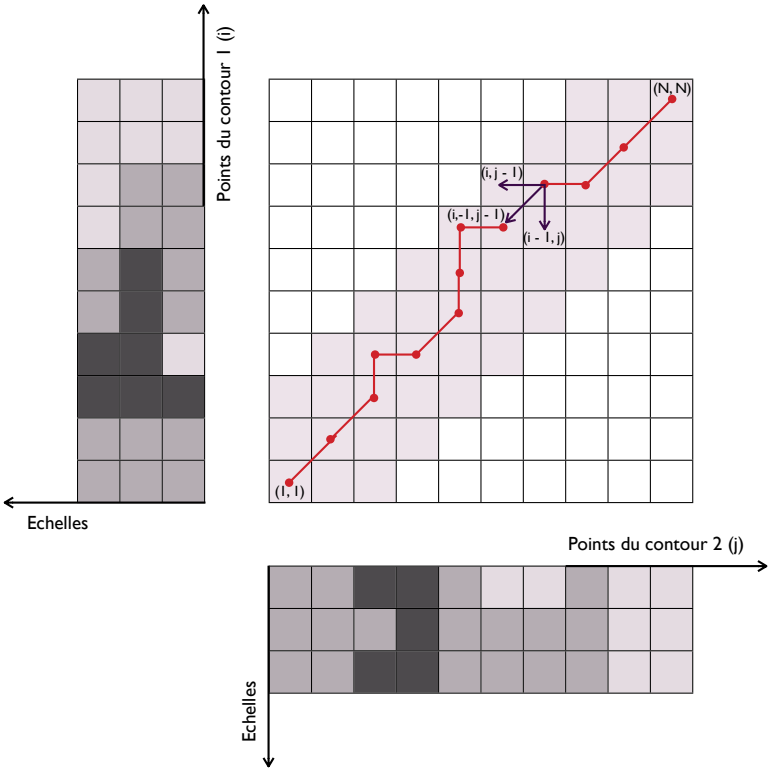
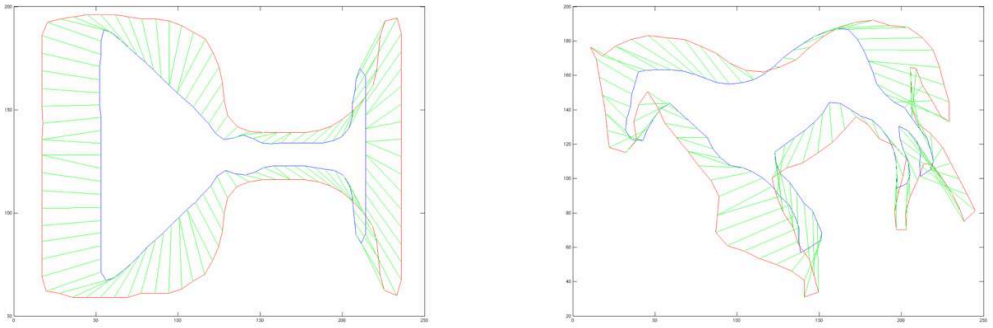


FIGURE 6.8 – *Illustration de l'algorithme 6.2 expliquant la "déformation temporelle dynamique". Cette méthode nous permet de comparer les informations de convexités/concavités de deux contours.*



(a) *Appariement de deux verres* (b) *Appariement de deux chevaux*

FIGURE 6.9 – *Deux exemples de mise en correspondance de contours en utilisant une "déformation temporelle dynamique" sur leurs informations de convexités/concavités. (a) et (b) représente respectivement l'appariement de deux verres et de deux chevaux.*

dans le chemin "optimal". Cette contrainte est prise en compte dans l'algorithme de "déformation temporelle dynamique" en mettant à jour les valeurs des cellules $D(i-1,j)$ et $D(i,j-1)$, avant le calcul de l'équation 6.5, comme suit :

$$D(i-1,j) = \begin{cases} D(i-1,j) & \text{si } pred_V(i-1,j) \neq j \\ \infty & \text{sinon} \end{cases} \tag{6.6}$$

et

$$D(i,j-1) = \begin{cases} D(i,j-1) & \text{si } pred_H(i,j-1) \neq i \\ \infty & \text{sinon} \end{cases} \tag{6.7}$$

avec $pred_V(i,j)$ et $pred_H(i,j)$ qui représente respectivement la position verticale et horizontale du prédécesseur de coût minimal pour la cellule $D(i,j)$.

• Avantages et inconvénients de ces mesures

Pour conclure, nous avons proposé des mesures de dissimilarité adaptées à nos deux représentations synthétiques traduisant les informations de convexités/concavités des contours. Les points forts de la mesure pour les histogrammes sont :

– La très grande rapidité de calcul de la mesure.
– La pondération qui tient compte des échelles.

Le seul point faible de cette mesure est son faible pouvoir discriminant qui vient plus des histogrammes que de la mesure. En ce qui concerne la mesure pour les vecteurs complets, les avantages sont :

– La grande élasticité de la mesure permettant une mise en correspondance puissante des informations de convexités/concavités.
– L'utilisation d'optimisation dans la "déformation temporelle dynamique" permettant de réduire la complexité algorithmique.

Même si cette mesure permet une comparaison juste de deux descriptions, elle a deux inconvénients majeurs. Le premier est l'utilisation de la "déformation temporelle dynamique" ne permettant pas de vérifier la définition d'une métrique selon Copson [30] (voir annexe A). Le second est la complexité élevée du calcul de cette mesure de dissimilarité ralentissant, de ce fait, le processus de comparaison global de deux signatures de ce type.

Mesure associée au descripteur par cartes de normales (CN)

Nous avons aussi proposé dans le chapitre 5 une description basée sur l'orientation du maillage d'un objet 3D que nous avons appelée "carte de normales". Afin de comparer efficacement ces informations, codées de façon synthétique dans des cœfficients de la transformée de Fourier, nous avons choisi de baser notre mesure de dissimilarité sur la distance L_1. Sachant que chaque carte de normales a été décrite à l'aide de n cœfficients, alors, la mesure de dissimilarité est la suivante :

$$Md(\mathcal{D}_{\mathcal{O}_r}, \mathcal{D}_{\mathcal{O}}) = \sum_{i=0}^{n} \|\mathcal{D}^i_{\mathcal{O}_r} - \mathcal{D}^i_{\mathcal{O}}\|_1 \tag{6.8}$$

où, $\mathcal{D}^i_{\mathcal{O}}$ représente la valeur i du vecteur stocké dans le descripteur $\mathcal{D}_{\mathcal{O}}$ de l'objet $\mathcal{O}$.

Pour conclure, nous avons proposé une mesure de dissimilarité permettant de comparer deux cartes de normales. Cette distance a l'avantage d'être rapide à calculer car basée sur la distance L_1.

6.4.2 Comparaison de deux descriptions

Les mesures de dissimilarité définies dans la section précédente permettent de comparer la description de forme d'une silhouette avec la requête. Sachant que nous avons choisi une approche multi-vues (voir section 5.2 du chapitre 5) pour la caractérisation des modèles 3D, il est nécessaire de mettre en place un processus permettant de comparer l'ensemble des silhouettes avec la requête.

Dans le but de répondre efficacement à la demande de l'utilisateur, nous proposons dans cette section une méthode permettant de comparer une requête avec un objet 3D décrit à partir de plusieurs silhouettes. Étant donné que nous avons choisi d'offrir à l'utilisateur plusieurs façons d'interrogations d'une base de données, nous proposons des traitements différents pour les requêtes de type "objet 3D" que pour celles de type "photo" ou "dessin au trait".

Cas d'une requête de type "objet 3D"

Étant donné que n silhouettes ont été obtenues par projection, la distance entre la requête $\mathcal{O}_r$ et un objet $\mathcal{O}$ de la base de données revient à calculer la dissimilarité entre leurs vues similaires, en terme de point de vue. En effet, afin de garder la cohérence spatiale entre les projections du modèle 3D, qui a préalablement été aligné, il est nécessaire de comparer les silhouettes obtenues à partir du même point de vue. Ainsi, la distance entre la requête et un objet de la base se calcule de la manière suivante :

$$d(\mathcal{O}_r, \mathcal{O}) = \sum_{i=0}^{n} Md(\mathcal{D}^i_{\mathcal{O}_r}, \mathcal{D}^i_{\mathcal{O}}) \tag{6.9}$$

où Md représente la mesure de dissimilarité entre les descripteurs $\mathcal{D}^i_{\mathcal{O}_r}$ et $\mathcal{D}^i_{\mathcal{O}}$ de la requête et de l'objet de la base pour la projection i.

Cependant, étant donné que le processus d'alignement proposé dans le chapitre 4 ne garantit pas la robustesse de l'ordre des axes de rotation ainsi que l'invariance en réflexions, il est nécessaire de les prendre en compte lors de la comparaison. Ainsi, sachant que nous avons calculé pour la requête, dans la section 6.3.1, les 48 orientations possibles du modèle, la distance entre une requête 3D et un objet de la base de données devient :

$$d(\mathcal{O}_r, \mathcal{O}) = \min_{1 \leq \Theta \leq 48} \sum_{i=0}^{n} Md(\mathcal{D}^{i,\Theta}_r, \mathcal{D}^i) \tag{6.10}$$

où Θ représente l'une des configurations possibles parmi les 48.

Cas d'une requête de type "photo" ou "dessin au trait"

Dans le cas d'une requête de type photo (ou dessin au trait), l'utilisateur peut proposer non pas une, mais plusieurs photos (dessins). Quel que soit leur nombre, il est nécessaire de les comparer séparément. En effet, il n'existe pas, à priori, de relation entre les points de vues des photos (dessins) et des projections de l'objet 3D.

Sachant que n_1 silhouettes ont été obtenues par projection et que l'utilisateur a donné n_2 photos (dessins), la distance entre cette requête $\mathcal{O}_r$ et un objet $\mathcal{O}$ de la base de données s'obtient en recherchant pour chaque photo (dessin) la vue du modèle 3D la plus similaire. En effet, rien ne garantit que l'utilisateur ait donné autant de dessins que de vues de l'objet 3D. Dans ce cas, la distance entre la requête et l'objet de la base se calcule de la manière suivante :

$$d(\mathcal{O}_r, \mathcal{O}) = \sum_{i=0}^{n_2} \min_{1 \leq j \leq n_1} Md(\mathcal{D}^i_r, \mathcal{D}^j) \tag{6.11}$$

où Md représente la mesure de dissimilarité entre les descripteurs $\mathcal{D}^i_{\mathcal{O}_r}$ et $\mathcal{D}^j_{\mathcal{O}}$ du dessin i de la requête et de la projection j de l'objet 3D de la base.

Afin de prendre en compte les erreurs d'alignement qui peuvent survenir après le prétraitement des objets 3D (voir section 4.3.4 du chapitre 4) nous avons choisi de calculer 8 configurations possibles pour chaque photo (dessin) (voir section 6.3.3). Afin de prendre en compte l'ensemble de ces configurations dans le calcul de distance entre la requête et un objet 3D, le calcul de celle-ci devient :

$$d(\mathcal{O}_r, \mathcal{O}) = \min_{1 \leq \Theta \leq 8} \sum_{i=0}^{n_2} \min_{1 \leq j \leq n_1} Md(\mathcal{D}_r^{i,\Theta}, \mathcal{D}^j) \tag{6.12}$$

où Θ représente l'une des configurations possibles parmi les 8.

6.4.3 Liste des résultats

Grâce aux étapes précédentes, nous sommes en mesure de comparer la requête fournie par l'utilisateur avec un objet 3D et de leur attribuer une distance représentant leur degré de dissemblance. Cependant, afin de répondre au besoin de l'utilisateur, à savoir retrouver les objets 3D répondant à sa demande, il est nécessaire de comparer sa requête avec chaque objet de la base de données.

En appliquant le processus de comparaison itérativement entre la requête et chaque objet de la base de données, nous obtenons une distance pour chacun d'entre eux (voir remarque 6.3). La distance calculée représentant un degré de dissemblance, plus celle-ci est basse, plus l'objet à laquelle elle est associée a une forme dissimilaire à la requête de l'utilisateur. Ainsi, il est alors possible de retourner à l'utilisateur les modèles 3D qu'il recherche. Cependant, même si nous sommes désormais capables de renvoyer cette liste d'objets ordonnée par distance croissante, il est peut-être intéressant de réfléchir au besoin de l'utilisateur. En effet, il est fort probable que celui-ci cherche les objets ressemblants à une requête, mais pas ceux qui n'y correspondent pas. Pour cela plusieurs méthodes sont envisageables :

- La première est une approche de type "k plus proche". Dans celle-ci, on ne retourne à l'utilisateur que les k objets 3D les plus proches de sa requête. Le point négatif de cette approche, même si elle se focalise sur une liste pertinente, plus qu'exhaustive, de résultats, est le choix arbitraire de k qui ne correspond pas forcément au souhait de l'utilisateur. Afin de contourner cet aspect, il est possible d'organiser la liste de résultats sous forme de pages – comme pour le moteur de recherche de *Google* – avec k qui représente le nombre de résultats affichés par page. Ainsi, la navigation parmi les résultats est simplifiée tout en ne limitant pas leur nombre. Finalement, il peut être intéressant de laisser le choix du paramètre k à l'utilisateur garantissant ainsi une liste de résultats qui lui est adaptée (voir remarque 6.4).
- La seconde méthode permettant d'adapter la liste de résultats à l'utilisateur est appelée "approche par ϵ". Dans celle-ci, nous ne renvoyons plus à l'utilisateur les k plus proches résultats, mais ceux qui sont à une distance d inférieure à ϵ. La question principale à se poser ici est le choix du paramètre ϵ. Une première méthode peut être son réglage de façon arbitraire après étude statistique des résultats sur

un sous-ensemble de requête. Une seconde approche plus pertinente est la définition automatique de seuil par une approche analysant la distance entre la requête et un objet 3D représentant une forme aléatoire telle que dans les travaux de Hurtut [52].

Remarque 6.3. *Il est clair que ce type de recherche, où la complexité de comparaison est linéaire, n'est pas pertinent dans le cas de très grandes bases de données car ne passant pas à l'échelle. Dans le cas d'une recherche dans une bibliothèque de grande taille, il est nécessaire de mettre en place des techniques d'accélération de la recherche permettant d'offrir une complexité non linéaire. L'utilisation d'arbre de recherche, de techniques utilisant les pivots telles que celles proposées par Bustos* et al. *[21][18] ou encore des algorithmes distribués sur des fermes de machines peuvent résoudre ces problèmes de recherche dans de grandes bases de données.*

Remarque 6.4. *Si nous étudions le comportement d'un utilisateur face à un moteur de recherche, nous pouvons nous rendre compte que celui-ci effectue, en général, une dizaine d'opérations (consultation d'un résultat ou demande d'une nouvelle page de résultats) avant d'obtenir ce qu'il souhaite. À partir de ce constat, des méthodes ont été développées pour accélérer le processus de recherche. En particulier, les travaux de Crucianu* et al. *[32] montrent que si l'utilisateur désigne, dans la première page de résultats, un sous-ensemble de documents proches de sa recherche, alors il est possible de renvoyer rapidement une liste plus pertinente. Finalement, le nombre d'opérations nécessaires pour atteindre un document précis est inférieur à celui d'une recherche classique.*

6.5 Optimisations de la recherche

Nous avons présenté dans les sections précédentes de ce chapitre un processus permettant de renvoyer à l'utilisateur une liste de résultats répondant à sa requête. Pour cela, nous utilisons des mesures de dissimilarité adaptées à chacune des descriptions introduites dans le chapitre 5, associées à une comparaison de la requête avec l'ensemble des objets 3D de la base de données (à travers leur signature).

Du fait de ces choix, il est impossible d'utiliser plusieurs descriptions de forme pour rechercher un modèle précis. En effet, la caractérisation de la requête au moyen d'une méthode particulière nous oblige à utiliser le même type de signatures pour les objets 3D de la base de données. De ce fait, le résultat est lié au pouvoir discriminant de la méthode ainsi qu'à la mesure de dissimilarité qui lui est associée. Afin d'optimiser la pertinence des réponses de notre système, nous nous sommes penchés, dans la section 6.5.1, sur une approche de fusion des résultats. Celle-ci permet de combiner les distances obtenues avec les différents types de descripteurs du chapitre 5.

D'autre part, les mesures de dissimilarité que nous avons proposées dans la section 6.4.1 ont une complexité qui varie en fonction de leur précision. De ce fait, une recherche utilisant

une mesure simple sera efficace en temps, mais aura l'inconvénient d'être peu pertinente. En revanche, d'autres mesures plus puissantes, comme celles basées sur la "déformation temporelle dynamique", offriront des résultats répondant bien à la requête de l'utilisateur, mais dans un temps plus long. Afin d'offrir des résultats pertinents de façon rapide, nous proposons dans la section 6.5.2 une approche d'élagage précoce permettant de porter une attention plus particulière aux objets qui intéressent davantage l'utilisateur final.

6.5.1 Optimisation multi-descripteurs

Le processus de recherche présenté, dans la section 6.4, oblige à choisir l'un des trois descripteurs proposés, dans le chapitre 5, pour caractériser la forme de la requête. De ce fait, la comparaison de celle-ci avec la base de données de signatures est liée à la description choisie, tout comme la pertinence des résultats. Ce processus de recherche, bien que fonctionnel, ne permet donc pas de tirer parti de la force de chacun de nos descripteurs.

Afin d'améliorer les résultats renvoyés à l'utilisateur, nous nous sommes intéressés à la fusion d'informations, ou agrégation, issues de plusieurs sources, ou descriptions. Sachant que le processus de recherche que nous avons développé renvoie une liste d'objets résultats grâce à un calcul de distance par rapport à la requête (voir section 6.4), deux possibilités se sont offertes à nous :

1. Une fusion au niveau des distances, entre la requête et chaque objet 3D.
2. Une fusion au niveau des rangs, de chaque modèle 3D, dans la liste des résultats.

Pour garantir la pertinence des sources, nous avons choisi de les combiner le plus tôt possible en utilisant la première des deux méthodes. Ainsi, l'opération de fusion revient à combiner les distances, calculées entre la requête et un objet de la base, obtenues avec une partie ou l'ensemble des signatures extraites. La tâche principale est donc la définition d'un opérateur de fusion permettant de tirer parti de chaque source d'informations.

Nous notons, dans la suite de cette partie, s_i avec $1 \leq i \leq n$ les valeurs issues de n sources de données à fusionner et $\mathcal{A}(s_1,\ ...,\ s_n)$ le résultat de l'agrégation. Pour rester dans le formalisme de la fusion de données, nous utilisons non pas des distances, mais des scores, ou valeurs, tel que $v = 1 - d$ où d représente une distance. Ainsi, un objet proche de la requête aura un score proche de 1 tandis qu'un objet éloigné aura un score proche de 0. Finalement, nous supposons que les distances sont normalisées : $d \in [0,\ 1]$.

Théorie des opérateurs de fusion

Comme le montrent les travaux de Capitaine [23], la fusion, ou l'agrégation de données, est un problème complexe où rivalisent des opérateurs de fusion divers et variés. Afin de définir l'opération permettant de combiner des informations, il est nécessaire de définir une heuristique. Celle-ci doit permettre de savoir comment agréger deux informations venant

de deux sources différentes. L'heuristique peut, par exemple, être une estimation de la confiance envers les différentes sources. Par leur comportement, les opérateurs d'agrégations peuvent être classés en plusieurs catégories selon qu'ils sont conjonctifs, disjonctifs, de compensation... Leur définition est la suivante :

- **Opérateurs conjonctifs** : ces opérateurs offrent une vision "intolérante" ou "pessimiste" des sources. En effet, ils ne fournissent un score de fusion élevé que si l'ensemble des valeurs des sources est élevé. Ces opérateurs s'apparentent au "ET" logique où toutes les valeurs doivent être non nulles pour engendrer un score non nul. Ils sont définis comme suit :

  Définition 6.1. *Un opérateur $\mathcal{A}$ sera dit conjonctif si et seulement si :*

$$\mathcal{A}(s_1,\ ...,\ s_n) \leq \min(s_1,\ ...,\ s_n) \tag{6.13}$$

- **Opérateurs disjonctifs** : Contrairement aux précédents, ces opérateurs offrent une vision "tolérante" ou "optimiste" des sources. En effet, ils fournissent un score de fusion élevé si au moins une des valeurs des sources est élevée. Ces opérateurs s'apparentent au "OU" logique puisque qu'il suffit d'une seule valeur non nulle pour obtenir un score non nul. Ils sont définis comme suit :

  Définition 6.2. *Un opérateur $\mathcal{A}$ sera dit disjonctif si et seulement si :*

$$\mathcal{A}(s_1,\ ...,\ s_n) \geq \max(s_1,\ ...,\ s_n) \tag{6.14}$$

- **Opérateurs de compromis** : Comme leur nom l'indique, ces opérateurs tentent de définir un compromis entre les différentes sources. Ceux-ci cherchent à atténuer l'influence d'un faible score par un score plus élevé et vice-versa. Finalement, le résultat de la fusion sera compris entre les valeurs minimum et maximum des différentes sources. Ils sont définis comme suit :

  Définition 6.3. *Un opérateur $\mathcal{A}$ sera dit de compromis si et seulement si :*

$$\min(s_1,\ ...,\ s_n) \leq \mathcal{A}(s_1,\ ...,\ s_n) \leq \max(s_1,\ ...,\ s_n) \tag{6.15}$$

En général, l'heuristique choisie pour l'agrégation des données définiet la classe de l'opérateur de fusion qui lui est associée. Étant donné que nous n'utiliserons, pour notre fusion, que ces trois types d'opérateurs, nous invitons le lecteur à lire les travaux de Capitaine [23] pour une description précise des autres opérateurs d'agrégations.

Choix des opérateurs de fusion

Afin de fusionner nos informations de manière robuste, nous nous sommes intéressés à cinq opérateurs basés sur des heuristiques différentes. Certaines d'entre elles sont déconnectées des données tandis que d'autres utilisent des valeurs de confiance ou des mesures

de conflit envers les sources. Les opérateurs choisis sont définis comme suit (voir aussi figure 6.10) :

1. **Agrégation par minimum :** Avec ce premier opérateur, nous avons choisi comme heuristique : "nous ne faisons pas confiance aux sources de données". Dans ce cadre, nous définissons une agrégation "pessimiste" correspondant à la valeur minimum parmi les sources :
$$\mathcal{A}(s_1,\ ...,\ s_n) = \min(s_1,\ ...,\ s_n) \tag{6.16}$$
Ainsi, nous garantissons une fusion "prudente". La distance entre l'objet et la requête est alors la distance la plus élevée calculée avec les différents descripteurs. C'est un opérateur "conjonctif" (voir figure (a)).
2. **Agrégation par maximum :** Ce second opérateur est l'inverse du précédent. L'heuristique est la suivante : "nous faisons confiance aux sources de données". Ainsi, l'agrégation devient "optimiste" et est définie comme suit :
$$\mathcal{A}(s_1,\ ...,\ s_n) = \max(s_1,\ ...,\ s_n) \tag{6.17}$$
Avec cet opérateur, le résultat de la fusion est la distance minimale entre l'objet et la requête, ceci en utilisant chacun de nos descripteurs. C'est un opérateur "disjonctif" (voir figure (b)).
3. **Agrégation par moyenne :** Cet opérateur de fusion est basé sur une heuristique donnant autant de poids à chaque source. L'heuristique est la suivante : "nous avons une confiance identique envers chaque source de données". L'opérateur d'agrégation est le suivant :
$$\mathcal{A}(s_1,\ ...,\ s_n) = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{6.18}$$
Ici, le résultat de la fusion sera égal à la moyenne des distances, obtenues avec chaque signature, entre la requête est l'objet. Cet opérateur est "de compromis" (voir figure (c)).
4. **Agrégation par moyenne pondérée :** Pour ce quatrième opérateur, nous avons voulu améliorer l'opérateur "de compromis" précédent en lui ajoutant une valeur de confiance pour chaque source. L'heuristique devient : "nous avons une confiance différente envers chaque source de données". Notons $confiance(s_i)$ la confiance que nous avons envers la valeur s_i de la source de données i, alors l'opérateur d'agrégation devient :
$$\mathcal{A}(s_1,\ ...,\ s_n) = \frac{1}{\sum_{i=1}^{n} confiance(s_i)}\sum_{i=1}^{n} confiance(s_i) * s_i \tag{6.19}$$
si les valeurs de confiance sont normalisées comme suit :
$$\sum_{i=1}^{n} confiance(s_i) = 1 \tag{6.20}$$

nous obtenons alors :

$$\mathcal{A}(s_1,\ ...,\ s_n) = \sum_{i=1}^{n} confiance(s_i) * s_i \tag{6.21}$$

Avec cette définition, l'opérateur reste "de compromis" et le résultat de la fusion est une moyenne pondérée entre les distances des différentes sources de données. Dans notre implémentation, nous utilisons les valeurs de confiance suivante : 0.2 pour les intersections, 0.4 pour les convexités/concavités avec histogrammes, 0.6 pour les cartes de normales et les convexités/concavités complètes (voir figure (d)).

5. **Agrégation par moyenne et conflit :** Le dernier opérateur de fusion tente de résoudre deux problèmes. Le premier est le point faible du précédent opérateur, à savoir : le choix des valeurs de confiance envers les sources. Le second et le caractère non adaptatif des quatre opérateurs proposés. En effet, quelle que soit la valeur des données, le traitement et toujours le même. Dans ce cadre, nous proposons une nouvelle heuristique : "lorsque deux sources sont en conflit, il est préférable de favoriser la plus sûre". Pour mettre en place cet opérateur, il est nécessaire de définir une mesure de conflit. Nous choisissons le conflit entre deux valeurs s_i et s_j issues de deux sources de données comme :

$$conflit(s_i,\ s_j) = \|s_i - s_j\|_1 \tag{6.22}$$

alors si on ordonne les valeurs s_i par ordre croissant de confiance on obtient comme opérateur d'agrégation :

$$\mathcal{A}(s_1,\ ...,\ s_n) = u_n \tag{6.23}$$

avec :

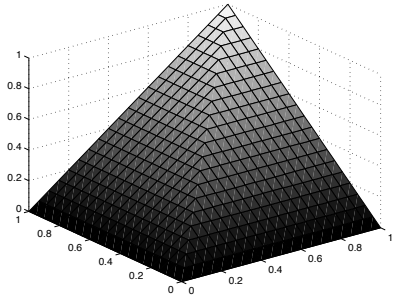
$$\begin{cases} u_1 = s_1 \\ \forall i \geq 1, \quad u_{i+1} = \frac{u_i + (1 + conflit(s_i,\ s_{i+1})) * s_{i+1}}{2 + conflit(s_i,\ s_{i+1})} \end{cases} \tag{6.24}$$

Cet opérateur "de compromis" est une moyenne pondérée où les pondérations sont fonction du conflit entre les sources. Plus les sources sont en conflits plus la moyenne va "pencher" vers les sources les plus fiables. Le grand intérêt de ce dernier opérateur d'agrégation est sa simplicité de mise en place. En effet, il ne nécessite que de savoir ordonner les sources selon leur confiance (voir figure (e)).

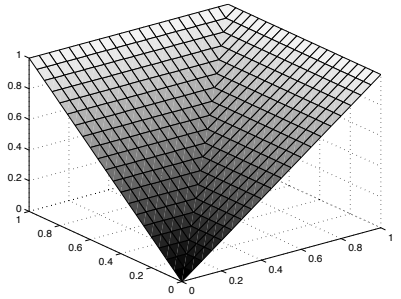
Finalement, la liste de résultats retournée à l'utilisateur s'obtient de la même façon que pour le procédé classique présenté dans la section 6.4 où la distance d est le résultat de la fusion soit : $1 - \mathcal{A}(s_1,\ ...,\ s_n)$.

6.5.2 Optimisation par élagage

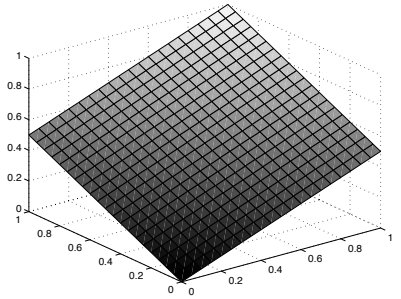
L'un des critères de recherche important pour l'utilisateur est la rapidité avec laquelle le système répond à sa requête. Pour traiter cette contrainte, la manière la plus simple est d'offrir un descripteur synthétique associé à une distance de type L_1 par exemple. Cependant, ce type de solution engendre deux problèmes. Le premier est la pertinence de



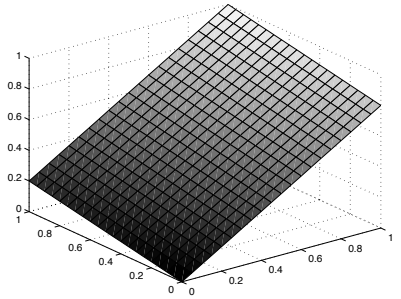
(a) *Opérateur d'agrégation par minimum*



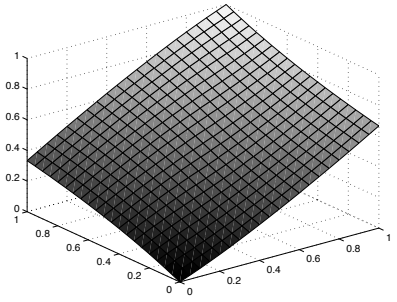
(b) *Opérateur d'agrégation par maximum*



(c) *Opérateur d'agrégation par moyenne*



(d) *Opérateur d'agrégation par moyenne pondérée*



(e) *Opérateur d'agrégation par moyenne et conflit*

FIGURE 6.10 – *Illustration des cinq opérateurs d'agrégation choisis. (a), (b), (c), (d) et (e) représentent respectivement le minimum, le maximum, la moyenne, la moyenne pondérée, la moyenne associée au conflit.*

la signature. En effet, qualité et espace de stockage sont souvent liés induisant, de ce fait, que les descriptions synthétiques ne caractérisent que grossièrement la forme d'un modèle 3D. Le second problème est la pertinence de la mesure de dissimilarité associée. En effet, même si la caractérisation compacte de la forme est robuste, il est nécessaire de l'associer au calcul rapide d'un degré de dissemblance. Du fait de cette rapidité, la mesure n'est

souvent que peu discriminante et ne permet donc pas de renvoyer des résultats pertinents à l'utilisateur.

Afin d'offrir à l'utilisateur un système de recherche alliant à la fois robustesse des résultats et rapidité, nous nous sommes tournés vers une technique d'élagage précoce des objets peu pertinents. Dans celle-ci, la vitesse est obtenue à travers l'utilisation d'un descripteur au pouvoir discriminant limité, mais compact et rapide à comparer. Tandis que la pertinence des résultats est garantie par une caractérisation plus fine de la forme des objets 3D nécessitant une comparaison plus complexe. Le premier descripteur va permettre d'établir une liste grossière de résultats. Ensuite, les distances associées aux objets les plus proches de la requête de l'utilisateur ($rang < k$) vont être réordonnés au moyen du second descripteur offrant une plus grande robustesse (voir figure 6.11).

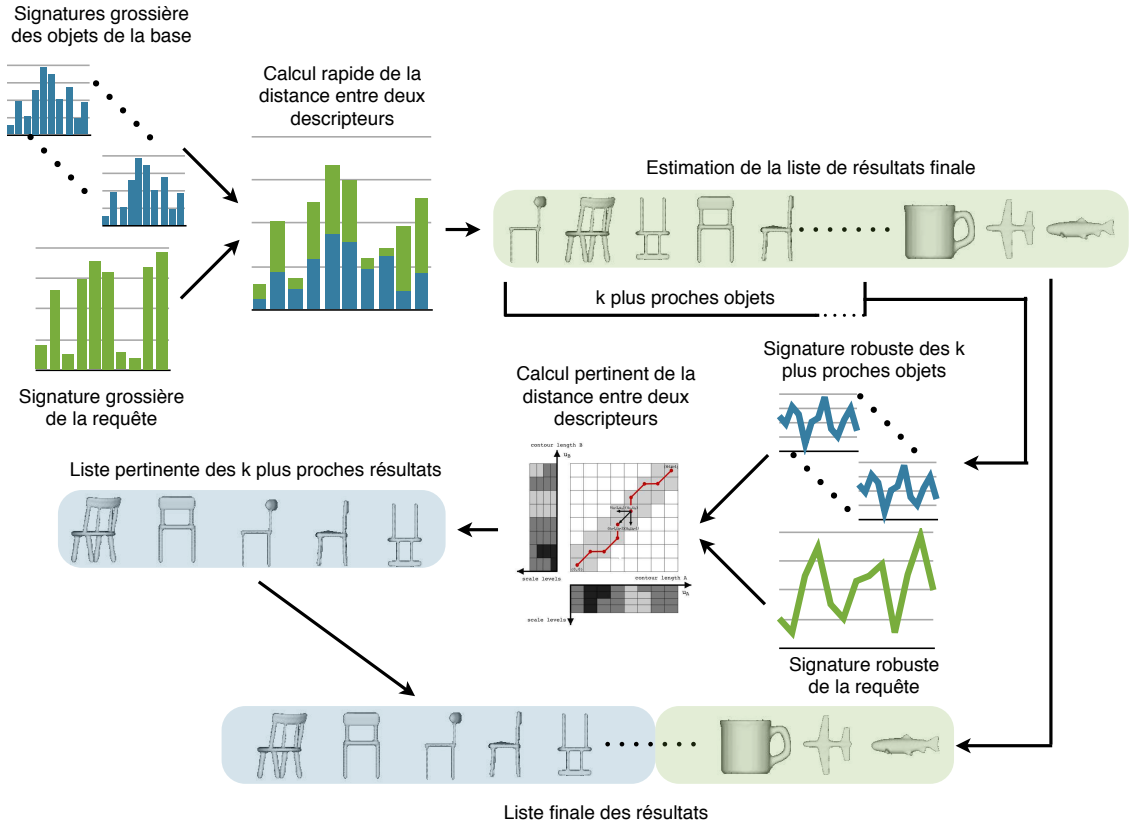


FIGURE 6.11 – *Schéma présentant un processus de recherche utilisant l'optimisation par élagage précoce. Dans cette figure, nous supposons que la complexité de la mesure de dissimilarité du descripteur par histogramme est inférieure à celle du descripteur par vecteur de données. Ainsi, une première liste de résultats est obtenue par le descripteur grossier. Ensuite, un affinement des k premiers objets est effectué par le descripteur offrant une plus grande robustesse.*

Appelons faible et fort deux descripteurs différents $\mathcal{D}_1$ et $\mathcal{D}_2$ associés à deux mesures de dissimilarité Md_1 et Md_2, dont les complexités sont respectivement $\mathcal{O}(n1)$ et $\mathcal{O}(n2)$ avec $n1 < n2$. Ainsi, le processus permettant l'élagage peut être formulé comme décrit dans l'algorithme 6.3.

Algorithme 6.3 *Algorithme utilisant un élagage précoce pour accélérer le calcul des objets les plus proches de la requête d'un utilisateur.*

1: **Pour** $i = 1$ à taille_base **Faire**
2: résultats[i] = Calculer la distance entre $\mathcal{D}_1(\mathcal{O}_i)$ et $\mathcal{D}_1(\mathcal{O}_r)$
3: **Fin Pour**
4: Ordonner par ordre croissant le tableau de résultats
5: **Pour** $j = 1$ à k **Faire**
6: l = indice de l'objet placé au rang j
7: résultats[j] = Calculer la distance entre $\mathcal{D}_2(\mathcal{O}_l)$ et $\mathcal{D}_2(\mathcal{O}_r)$
8: **Fin Pour**
9: Ordonner par ordre croissant les k premières cases du tableau de résultats
10: Renvoyer la liste de résultats à l'utilisateur

Notons $\mathcal{T}(\mathcal{D}_1)$ et $\mathcal{T}(\mathcal{D}_2)$ le temps nécessaire à la comparaison de deux descripteurs $\mathcal{D}_1$) ou $\mathcal{D}_2$). Alors Le temps de recherche pour une base de données de n objets est : $n * \mathcal{T}(\mathcal{D}_1) + k * \mathcal{T}(\mathcal{D}_2)$.

Nous appelons "élagage précoce", de $\mathcal{D}_2$ par $\mathcal{D}_1$, cette accélération car elle permet de mettre rapidement de coté – laisser à la fin de la liste des résultats – les objets peu intéressants pour l'utilisateur. En effet, grâce à l'utilisation d'un descripteur de forme grossier, nous sommes en capacité de concevoir une première estimation de la liste. Ainsi, il est possible de consacrer plus de temps à la comparaison des objets proches de la requête, en terme de forme. Le choix de la frontière entre les modèles pertinents ou non pertinents pour l'utilisateur se fait de manière rigide par le biais d'un seuil k à fixer. Il est utile de noter que ce paramètre permet de jouer sur le compromis vitesse/rapidité sous-jacent à cette optimisation. Afin de répondre du mieux possible aux besoins de l'utilisateur, il peut être intéressant de lui laisser le choix du réglage de ce paramètre (voir aussi remarque 6.5). Finalement, notons que la valeur du seuil k peut facilement être choisie si l'on connaît la taille t de la liste de résultats, qui va être retournée à l'utilisateur, et la taille de la classe de la requête, $|C(\mathcal{O}_r)|$ – elle peut être estimée par la moyenne de la taille des classes dans une base de données homogène –. En effet, dans ce cas il est cohérent de choisir pour k la valeur suivante :

$$k = \alpha * \min(t, |C(\mathcal{O}_r)|) \tag{6.25}$$

Cette définition de k permet de limiter le rayon d'étude approfondie aux seuls ob-

jets intéressants l'utilisateur. Le paramètre α permet, quant à lui, de prendre en compte les faiblesses du descripteur grossier qui ne donne qu'une estimation de la distance entre les objets 3D et la requête. Un choix de $\alpha = 2$ permet de garantir un bon compromis vitesse/robustesse en considérant que les objets recherchés par l'utilisateur sont placés par le descripteur grossier dans un rayon de deux fois la taille des résultats qu'il souhaite recevoir.

Finalement, nous avons proposé une optimisation permettant d'accélérer le processus de recherche tout en garantissant une robustesse des résultats. Le compromis vitesse/robustesse peut facilement être réglé grâce au paramètre k qu'il est possible de définir automatiquement. D'autre part, un des intérêts majeurs de cette optimisation est qu'elle peut être appliquée à tout processus de recherche. En effet, la seule contrainte de ce procédé est l'utilisation de deux descriptions de forme ayant des complexités différentes. Pour conclure, l'optimisation que nous avons développée permet de passer plus de temps sur la comparaison des objets ayant une ressemblance forte, avec la requête, que sur les modèles 3D qui en sont éloignés.

> **Remarque 6.5.** *L'optimisation que nous avons proposée s'appuie sur un paramètre k permettant de jouer avec le compromis vitesse/robustesse. Le réglage de celui-ci peut être confié à l'utilisateur afin de répondre, au mieux, à ses attentes. Cependant, nous nous rendons bien compte que celui-ci n'est pas aisé car trop subjectif. Dans ce sens, nous pouvons proposer à l'utilisateur une approche en "temps contraint" où il doit désormais définir le délai t dans lequel le moteur de recherche doit lui répondre, plutôt que le paramètre k. Afin de rester cohérent avec notre propre système, il est nécessaire de définir la borne inférieure de t comme étant égale au temps qu'il faut pour établir la liste de résultats avec le descripteur grossier. Une fois cette condition posée, il est facile de comprendre que le temps restant permet de réordonner la liste de résultats, en partant du début vers la fin, jusqu'à épuisement du temps t alloué par l'utilisateur.*

6.6 Outil et prototype

Il existe actuellement de nombreux moteurs de recherche permettant d'interroger une base de données d'objets 3D dont certains sont accessibles sur Internet. Cependant, ceux-ci n'offrent qu'une ergonomie limitée et des résultats souvent peu satisfaisants. À cause de ces lacunes, les moteurs de recherches d'objets 3D sont peu utilisés et restent en marge d'autres outils permettant d'interroger des bases de données de pages Internet ou d'images, par exemple.

Dans le but de prouver la faisabilité d'un outil de recherche pertinent, rapide et ergonomique, nous avons développé un prototype fonctionnant sur un terminal mobile de type *iPhone*. Ce logiciel reprend les principales contributions de nos travaux qui visent, avant tout, à simplifier l'utilisation. Les éléments qui ont guidé la création des processus d'indexation et de recherche, de l'interface graphique ainsi que des choix techniques sont

présentés dans la section 6.6.1.

Au cours de cette thèse, nous avons eu besoin de tester, d'évaluer et d'ajuster les paramètres de nos différentes méthodes d'indexation et de recherche d'objets 3D. Dans ce but, nous avons développé et enrichi un outil complet permettant d'effectuer rapidement ces tâches, ceci à l'aide d'une interface graphique adaptée. Les fonctionnalités ainsi que les choix techniques qui ont guidé la création de ce logiciel sont discutés dans la section 6.6.2 de ce chapitre.

6.6.1 Moteur de recherche basé sur les dessins au trait

Afin de prouver la possibilité de mettre en œuvre des méthodes d'indexation et de recherche d'objets 3D pour le grand public, nous avons développé une application mobile. Le but, visé par celle-ci, est de fournir à l'utilisateur un outil lui permettant de rechercher un modèle 3D qu'il visualise mentalement. Dans ce sens, nous avons dû, lors de la conception de ce prototype, faire des choix concernant le processus de recherche, l'interface ou encore les techniques informatiques employées.

Voici une liste des contraintes que nous nous sommes fixées :

1. Une bonne ergonomie de l'interface graphique.
2. Des requêtes de type "dessin au trait" pouvant être multiples.
3. Une pertinence des résultats retournés.
4. Un temps de recherche inférieur à 1 seconde pour une base d'environ 1000 objets.
5. La visualisation des objets 3D résultats.
6. Le stockage de la base de données, de signatures et d'objets 3D, sur un serveur distant.
7. Une utilisation sur terminal mobile.

Processus d'indexation et de recherche

Les processus d'indexation et de recherche, que nous embarquons dans notre application, doivent satisfaire les contraintes 2, 3 et 4 de la liste précédente. Dans ce sens, nous utilisons des requêtes de type "dessin au trait" associées à une caractérisation de la forme par convexités/concavités codée dans les deux signatures proposées, à savoir histogrammes et valeurs complètes (voir chapitre 5). Ainsi, l'utilisateur peut dessiner une représentation de sa requête mentale qui sera alors traitée avec le processus de la section 6.3.3.

Les aspects robustesse et rapidité sont résolus en utilisant l'optimisation introduite dans la section 6.5.2. Grâce à celle-ci et au régalage du paramètre k, il est possible de garantir une bonne robustesse, venant du descripteur fort, ainsi qu'une grande rapidité, venant de l'élagage précoce par descripteur faible. Dans notre implémentation, nous utilisons $k = 50$ comme compromis pertinence/vitesse (voir remarque 6.6). Finalement, en

utilisant un processus similaire à celui proposé dans la section 6.4, nous sommes en mesure de renvoyer à l'utilisateur une liste d'objets 3D ressemblant à sa requête.

> **Remarque 6.6.** *Afin de garantir à coup sûr un temps inférieur à la seconde, il aurait pu être intéressant d'employer une approche par "temps contraint" (voir remarque 6.5). En effet, ainsi, le paramètre k correspondrait au nombre d'objets que nous pouvons comparer en un temps de 1 seconde moins le temps de comparaison du descripteur faible (MCC-H). Ainsi, la valeur de k varie en fonction du temps restant, conditionné par la puissance de calcul du serveur ainsi que par la complexité du descripteur fort (MCC).*

L'interface graphique

L'interface graphique de notre application doit satisfaire les contraintes 1, 5 et 7 de notre liste. Dans ce cadre, nous avons choisi le terminal mobile *iPhone*[2] d'*Apple* car il permet d'offrir une grande ergonomie à l'utilisateur, ceci grâce à son interface multi-tactile[3].

Nous avons voulu organiser cet outil en quatre vues. La première permet de gérer la liste de "dessins au trait" en permettant l'ajout, la modification ou encore l'analyse de ceux-ci. La seconde offre à l'utilisateur une surface de dessin où il est possible d'esquisser, d'effacer, d'annuler ou de valider un "dessin au trait". La troisième, quant à elle, permet de visualiser la liste des résultats fournis par le serveur, celle-ci comporte une visualisation 2D de l'objet ainsi que sa classe si elle est disponible. Finalement, la quatrième vue offre un visualisateur 3D où il aisé de déplacer, tourner où changer l'échelle du modèle observé. La figure 6.12 présente le schéma du fonctionnement global de notre outil.

Avec un tel découpage, l'utilisateur comprend rapidement ce qu'il peut faire et navigue facilement entre les différentes vues de l'interface. Chacune d'entre elles a un nombre limité de possibilités garantissant une bonne ergonomie. L'implémentation de ce modèle avec l'environnement de l'*iPhone* est présentée dans la figure 6.13.

Afin de prendre en compte toutes les possibilités du modèle présenté dans la figure 6.12, nous proposons à l'utilisateur des surfaces actives, ou boutons, dans chacune des vues. La figure 6.14 présente les éléments permettant à l'utilisateur d'interagir avec l'interface graphique du prototype.

Choix techniques

Le développement de notre moteur d'indexation et de recherche d'objets 3D s'est fait en accord avec les contraintes introduites au début de cette partie. En particulier, nous avons orienté sa conception pour obtenir un outil rapide et fonctionnel. Dans ce sens, nous avons dû faire des choix techniques permettant de mettre au point la communication avec

2. http://www.apple.com/fr/iphone/
3. On utilise fréquemment le terme anglais *multi-touch* pour parler de cette technologie.

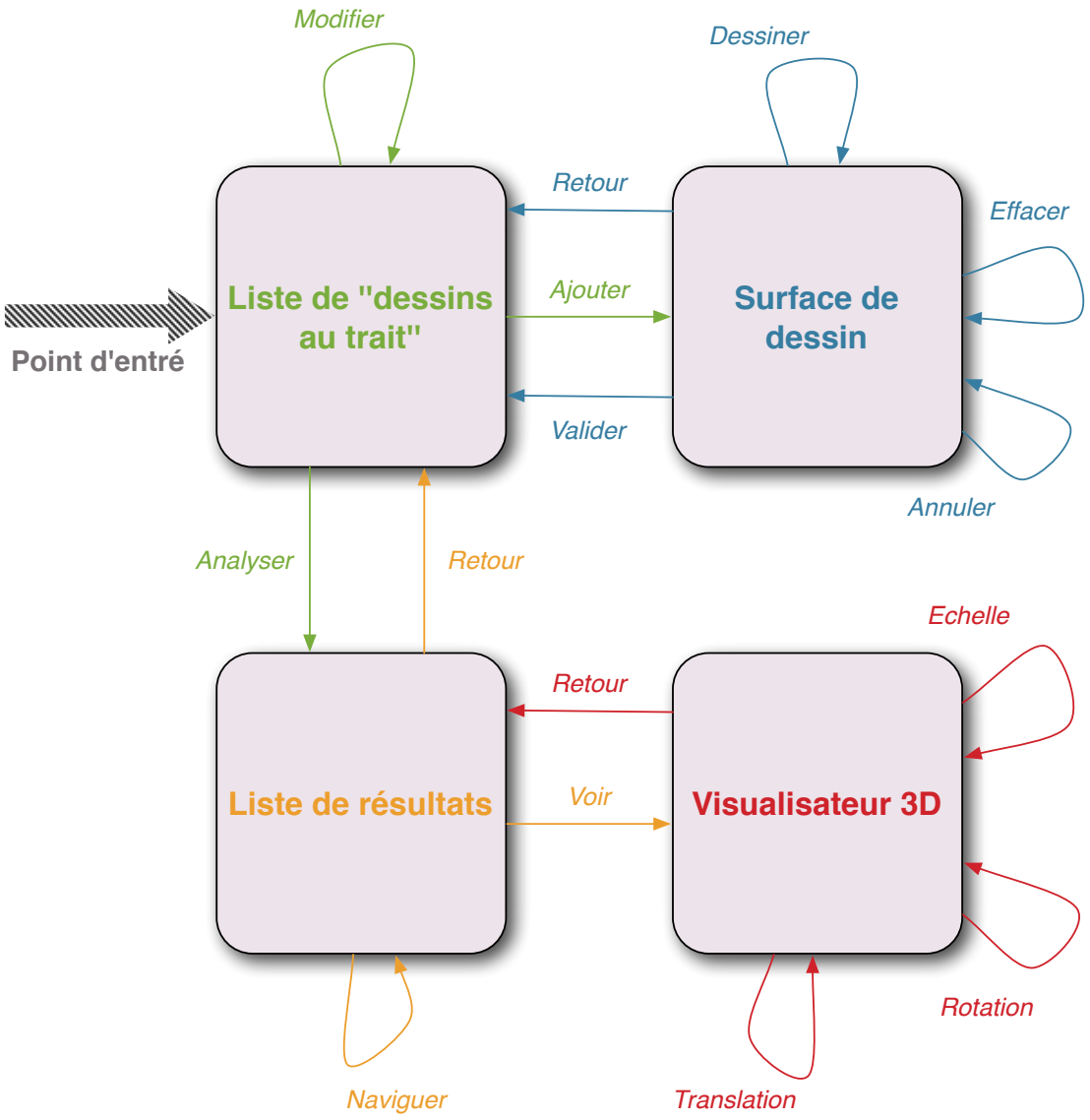


FIGURE 6.12 – *Schéma reprenant le fonctionnement global de notre prototype. Les quatre vues développées plus haut sont respectivement de couleur verte, bleue, orange et rouge. Les flèches représentent ici les possibilités d'interaction avec l'interface graphique.*

le serveur de fichier, la surface de dessins, la liste de résultats ainsi que le visualisateur 3D.

La plateforme de test étant un *iPhone*, nous avons dû nous plier aux contraintes de développement qu'elle engendre. En particulier, nous avons implémenté l'ensemble de l'interface graphique ainsi que les processus d'indexation et de recherche dans le langage *Objective-C*[4] associé à la bibliothèque *Cocoa*[5]. La partie communication avec le serveur

4. http://developer.apple.com/
5. http://developer.apple.com/

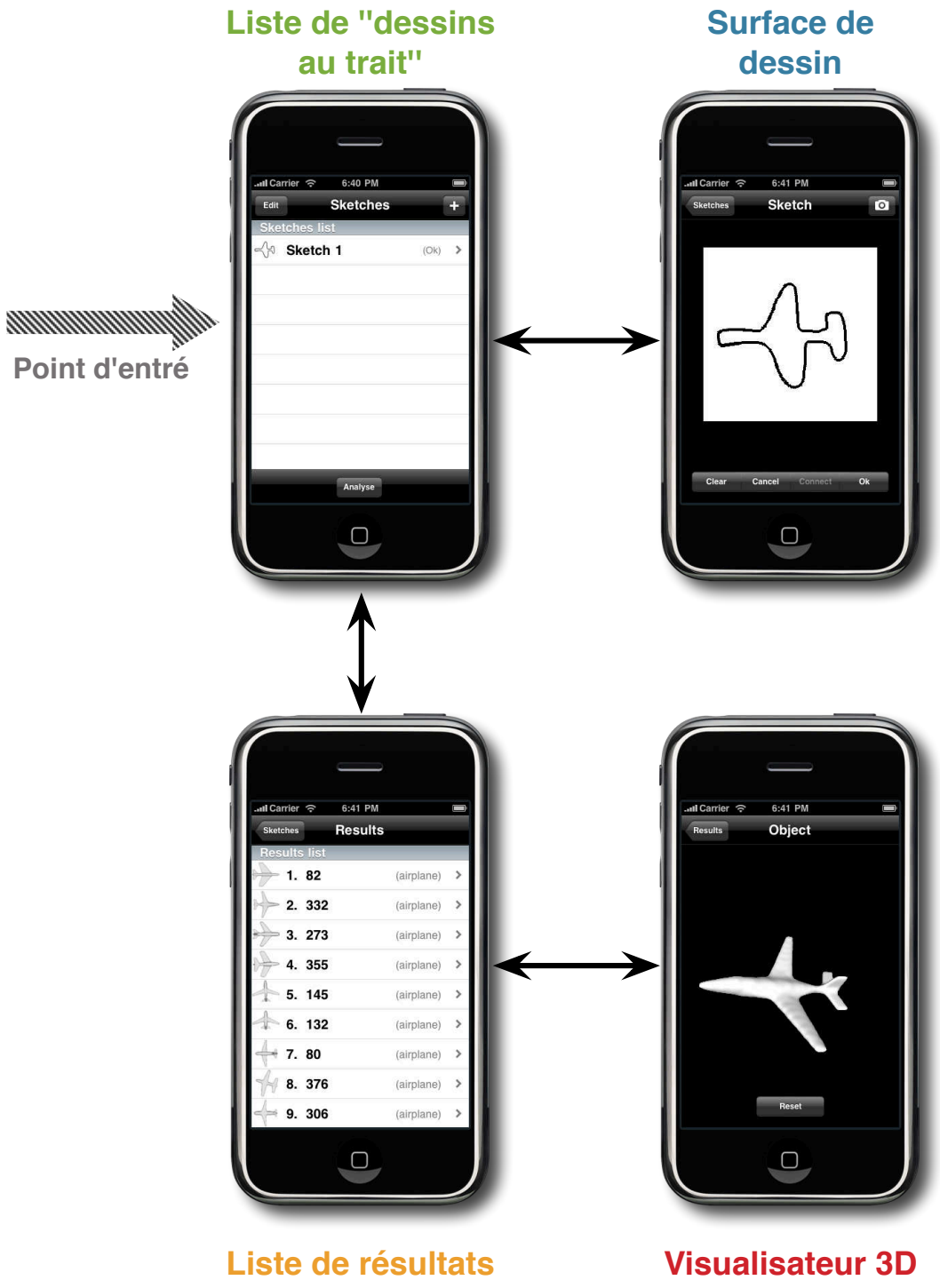
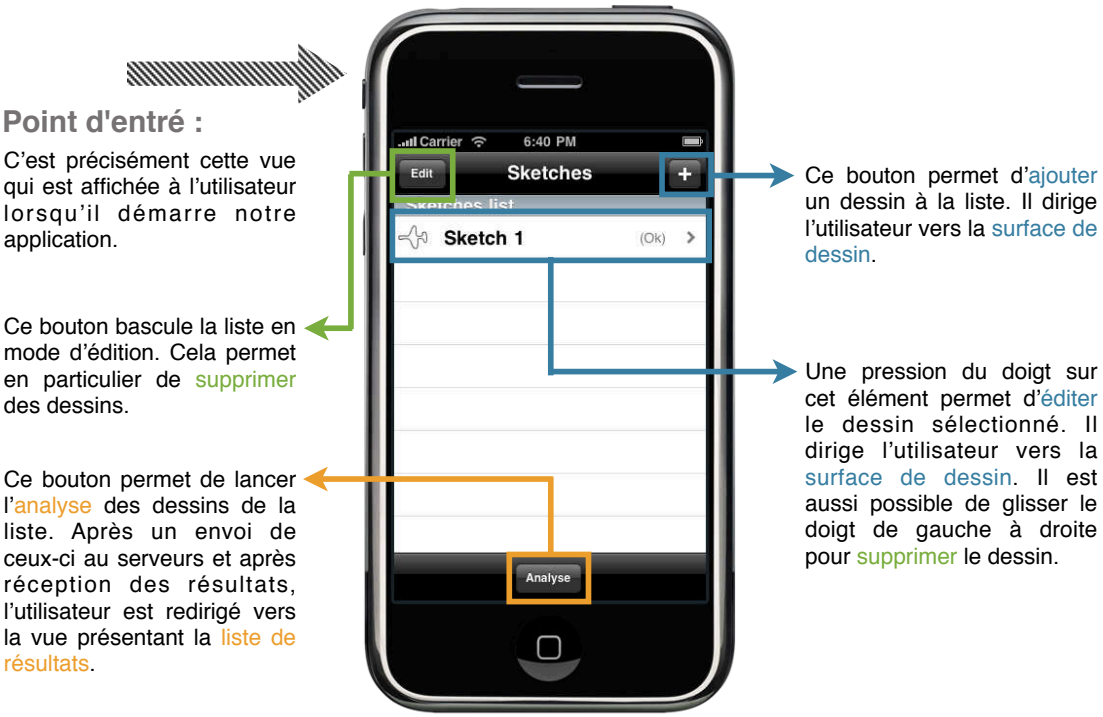
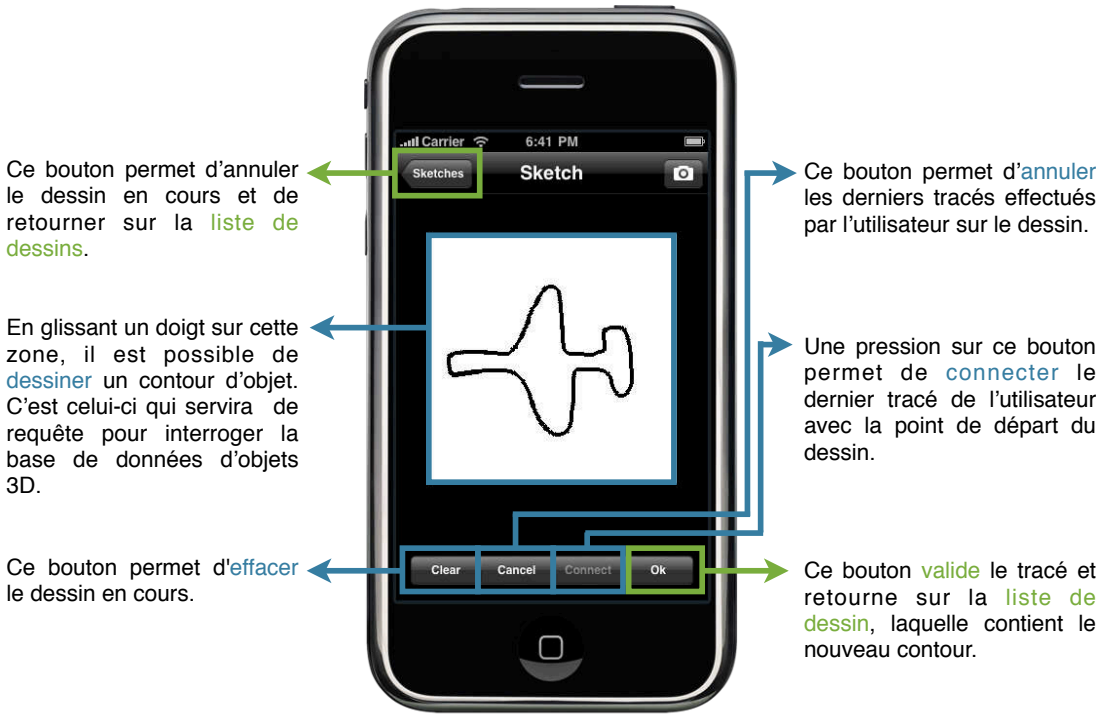


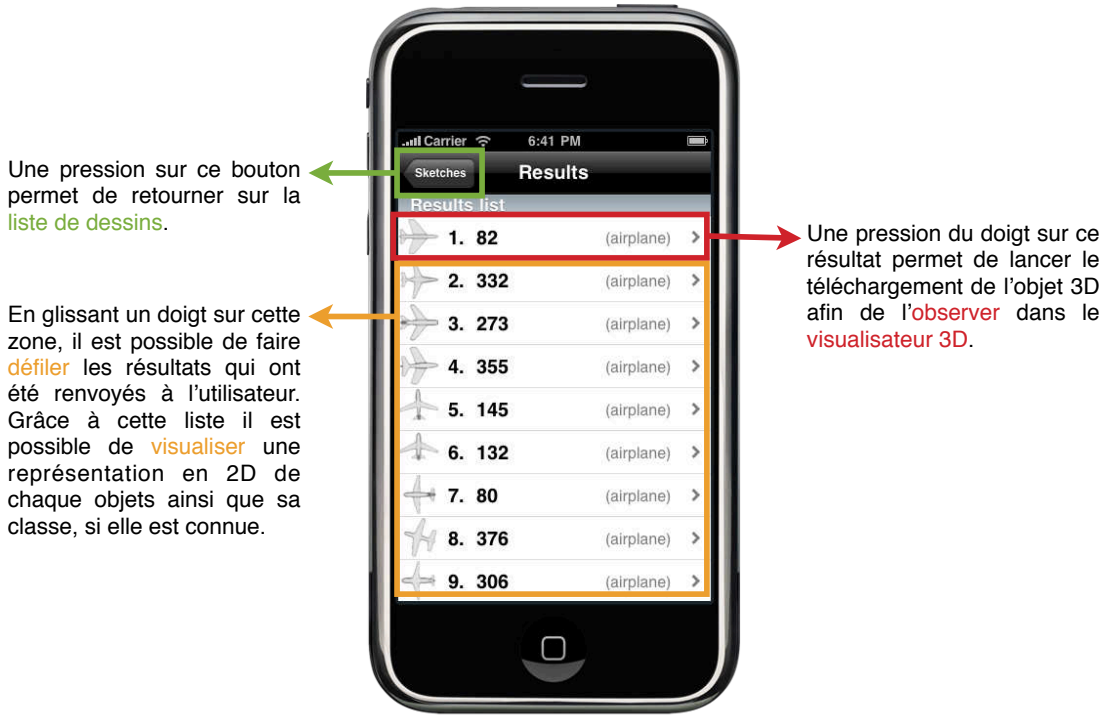
FIGURE 6.13 – *Implémentation du modèle proposé dans la figure 6.12 sur un terminal mobile de type iPhone. Ici, les quatre vues sont organisée de la même façon que sur le schéma de la figure 6.12.*



(a) *Liste de dessins*



(b) *Surface de dessin*



(c) *Liste de résutlats*

s'est fait par l'utilisation de *socket* permettant d'interroger à distance la base de données et de récupérer les images 2D des objets (pour la liste de résultats) ainsi que les objets 3D (pour le visualisateur 3D). L'ensemble de ces communications peuvent s'effectuer à travers les réseaux mobiles *Edge* et *3G* ainsi que par *Wifi*. Finalement, la surface de dessins, d'une résolution de $256 * 256$ pixels, ainsi que le visualisateur 3D s'appuie sur la technologie d'accélération graphique *OpenGL ES*[6].

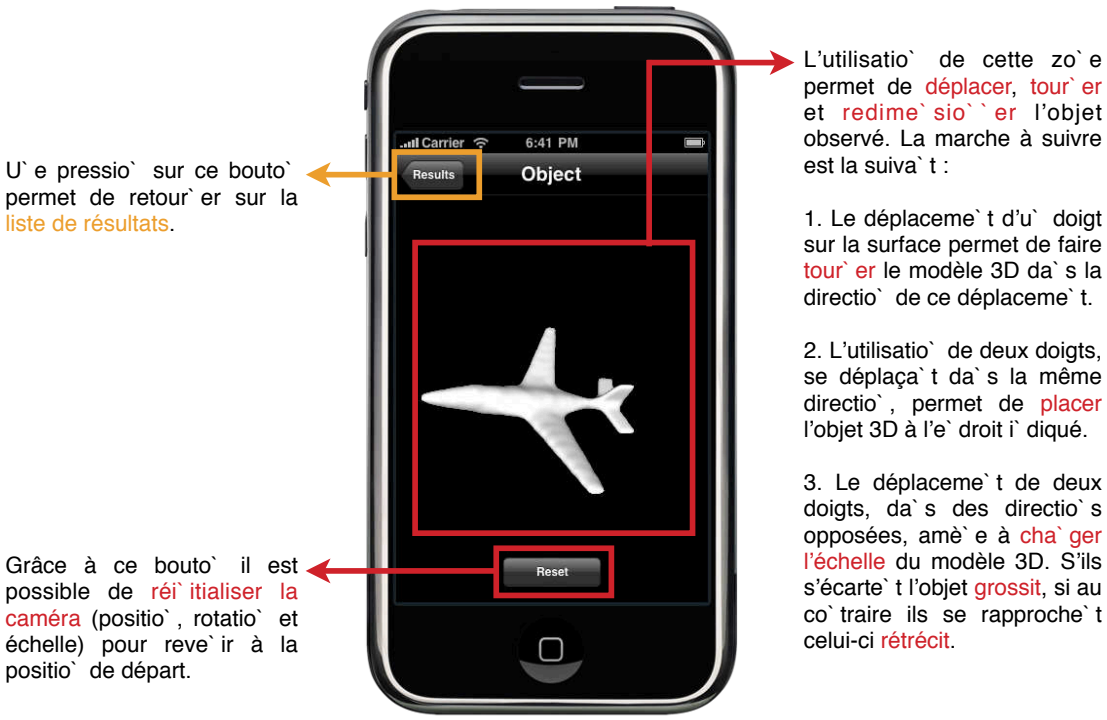
Pour conclure, c'est grâce à l'ensemble de ces choix techniques que notre prototype est fonctionnel, facile de prise en main et adapté à l'utilisateur.

6.6.2 Moteur d'Indexation et de Recherche d'Objets 3D "Miro3D"

L'étude et la conception de méthodes d'indexation et de recherche d'objets 3D s'appuient sur des expérimentations logicielles. En effet, afin de mettre au point des techniques pertinentes, il est nécessaire de les tester, de les ajuster, de les modifier... Dans ce cadre, nous nous sommes posé la question des outils nécessaires pour effectuer ces tâches plus ou moins complexes.

Nous présentons donc dans cette partie les outils que nous avons utilisés et en particulier l'application que nous avons développée. Nous expliquerons l'enjeu de cet outil ainsi

6. http://www.khronos.org/opengles/



(d) *Visualisateur 3D*

FIGURE 6.14 – *Cette figure explique en détail les possibilités d'interaction avec notre outil d'indexation et de recherche d'objet 3D. (a) montre la vue présentant la liste de dessins. (b), quant à elle, explique les zones activent de la surface de dessins. La figure (c) présente la liste de résultats ainsi que ses actions associées. Finalement, (d) illustre les interactions avec l'objet dans le visualisateur 3D.*

que les possibilités qu'il offre avant de conclure avec les choix techniques qui l'ont guidé.

Pourquoi un tel outil ?

Étant donné qu'il n'existe pas de logiciel spécialisé pour l'indexation et la recherche d'objet 3D, nous nous sommes confrontés à des outils permettant la mise au point de certains algorithmes, mais qui était trop souvent inadaptée à l'univers 3D. Afin de tirer parti des avantages de ces logiciels sans pour autant négliger l'aspect représentation 3D, nous avons opté pour une démarche bipartite. La première a été l'élaboration et le test de nos différents algorithmes à l'aide du logiciel *Matlab*[7] qui offre un prototypage rapide grâce à une grande variété de fonctions. La seconde, quant à elle, a été le développement d'une application permettant l'indexation et la recherche d'objets 3D.

Cette application nommée "Miro3D" pour Moteur d'Indexation et de Recherche d'Ob-

7. http://www.mathworks.fr/

jets 3D offre entre autres des possibilités de réglage des algorithmes (voir figure 6.15). Son rôle est d'offrir des moyens simples et rapides d'extraire les descriptions de forme introduite au chapitre 5. Elle propose aussi un moteur de recherche permettant d'interroger une base de données de signatures en utilisant l'ensemble des techniques proposées dans ce chapitre. C'est en particulier grâce à cet outil que nous avons pu évaluer notre processus complet d'indexation et de recherche d'objet 3D. Même si notre outil offre de nombreuses possibilités, il n'a pas été développé pour un utilisateur, comme notre prototype sur *iPhone* (voir section 6.6.1), mais pour le chercheur.

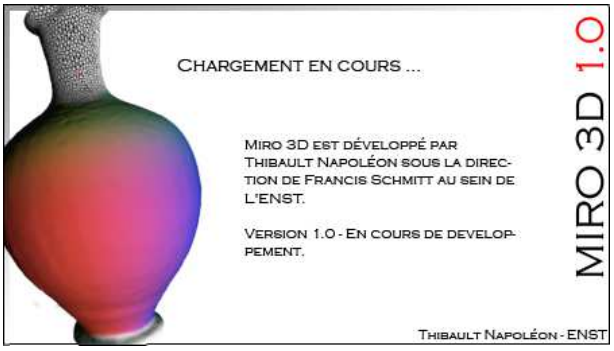


FIGURE 6.15 – *Fenêtre affichée durant le lancement de notre application.*

Inventaire des possibilités de l'outil

L'application que nous avons développée s'est enrichie au jour le jour au cours de notre thèse. De ce fait, elle possède de nombreuses possibilités qu'il serait peu intéressant d'énumérer de façon exhaustive. Ainsi, nous dressons une liste des fonctionnalités principales. Des captures d'écran de cette application sont visibles dans la figure 6.16.

- Chargement d'une liste d'objets 3D à travers un fichier contenant leur liste. Support des formats 3D ".off", ".stl", ".tri" et ".wrl".
- Choix de la normalisation parmi : notre méthode (voir chapitre 4), centre de gravité, boite ou sphère englobante, "ACP", "ACP Continue" et "ACP par normales".
- Outil d'extraction des descriptions par intersection, par convexités/concavités et par cartes de normales.
- Possibilité de choisir le nombre de vues parmi celles proposées dans la section 5.2 du chapitre 5.
- Choix de la résolution des silhouettes entre $64 * 64$, $128 * 128$ et $256 * 256$ pixels.
- Mise en correspondance des trois signatures proposées.
- Optimisation par élagage précoce (voir section 6.5.2 de ce chapitre).
- Outil permettant de reconstruire l'enveloppe visuelle d'un objet à partir de trois de ses vues.
- Export des alignements, des objets 3D, des résultats... en format ".txt" et ".html" (page Internet).
- Visualisation des objets 3D.

– Options de visualisations diverses et variées incluant : volumes englobants, normales des faces et des sommets, éclairage, mode de rendu en fil de fer et type de projections.

Choix techniques

Comme pour notre prototype sur téléphone mobile, nous avons dû, lors du développement de l'application "Miro3D", faire des choix techniques. Nous avons plus particulièrement été confrontés à des questions concernant la conception globale de l'outil (langage de programmation, architecture...). Ce logiciel devant perpétuellement s'enrichir en accord avec nos travaux de recherches, c'est plus l'aspect évolution que performance qui nous a intéressés.

Le langage de programmation choisi a été le '*C++*, ceci pour son aspect *objet* qui favorise la maintenance et l'évolution du code. Afin de pouvoir utiliser notre application indifféremment sur des systèmes *Mac*, *Linux* et *Windows*, nous avons opté pour des librairies graphiques multi-plateformes. Il s'agit de *Qt*[8] pour la partie interface graphique, *OpenGL*[9] pour le rendu des objets 3D et *ImageMagick*[10] pour le traitement des images.

En ce qui concerne la partie modélisation de notre application, nous avons utilisé le concept de *plugin* afin de garantir les évolutions futures. Ainsi, il est aisé de développer de nouveaux outils qui vont alors pouvoir se greffer rapidement au logiciel existant. Coté interface graphique (voir figure 6.16), nous avons choisi un découpage en trois vues associées à deux barres d'outils. Les différentes vues permettent respectivement le rendu des objets 3D et des résultats, l'affichage de la base de données sous forme de liste et la représentation du modèle 3D étudié en deux dimensions. Les barres d'outils, quant à elles, permettent pour la première, qui est placée verticalement à gauche de l'application, de changer les attributs de rendu du modèle (éclairage, normales, volume englobant...) et pour la seconde, situé horizontalement en haut du logiciel, d'utiliser les différents outils d'indexation et de recherche d'objets 3D.

6.7 Conclusion

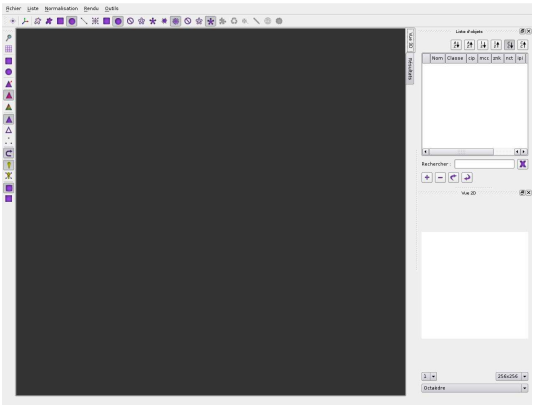
Dans ce chapitre, nous avons introduit le processus permettant à un utilisateur de rechercher un objet 3D dans la base de données. Celui-ci s'appuie sur des requêtes multimodales associées à des mesures de dissimilarité adaptées à chacun des descripteurs introduits dans le chapitre 5. Ainsi, l'utilisateur peut très facilement transcrire une représentation mentale en une requête 2D sans pour autant réduire la robustesse des résultats. L'enchaînement précis des étapes de ce processus "en ligne" à particulièrement été discuté dans les sections 6.2 et 6.4 de ce chapitre.

8. http://qt.nokia.com/
9. http://www.opengl.org/
10. http://www.imagemagick.org/

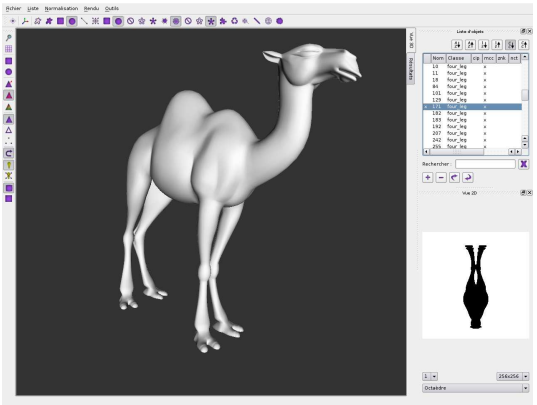
Afin de garantir la rapidité, mais aussi la pertinence de notre approche, nous avons proposé deux optimisations. La première vise à améliorer la qualité des résultats en combinant, grâce à différents opérateurs d'agrégation (voir section 6.5.1), nos différentes signatures. La seconde, quant à elle, cherche à accélérer le processus de comparaison en effectuant un élagage précoce des résultats peu pertinents pour l'utilisateur (voir section 6.5.2).

Ce chapitre étant avant tout tourné vers l'utilisateur, nous y avons aussi présenté un outil novateur en indexation et recherche d'objet 3D. Celui-ci est un prototype, démontrant qu'il est désormais possible de développer des moteurs de recherche embarqués sur des terminaux mobiles de nouvelle génération. D'une part, ses requêtes, de type "dessin au trait", lui confèrent une grande prise en main et, d'autre part, son ergonomie facilite la recherche de modèles 3D ainsi que leur visualisation.

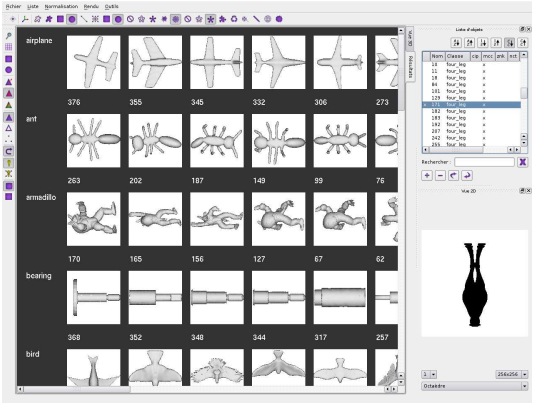
Pour conclure, ce chapitre clôture nos contributions à l'indexation et à la recherche d'objets 3D avec la partie "en ligne" du traitement. Nous avons proposé des méthodes nouvelles, principalement orientées vers l'utilisateur, mais qui s'appuie pleinement sur la phase "hors-ligne" du processus de traitement. Afin de paramétrer et d'évaluer l'ensemble de nos travaux, nous proposons dans le chapitre 7 un ensemble d'expérimentations et de résultats.



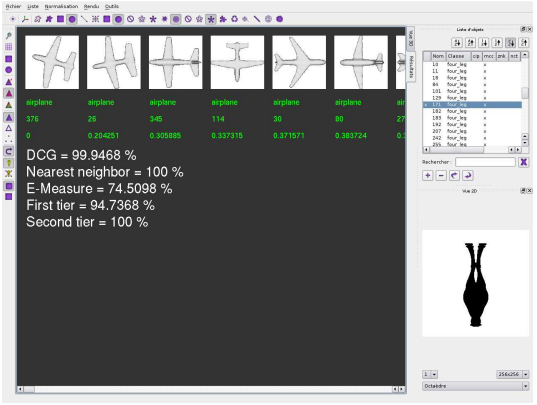
(a) *Interface après le lancement*



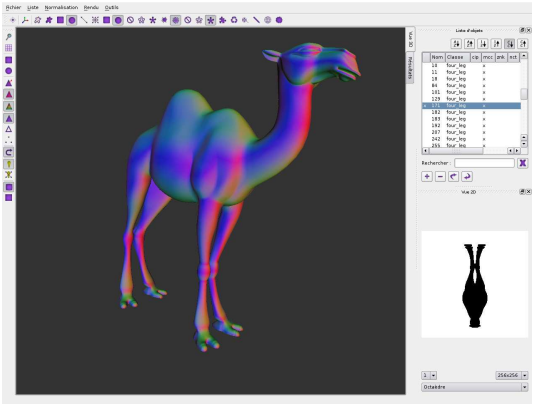
(b) *Visualisation d'un objet 3D*



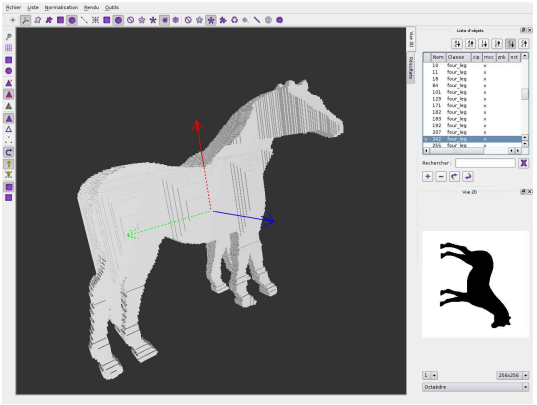
(c) *Visualisation de la base de données*



(d) *Visualisation des résultats*



(e) *Visualisation des normales*



(f) *Visualisation de l'enveloppe visuelle d'un objet*

FIGURE 6.16 – *Cette figure présente notre application "Miro3D" à travers différentes captures d'écrans.*

CHAPITRE 7

Expérimentations et résultats

Résumé *: Dans ce chapitre, nous introduisons les diverses expériences qui ont guidé nos choix dans la conception de nos approches d'indexation et de recherche d'objets 3D. Nous y présentons aussi les résultats obtenus. Finalement, nous faisons une étude comparative de nos méthodes avec les approches existantes de la littérature à travers les évaluations SHREC.*

7.1 Introduction

Les trois derniers chapitres de cette thèse ont introduit diverses contributions portant sur les prétraitements nécessaires à l'indexation et à la recherche d'objets 3D ainsi que sur ces processus eux-mêmes. Le développement de ces nouvelles méthodes s'est fait en accord avec les résultats des expérimentations et des évaluations effectuées sur différentes bases de données de modèles 3D. Celles-ci nous ont en particulier aidés à régler les paramètres de chaque approche afin de gagner en pertinence sans pour autant augmenter trop fortement le temps de recherche pour l'utilisateur. D'autre part, pour comparer nos propres résultats avec ceux d'autres équipes de recherche, travaillant sur l'indexation et la recherche d'objets 3D, nous avons participé au concours annuel *SHREC*, dont nous montrerons les résultats. Finalement, nous avons aussi comparé nos méthodes à celles présentées par le *Princeton Shape Benchmark*.

Nous débutons ce chapitre par la section 7.3 qui propose les expérimentations et les évaluations de notre contribution portant sur la normalisation et l'estimation de l'alignement. Nous montrons ensuite, dans la section 7.4, les résultats de notre méthode d'indexation et de recherche d'objets 3D. Nous nous attardons en particulier sur les trois descripteurs [1] proposés ainsi que sur nos deux méthodes d'optimisation. Finalement, une étude comparative avec d'autres méthodes de la littérature est proposée dans la section 7.5.

7.2 Bases de données et estimation de la pertinence

Notons que les bases de données de tests sur lesquelles nous avons expérimenté et testé nos approches, ainsi que les mesures utilisées pour mesurer la pertinence de nos résultats, sont proposées dans l'annexe D.

1. Nous utiliserons l'abréviation EP pour les ensembles de pixels, MCC-H et MCC pour les descripteurs basés sur les informations de convexités/concavités avec histogrammes ou non et CN les cartes de normales.

7.3 Expérimentations sur la normalisation et l'alignement

Nous avons proposé, dans le chapitre 4, une nouvelle approche permettant de normaliser et d'aligner un objet 3D. Celle-ci ayant été développée en accord avec les descripteurs introduits dans le chapitre 5, nous présentons dans la section 7.3.1 les expérimentations visant à valider notre contribution. D'autre part, afin de confronter notre méthode avec la littérature existante, nous dressons dans la section 7.3.2 une comparaison de notre approche avec d'autres, sur les aspects de normalisation et d'alignement d'objets 3D.

7.3.1 Normalisation

Lors de la conception de notre approche de normalisation, en centre et en échelle, d'un objet 3D, nous avons défini deux contraintes portant sur les silhouettes du modèle (voir section 4.2.1). La première est l'obtention d'un contour de périmètre maximal pour une résolution d'image donnée. La seconde est l'assurance que la projection contienne l'intégralité de l'objet 3D. Nous avons vu que l'utilisation de la sphère minimale englobante, associée à un processus cohérent de capture des silhouettes (voir section 5.2), permet de valider la seconde contrainte. En revanche, rien ne garantit la satisfaction de la première. Dans ce sens, nous mettons en place plusieurs expériences permettant de comprendre le comportement de notre normalisation face à d'autres méthodes dites "classiques".

En particulier, c'est face à deux autres approches, le cube englobant et le centre de gravité surfacique associé au rayon maximum (voir section 3.2 du chapitre 3), que nous confrontons notre méthode. Cette évaluation s'effectue à l'aide des valeurs d'aires et de périmètres calculés pour chaque silhouette et sommée sur chaque objet. Ici, l'aire d'une silhouette correspond au nombre de pixels de l'objet à indexer, tandis que le périmètre représente la longueur du contour externe d'une silhouette. Celui-ci est obtenu en additionnant les distances euclidiennes entre chaque point consécutif du contour.

Dans ce protocole expérimental, nous ne considérons que l'aire globale d'un objet en deux dimensions – somme des aires des trois silhouettes – et non l'aire de chacune. En effet, étant donné qu'aucune normalisation en rotation n'a été effectuée, les différentes silhouettes sont indépendantes et ne sont donc pas comparables. La base de test choisie est celle de *SHREC'07 Watertight*, présentée dans la section D, et la résolution de projection est de $256 * 256$ pixels. Les figures 7.1 et 7.2 représentent donc, respectivement, le périmètre moyen des objets de chaque classe, et l'aire de celle-ci, en comparaison aux deux autres méthodes "classiques". L'aire et le périmètre calculés globalement, sur l'ensemble de la base, sont représentés de la même façon sur la figure 7.3.

Avant même de rentrer dans une étude approfondie de ces résultats, nous pouvons noter que notre méthode répond bien au critère d'optimisation de la qualité des contours, caractérisée par la maximisation du périmètre et de l'aire des silhouettes 2D. En effet, notre méthode fournit une aire et un périmètre supérieurs aux deux autres méthodes de la littérature, et ceci, grâce à l'utilisation de la sphère englobante. Il est de plus intéressant

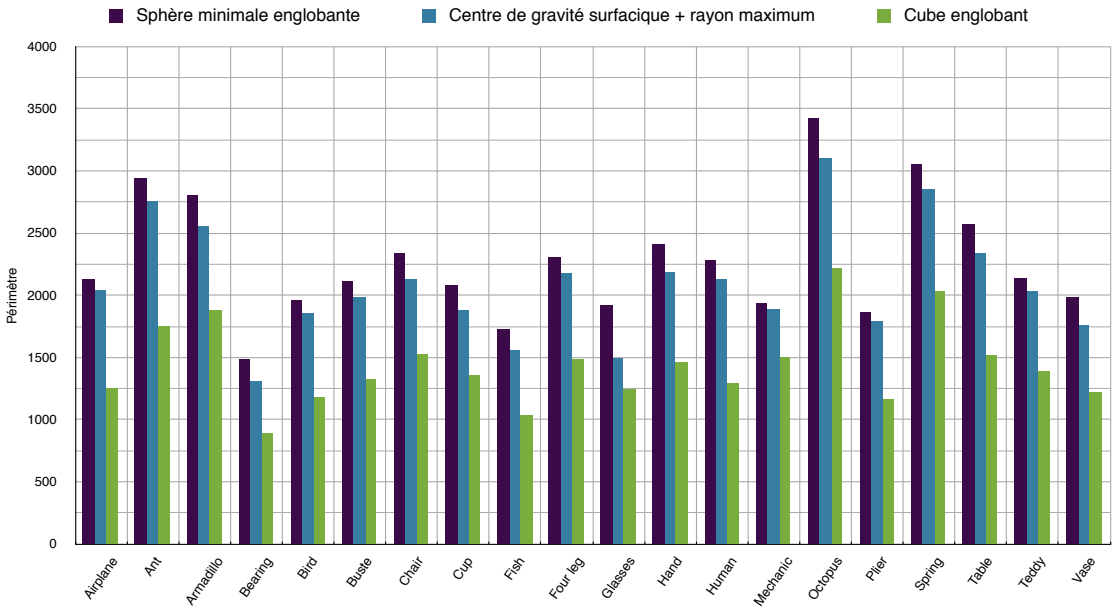


FIGURE 7.1 – *Comparaison du périmètre moyen des objets de chaque classe face à deux autres normalisations. Nous notons que la méthode de normalisation proposée surclasse clairement les autres approches, et ceci, quelle que soit la classe de l'objet.*

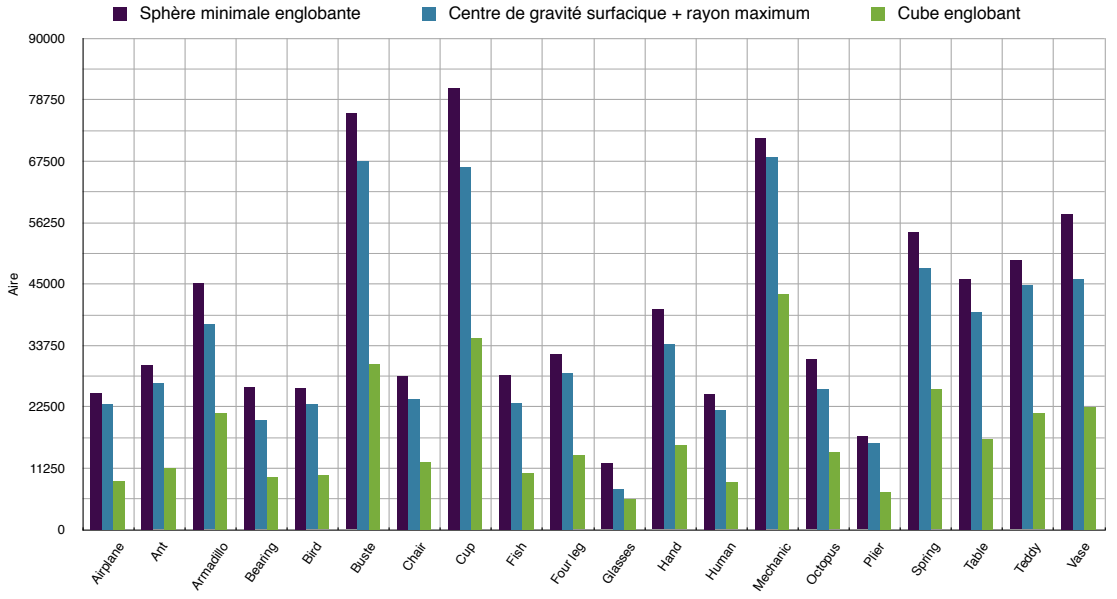


FIGURE 7.2 – *Comparaison de l'aire moyenne des objets de chaque classe face à deux autres normalisations. Nous notons que la méthode de normalisation proposée surclasse clairement les autres approches, et ceci, quelle que soit la classe de l'objet.*

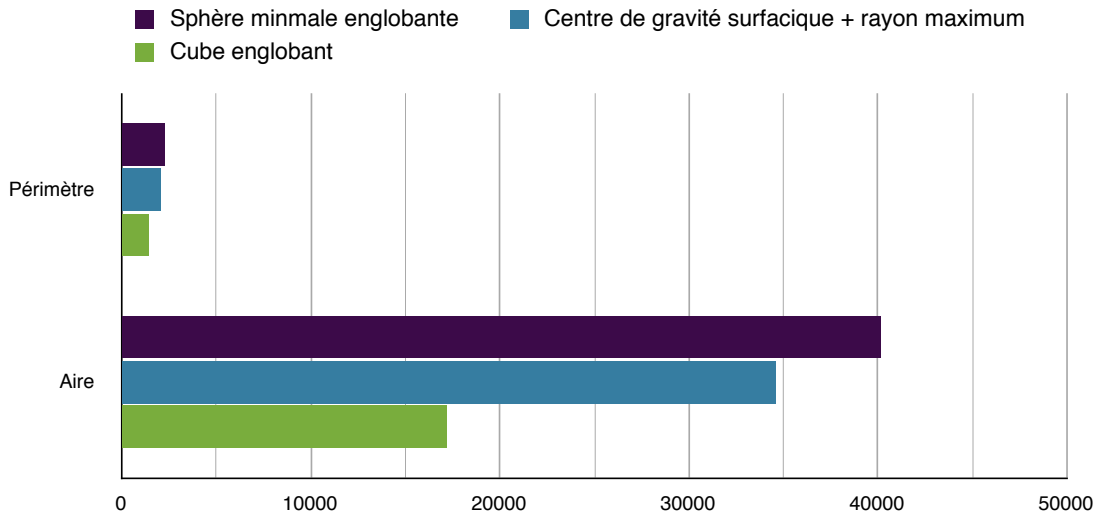


FIGURE 7.3 – *Comparaison du périmètre et de l'aire moyenne des objets de la base face à deux autres normalisations. Nous notons que la méthode de normalisation proposée surclasse clairement les autres approches, et ceci, quelle que soit la classe de l'objet.*

de noter que notre méthode de normalisation donne de meilleurs résultats quelle que soit la classe de test.

Il est clairement visible sur les différentes figures, que notre méthode donne de très bons résultats face au cube englobant, tandis que la différence entre la sphère englobante et le centre de gravité surfacique, vois section 3.2, associé au rayon maximum est moindre. Ce comportement est principalement dû au fait que le centre de gravité surfacique est proche du centre de la sphère minimale englobante, pour la plus grande partie des objets, et ainsi, le rayon maximum se rapproche du rayon de la sphère.

En ce qui concerne l'aire, nous pouvons voir sur la figure 7.2 que notre méthode de normalisation est à peine meilleure que le centre de gravité surfacique pour les classes : *Airplane, Bird, Buste, Mechanic* et *Plier*. C'est en partie parce que ces classes représentent des objets uniformément étendus autour de leur centre de gravité. Néanmoins, la figure 7.1 montre que même dans ces classes, la sphère minimale englobante augmente le périmètre des contours, parfois même, significativement, permettant de ce fait une meilleure extraction des informations qu'il contient.

Finalement, nous avons vu que notre processus de normalisation répond bien aux contraintes que nous nous sommes fixées. Grâce à l'utilisation de la sphère minimale englobante, pour déterminer le centre et l'échelle d'un objet 3D, nous garantissons une bonne qualité des contours ainsi que des projections sans perte d'information. Nous pouvons donc conclure que la méthode proposée est adaptée aux descripteurs “2.5D/3D” et “2D/3D”.

7.3.2 Estimation de l'alignement

Nous avons introduit, dans la section 4.3 du chapitre 4, une méthode permettant d'aligner un objet 3D. Celle-ci s'appuie sur un critère d'estimation définissant la pertinence d'un alignement par rapport à un autre. Nous présentons ici un ensemble de résultats montrant à quelle hauteur, en pourcentage, chaque méthode contribue à l'alignement final d'un objet.

Résultats sur la base de *SHREC'07 Watertight*

Afin valider notre approche, nous avons cherché à estimer la qualité de l'alignement des objets de la base de *SHREC'07 Watertight*. Pour cela, nous avons construit une vérité terrain où chaque modèle 3D a été aligné à la main. Grâce à celle-ci, nous testons si les objets 3D sont bien alignés, ou non, avec différentes méthodes. Cette estimation se base sur le calcul de l'angle ϵ (en radian) entre la pose obtenue et la pose correcte – durant ce calcul, nous garantissons l'invariance par rapport à l'ordre des axes ainsi qu'à leur signe –. Dans cette expérimentation, proposée dans la figure 7.5, nous regardons le pourcentage d'objets 3D bien alignés en prenant en compte une tolérance de l'angle ϵ allant de 0.1 à 0.5 radians, comme présenté sur la figure 7.5. Pour cette estimation, notre méthode choisie la meilleure pose parmi : la pose initiale, l'ACP Continue (ACPC) et l'ACP par normales (ACPN) (voir chapitre 4).

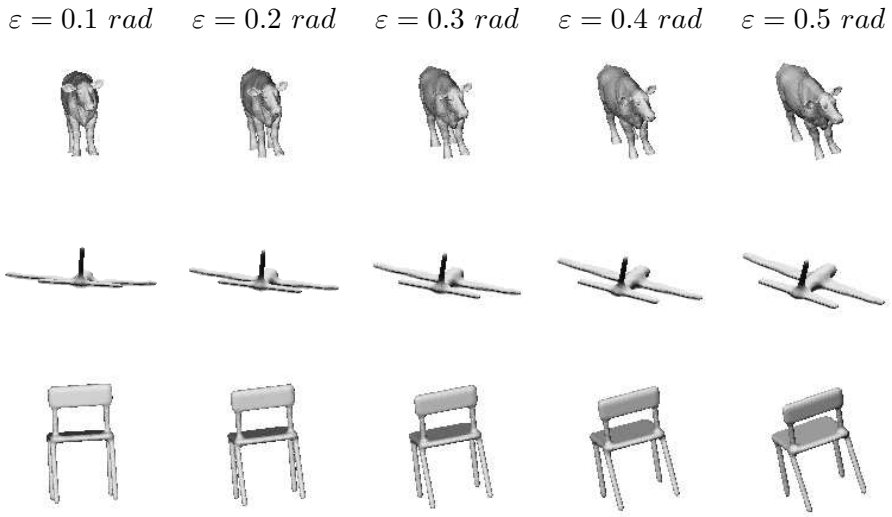


FIGURE 7.4 – *Illustration de l'alignement correct de trois objets 3D à un angle (ϵ) de tolérance près en radians. (voir aussi figure 7.5).*

Les résultats obtenus montrent clairement que notre méthode d'alignement est plus pertinente que chacune des approches testées. Logiquement, le pourcentage d'objets bien alignés augmente avec la tolérance de l'angle ϵ, et ceci, quelle que soit la méthode. D'autre part, cette première évaluation illustre bien le point fort de notre contribution, à savoir sa capacité à tirer parti de plusieurs alignements existants d'un même objet. Ainsi, même

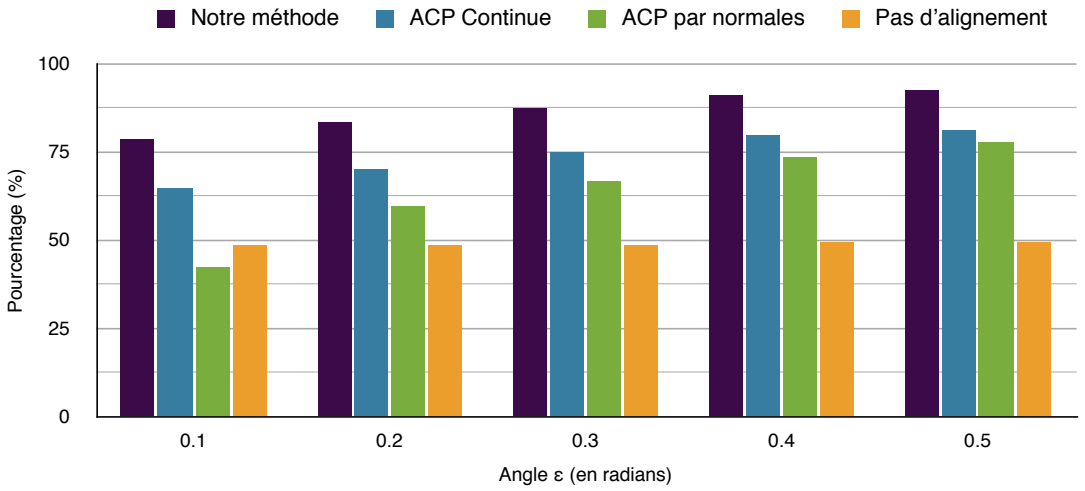


FIGURE 7.5 – *Cette figure représente le pourcentage d'alignement correct en fonction de la tolérance d'angle ϵ (en radians) pour quatre méthodes d'alignements. Cette évaluation est effectuée sur la base de SHREC'07 Watertight avec des silhouettes de* $256 * 256$ *pixels. Il est clairement visible que notre approche améliore les méthodes testées.*

si chaque méthode, prise séparément, n'est que peu pertinente, leur combinaison peut améliorer l'alignement global de la base de données. Finalement, notons que l'ACP par normales est moins pertinente que l'ACP Continue sur cette base, certainement à cause du caractère "articulé" de cette base peu adaptée à l'approche utilisant les normales.

Expérimentations sur la base de *SHREC'09 Generic*

Étant donné que nous avons proposé, dans le chapitre 4, un estimateur permettant de comparer plusieurs méthodes d'alignements, il nous semble intéressant de voir leur pertinence globale. Pour cela, nous regardons le pourcentage de contribution des différentes approches dans notre alignement final. Cela correspond au nombre de fois où l'on estime une pose, d'une méthode donnée, comme étant meilleure que les autres.

Cette expérimentation a été effectuée sur la base de *SHREC'09 Generic* avec des vues de $256 * 256$ pixels. Dans celle-ci, notre estimateur choisi la meilleure pose parmi : l'ACP Continue, l'ACP par normales et l'approche de Chaouch [24][27], basé sur les propriétés de symétrie, présentée dans la section 3.2.2 du chapitre 3. Afin de bien comprendre le comportement de chaque méthode d'alignement, nous présentons dans la figure 7.7 les résultats globaux et dans la figure 7.6 le détail par classe.

Avant de rentrer dans une étude plus complète des résultats, nous pouvons voir sur la figure 7.7 que la méthode basée sur les propriétés de symétrie est plus pertinente que les deux autres. Cela est cohérent avec les expérimentations de l'auteur effectuées dans

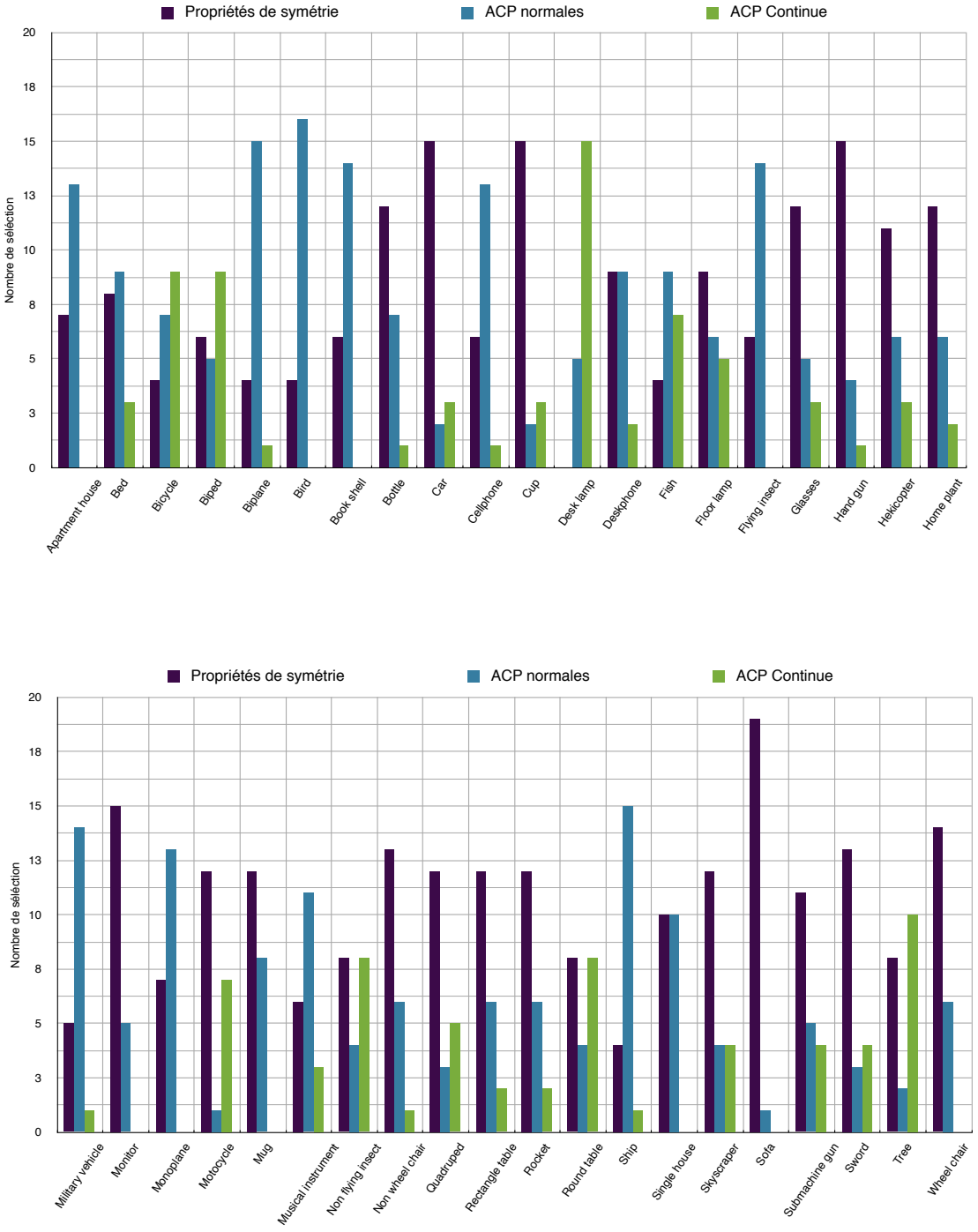


FIGURE 7.6 – *Cette figure représente la contribution par classe de trois méthodes d'alignement (ACP Continue, ACP Normales, Symétrie) dans notre processus d'estimation de la pose. Cette évaluation est effectuée sur la base de SHREC'09 Generic avec des silhouettes de* $256 * 256$ *pixels. .*

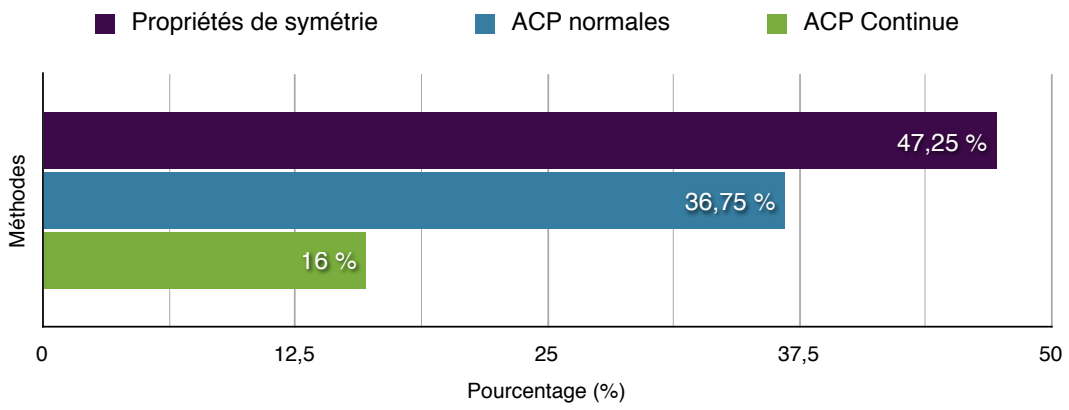


FIGURE 7.7 – *Cette figure représente la contribution de trois méthodes d'alignement (ACP Continue, ACP Normales, Symétrie) dans notre processus d'estimation de la pose. Cette évaluation est effectuée sur la base de SHREC'09 Generic avec des silhouettes de* $256 * 256$ *pixels. Il est clairement visible que l'approche de Chaouch [24][27], utilisant les propriétés de symétrie, aligne mieux les objets que les deux autres.*

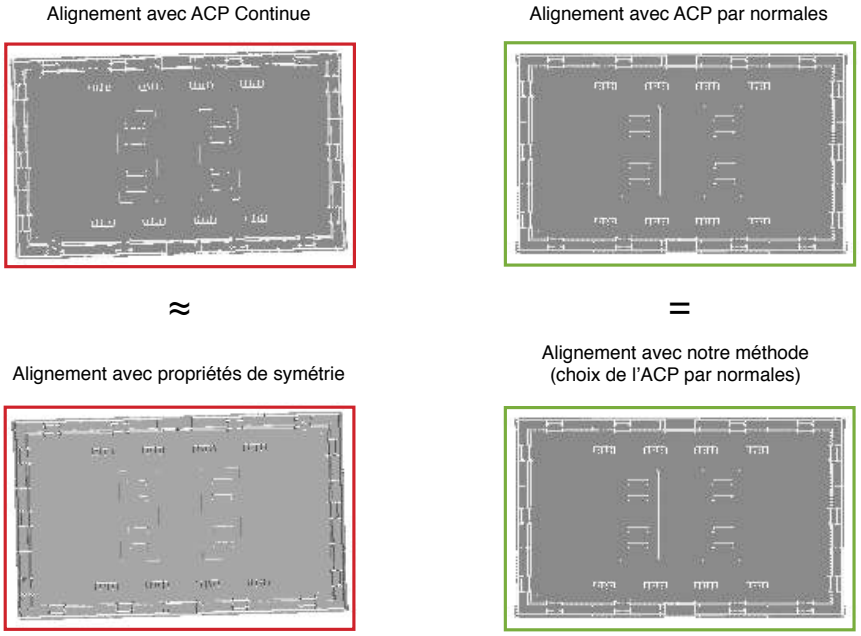


FIGURE 7.8 – *Illustration des résultats de quatre alignements sur un objet 3D de la classe "Apartment house" de la base SHREC'09 Generic. Il est clairement visible que la méthode par propriétés de symétrie est basé sur l'ACP Continue tandis que la notre estime que l'ACP par normales est plus pertinentes. Finalement, notons que les résultats des deux ACP sont tout de même très proches.*

[24][27]. Nous pouvons également nous rendre compte que sur cette base, l'ACP par normales offre un alignement plus pertinent que l'ACP Continue. Ce constat est opposé à celui effectué sur la base *SHREC'07 Watertight* (voir figure 7.5), ceci certainement à cause du caractère "générique" de la base de données, plus adapté à l'approche par normales.

La figure 7.6 montre que sur certaines classes, telles que "Apartment house", "Biplane", "Book shell" ou encore "Ship", c'est l'ACP par normales qui est la plus pertinente pour notre approche (voir figure 7.8). En revanche, sur la base complète c'est la méthode par propriétés de symétrie qui l'emporte. Ce constat montre bien la force de notre approche à choisir la pose la plus adaptée. Il faut cependant noter que les résultats proposés dans la figure 7.7 ne reflètent pas réellement la qualité d'un alignement. En effet, comme nous pouvons le voir dans la figure 7.8, il existe certains cas où les différences entre les poses sont minimes influençant, de ce fait, très peu la pertinence des descripteurs de forme capturés.

La figure 7.9 montre une comparaison de l'ACP Continue, de l'ACP par normales et de notre approche, pour quatre objets 3D différents. Nous pouvons voir que dans chaque cas notre estimateur permet de sélectionner la pose correcte. Dans les deux premiers cas (colonne 1 et 2) il s'agit de l'ACP Continue tandis que pour les deux derniers (colonne 3 et 4) notre estimateur choisit l'ACP par normales.

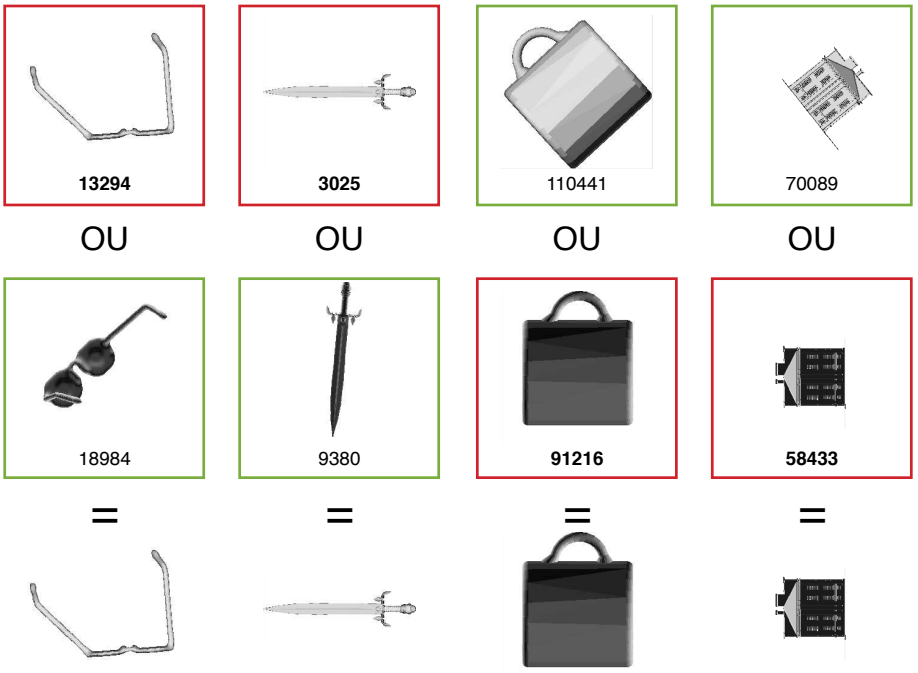


FIGURE 7.9 – *Cette figure présente un comparaison d'alignements utilisant : L'ACP Continue, l'ACP par normales et notre approche. La somme des aires des silhouettes indispensable au calcul de notre estimation est indiquée en dessous de chaque objet.*

Pour conclure, nous avons vu que notre estimateur permet d'améliorer les méthodes classiques d'alignement d'objets 3D en tirant parti des forces de chacune. Ainsi, dans la majorité des cas, nous sommes capables de sélectionner la pose la plus adaptée à un modèle 3D donné, et ceci, quelle que soit sa classe. Finalement, grâce à cette nouvelle méthode, il est possible d'aligner un objet en vue d'extraire ses caractéristiques de forme, qui elles ne sont pas invariantes à la pose.

Expérimentations sur l'alignement

Afin d'évaluer le gain de performances qu'amène notre approche d'alignement, nous avons comparé les résultats obtenus avec différentes méthodes permettant de définir la pose d'un objet 3D. En particulier, nous avons comparé l'ACP Continue (ACPC) et l'ACP par normales (ACPN) avec les performances obtenues avec l'estimateur que nous avons proposé. La figure 7.10 et les tableaux 7.1 présentent ces expérimentations pour les bases *SHREC'07 Watertight* et *SHREC'09 Generic* avec les descripteurs par ensemble de pixels (EP) et par histogrammes de convexités/concavités (MCC-H).

(a) *Descripteur EP - SHREC'07 Watertight*

	NN	FT	ST	DCG
ACPC	76.5	38.87	52.32	71.5
ACPN	75.75	37.5	50.63	70.35
Nous	**78**	**41.76**	**56.2**	**73.76**

(b) *Descripteur MCC-H - SHREC'07 Watertight*

	NN	FT	ST	DCG
ACPC	88.75	56.21	70.5	82.43
ACPN	88	53.76	67.6	81.01
Nous	**91**	**60.22**	**74.54**	**85.23**

(c) *Descripteur EP - SHREC'09 Generic*

	NN	FT	ST	DCG
ACPC	71.37	32.22	43.51	65.75
ACPN	72.12	31.28	42.13	64.82
Nous	**73.75**	**34.72**	**46.5**	**68.41**

(d) *Descripteur MCC-H - SHREC'09 Generic*

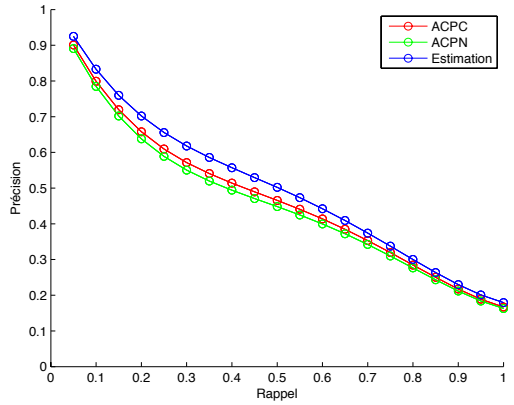
	NN	FT	ST	DCG
ACPC	81.62	45.08	58.13	75.23
ACPN	80.37	43.16	56.16	73.42
Nous	**84.12**	**50.28**	**65.01**	**79.76**

TABLE 7.1 – *Mesures de performances en % pour deux descripteurs en fonction de trois alignements sur les bases SHREC'07 Watertight et SHREC'09 Generic.*

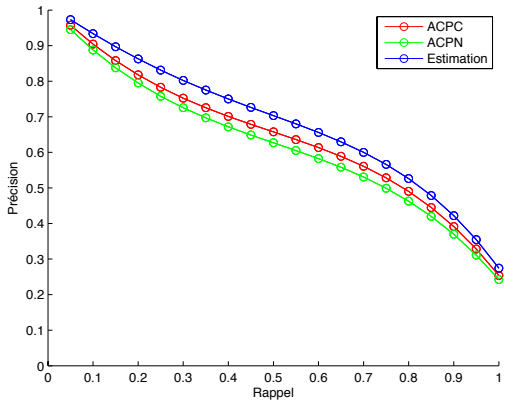
Nous pouvons voir sur ces résultats que dans tous les cas notre approche d'alignement permet d'améliorer la pertinence des résultats. Remarquons que le gain est comparable pour les deux descripteurs utilisés ainsi que pour les deux bases. Étant donné ces résultats, nous utiliserons dans toutes les expérimentations de ce chapitre, notre méthode d'alignement plutôt que l'ACP Continue ou l'ACP par normales.

Expérimentations sur l'ordonnancement des axes et sur les miroirs

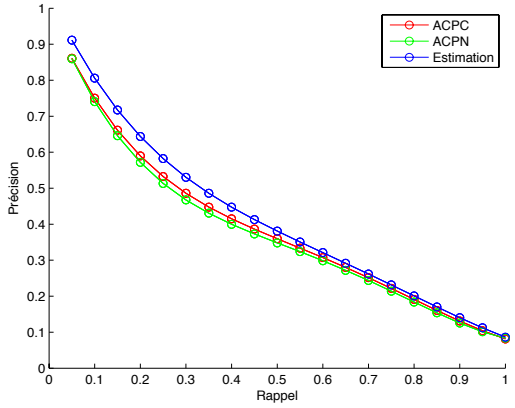
La méthode d'alignement d'objets 3D que nous avons proposée dans le chapitre 4 n'est pas robuste par rapport à l'ordre des axes ainsi qu'à leur signe (voir section 4.3.4). Afin de prendre en compte cette lacune, nous calculons pour chaque requête les 48 permutations possibles des axes du repère 3D. Afin d'évaluer l'intérêt de ce processus coûteux,



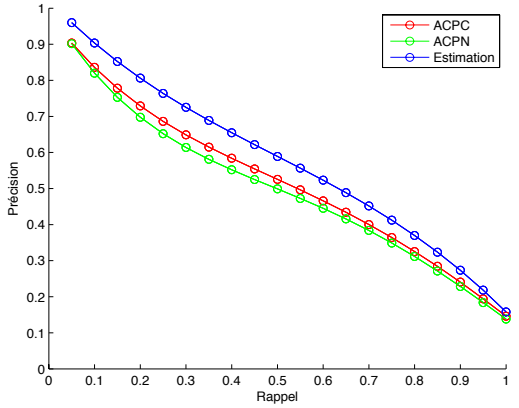
(a) *Descripteur EP - SHREC'07 Watertight*



(b) *Descripteur MCC-H - SHREC'07 Watertight*



(c) *Descripteur EP - SHREC'09 Generic*

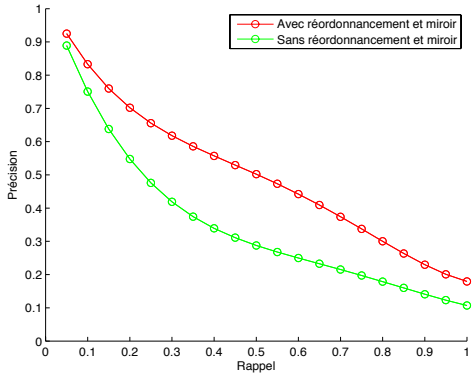


(d) *Descripteur MCC-H - SHREC'09 Generic*

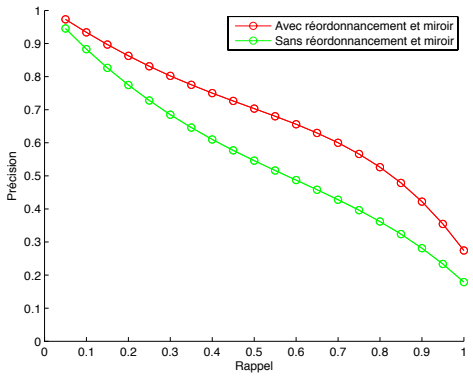
FIGURE 7.10 – *Courbes rappel/précision pour deux descripteurs en fonction de trois alignements sur les bases SHREC'07 Watertight et SHREC'09 Generic.*

nous proposons dans la figure 7.11 et les tableaux 7.2 un ensemble d'expérimentations. En particulier, nous évaluons le gain de performances apporté par la prise en compte de l'ordonnancement des axes et les miroirs. Ces tests ont été effectués avec les descripteurs EP et MMC-H sur les bases *SHREC'07 Watertight* et *SHREC'09 Generic*.

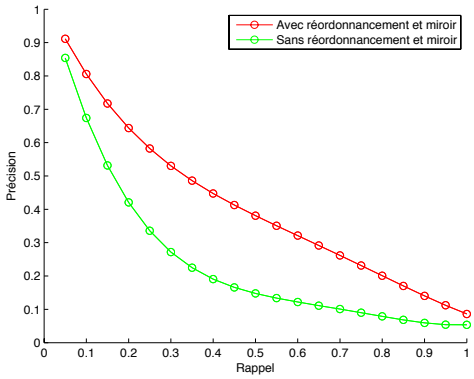
Les résultats que nous obtenons sont sans ambiguïtés et montrent qu'il est nécessaire de traiter le problème de l'ordonnancement des axes et de leur signe. Il est intéressant de noter que le gain de performances est nettement plus important pour le descripteur par ensembles de pixels. Ceci est dû à l'invariance en rotation et miroir, sur les projections 2D, de la description par histogrammes. Cependant, il est intéressant de noter que malgré cette invariance, une forte baisse des performances est enregistrée, ceci à cause de l'impossibilité d'aligner chaque projection indépendamment (voir remarque 6.2 du chapitre 6). Finalement, afin de garantir une recherche efficace, nous associons, dans la suite des expérimentations, notre méthode d'alignement au traitement des 48 permutations.



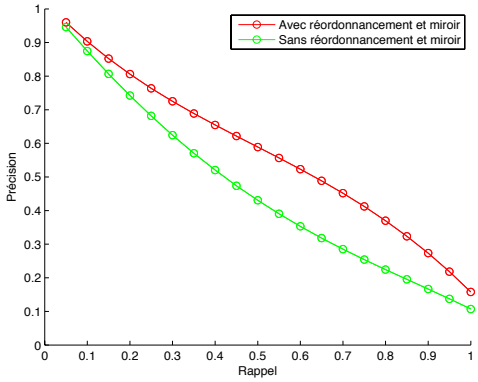
(a) *Descripteur EP - SHREC'07 Watertight*



(b) *Descripteur MCC-H - SHREC'07 Watertight*



(c) *Descripteur EP - SHREC'09 Generic*



(d) *Descripteur MCC-H - SHREC'09 Generic*

FIGURE 7.11 – *Courbes rappel/précision pour deux descripteurs en fonction du réordonnancement des axes et des miroirs de trois alignements sur les bases SHREC'07 Watertight et SHREC'09 Generic.*

(a) *Descripteur EP - SHREC'07 Watertight*

	NN	FT	ST	DCG
Sans	66.75	29.24	38.83	63.34
Avec	**78**	**41.76**	**56.2**	**73.76**

(b) *Descripteur MCC-H - SHREC'07 Watertight*

	NN	FT	ST	DCG
Sans	81.5	47	60.54	77.67
Avec	**91**	**60.22**	**74.54**	**85.23**

(c) *Descripteur EP - SHREC'09 Generic*

	NN	FT	ST	DCG
Sans	58	19.57	28.44	54.8
Avec	**73.75**	**34.72**	**46.5**	**68.41**

(d) *Descripteur MCC-H - SHREC'09 Generic*

	NN	FT	ST	DCG
Sans	80	39.58	53.41	73.48
Avec	**84.12**	**50.28**	**65.01**	**79.76**

TABLE 7.2 – *Mesures de performances en % pour deux descripteurs en fonction du réordonnancement des axes et des miroirs de trois alignements sur les bases SHREC'07 Watertight et SHREC'09 Generic.*

7.4 Expérimentation sur l'indexation et la recherche d'objets 3D

Nous avons proposé dans le chapitre 5 trois nouveaux descripteurs permettant de caractériser la forme d'un objet 3D. Leur calcul est généralement basé sur des paramètres permettant, le plus souvent, de jouer sur les aspects compacité et pertinence de la signature. Dans ce cadre, nous introduisons dans les sections 7.4.1 à 7.4.5 les résultats d'expérimentations visant à comprendre l'influence de ces paramètres sur les résultats. Celles-ci sont effectuées en accord avec le processus de recherche présenté dans le chapitre 6, avec comme requête des objets 3D.

D'autre part, nous avons développé, dans ce même chapitre 6, des optimisations permettant de réduire le temps de calcul et d'améliorer la pertinence des résultats retournés à l'utilisateur. Afin d'évaluer le gain de performance de nos deux approches, en fonction de leurs paramètres, nous commentons dans les sections 7.4.6 et 7.4.7 un ensemble d'expérimentations menées sur ces optimisations, avant de comparer nos descripteurs entre eux dans la section 7.4.8.

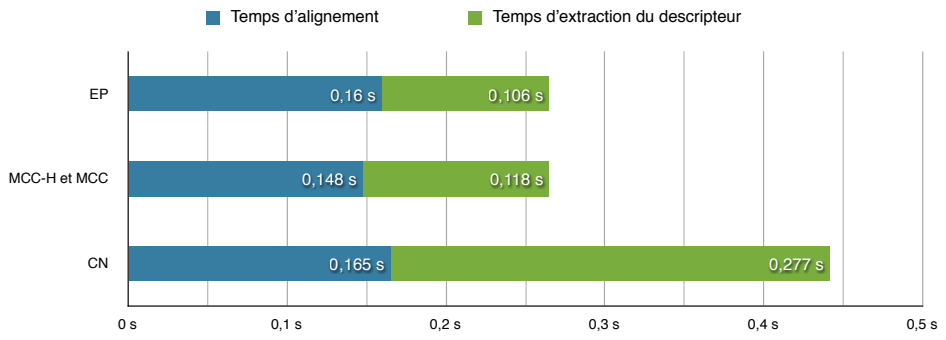
Finalement, nous terminons par la section 7.4.9 qui montre les performances de notre processus d'indexation et de recherche lorsque la requête de l'utilisateur est une photo ou un dessin. Ces expérimentations sont illustrées par des résultats de recherche obtenus avec le prototype mobile présenté dans le chapitre 6.

Notons que l'évaluation des performances est exprimée avec les mesures : *Nearest neighbor*, *first* et *second tier* ainsi que *Discounted cumulative gain* présentées dans l'annexe D, mais aussi avec des courbes rappel/précision introduites dans la même annexe. Les expérimentations sont effectuées en interrogeant une base de données avec, comme requête, chacun des objets 3D de cette même base. Lorsque ce n'est pas précisé, les expérimentations utilisent des projections de $256 * 256$ pixels, l'octaèdre plus trois bissectrices, comme jeu de points de vues, ainsi que notre méthode d'alignement.

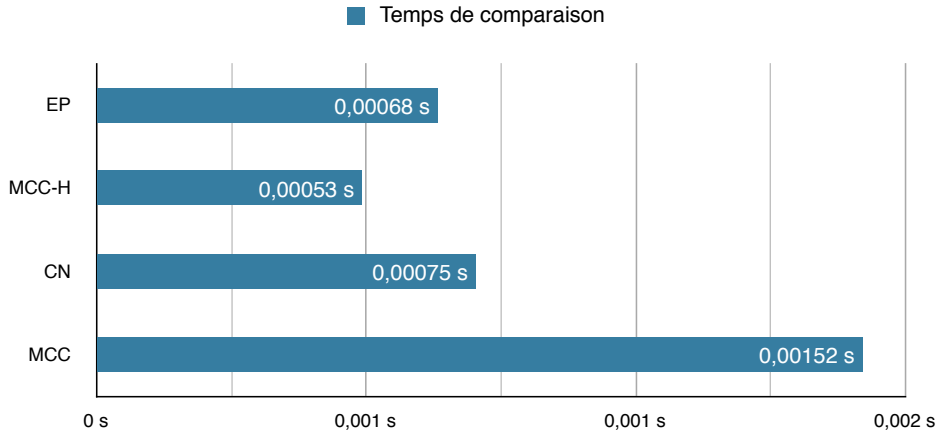
7.4.1 Temps d'extraction et de comparaison

Afin d'évaluer dans la pratique la complexité de nos différents descripteurs, nous proposons dans la figure 7.12 des mesures du temps nécessaire pour effectuer chaque étape de notre processus d'indexation et de recherche. Les valeurs correspondent au temps, en secondes, nécessaire à l'alignement d'un objet, à l'extraction de la signature et à la comparaison de deux descripteurs du même type (nous prenons en compte les 48 permutations). Le système sur lequel ces mesures ont été effectuées est un *MacBook* possédant un processeur *Intel Core 2 Duo*, *4 Go* de mémoire vive et une carte graphique *NVIDIA GeForce 9400M* ayant *256 Mo* de mémoire vive.

La figure 7.12 montre que le temps d'extraction du descripteur CN est un peu plus de deux fois celui des autres approches. Ceci vient du fait que ce descripteur nécessite deux fois plus de silhouettes (voir section 5.2.2 du chapitre 5) pour caractériser la forme d'un



(a) *Temps d'alignement et d'extraction des descripteurs*



(b) *Temps de comparaison de deux signatures*

FIGURE 7.12 – *Mesures du temps de calcul, en secondes, nécessaire l'alignement d'un objet, à l'extraction de la signature et à la comparaison de deux descripteurs du même types (en prenant en compte les 48 permutations).*

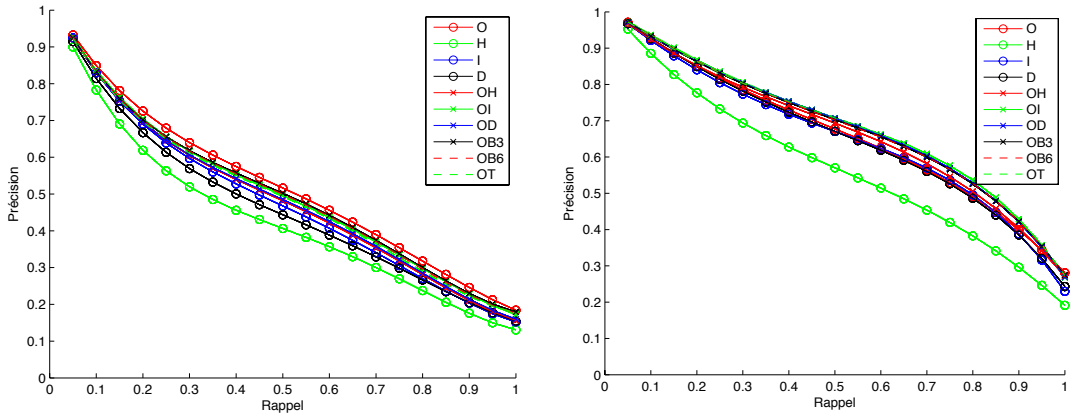
objet 3D. Nous pouvons aussi noter que notre processus d'alignement nécessite environ 0.16 seconde pour déterminer la pose d'un objet. Ce temps élevé est dû au coût élevé du calcul de l'aire des silhouettes qui pourrait être accéléré en transférant ce calcul sur le processeur graphique (GPU). En ce qui concerne le temps de comparaison, nos pouvons noter que l'utilisation de la "déformation temporelle dynamique" dans MCC ralentit fortement la recherche.

7.4.2 Choix du nombre de vues

Nous avons introduit, dans la section 5.2 du chapitre 5, dix jeux de points de vues à partir desquels nous pouvons extraire des projections 2D. Deux contraintes ont guidé leur choix. La première est le fait que ces points de vues soit uniformément réparties sur la sphère unité. Nous l'avons résolu en utilisant les polyèdres de Platon. La seconde, quant à elle, vise à tirer parti de l'alignement du chapitre 4 en plaçant des points de vues sur les

axes du repère de l'objet 3D aligné. Pour cela, nous combinons l'octaèdre avec les solides de Platon ou des bissectrices/trisectrices.

Afin d'évaluer les performances qu'offre chacun des points de vues, nous dressons dans les tableaux 7.3 et la figure 7.13 les résultats obtenus, pour une recherche exhaustive des objets de *SHREC'07 Watertight*, avec le descripteur par ensemble de pixels ainsi que pour celui basé sur les histogrammes de convexités/concavités.



(a) *Descripteur EP - SHREC'07 Watertight* (b) *Descripteur MCC-H - SHREC'07 Watertight*

FIGURE 7.13 – *Courbes rappel/précision pour deux descripteurs en fonction des dix jeux de points de vues définis dans la section 5.2 du chapitre 5. (a) et (b) représentent respectivement les descripteurs EP et MCC-H sur la base SHREC'07 Watertight.*

Les résultats que nous obtenons montrent clairement que le placement des points de vues doit être en corrélation avec les axes du repère 3D. En effet, si nous comparons les lignes 1 (O), 2 (H) et 5 (OH) des tableaux 7.3, nous nous rendons compte que c'est la position des points de vues, plus que leur nombre qui fait varier les performances. Cependant, nous pouvons aussi remarquer sur les lignes 8 (OB3), 9 (OB6) et 10 (OT) que l'ajout de points de vues supplémentaires à l'octaèdre permet d'améliorer sensiblement les résultats pour le descripteur MCC-H. En revanche, il engendre une baisse des performances avec EP.

Ces expérimentations montrent que le jeu de points de vues basé sur l'octaèdre (O) ainsi que celui lui ajoutant trois bissectrices (OB3) offrent de bonnes performances pour un nombre de projections limité. Dans ce cadre, nous proposons dans les tableaux 7.4 les performances sur les trois bases de données en utilisant ces deux jeux de points de vues et les descripteurs EP et MCC-H.

En accord avec ces résultats, nous utilisons l'octaèdre, ou sa version améliorée avec trois bissectrices, dans la suite de ce chapitre. En effet, ils garantissent des résultats robustes tout en limitant le nombre de points de vues et donc la taille de la signature à stocker. Finalement, nous conseillons de les utiliser pour toute approche "2.5D/3D" et "2D/3D".

(a) *Descripteur EP - SHREC'07 Watertight*

	NN	FT	ST	DCG
Octaèdre (O)	**79.5**	**43.62**	**57.51**	**75**
Hexaèdre (H)	71.25	34.85	48.37	68.38
Icosaèdre (I)	79.25	39.21	53.49	72.44
Dodécaèdre (D)	76.25	39.21	53.49	72.44
Octaèdre + Hexaèdre (OH)	77.25	40.54	53.93	72.83
Octaèdre + Icosaèdre (OI)	79.25	41.58	55.66	73.67
Octaèdre + Dodécaèdre (OD)	78.5	40.74	54.35	72.99
Octaèdre + 3 Bissectrice (OB3)	**78**	**41.76**	**56.2**	**73.76**
Octaèdre + 6 Bissectrice (OB6)	78.75	41.38	56.07	73.64
Octaèdre + 3 Trissectrice (OT)	78	41.12	55.57	73.34

(b) *Descripteur MCC-H - SHREC'07 Watertight*

	NN	FT	ST	DCG
Octaèdre (O)	**89.25**	**57.92**	**72.89**	**84.01**
Hexaèdre (H)	84	48.45	63.95	78.52
Icosaèdre (I)	89	57.1	71.54	83.56
Dodécaèdre (D)	88.5	56.79	71.63	83.7
Octaèdre + Hexaèdre (OH)	90	58.26	73.17	84.29
Octaèdre + Icosaèdre (OI)	91.25	60.38	74.51	85.46
Octaèdre + Dodécaèdre (OD)	91	59.74	74.54	85.21
Octaèdre + 3 Bissectrice (OB3)	**91**	**60.22**	**74.54**	**85.23**
Octaèdre + 6 Bissectrice (OB6)	91.75	59.88	74.03	85.17
Octaèdre + 3 Trissectrice (OT)	91.25	60.18	74.5	85.32

TABLE 7.3 – *Mesures de performances en % pour deux descripteurs en fonction des dix jeux de points de vues définis dans la section 5.2 du chapitre 5. (a) et (b) représentent respectivement les descripteurs EP et MCC-H sur la base SHREC'07 Watertight.*

(a) *Descripteur EP - SHREC'09 Generic*

	NN	FT	ST	DCG
(O)	71.37	34.45	46.24	67.57
(OB3)	**73.75**	**34.72**	**46.5**	**68.41**

(b) *Descripteur MCC-H - SHREC'09 Generic*

	NN	FT	ST	DCG
(O)	78.5	46.4	61.89	76.28
(OB3)	**84.12**	**50.28**	**65.01**	**79.76**

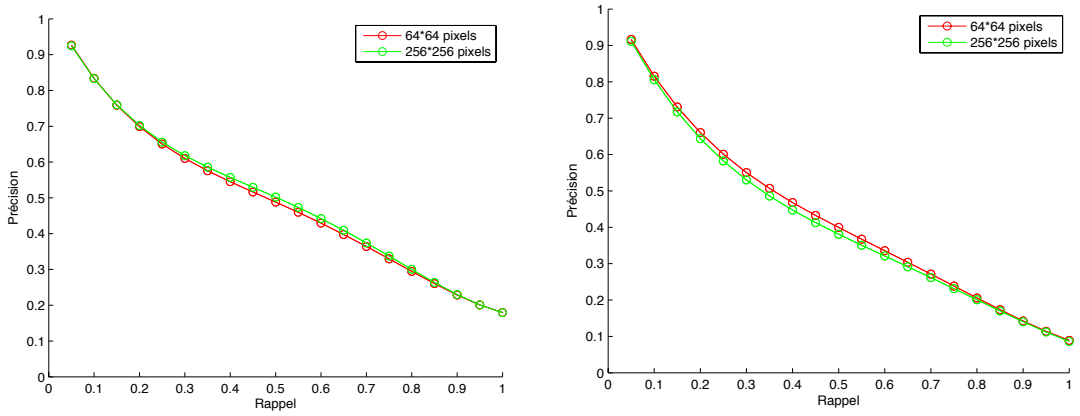
TABLE 7.4 – *Mesures de performances en % pour les descripteurs EP et MCC-H en fonction de deux jeux de points de vues (voir tableau 7.3). Chaque ligne de tableaux représente une même base de données.*

7.4.3 Descripteur par ensemble de pixel (EP)

Comme nous l'avons vu dans la section 5.3 du chapitre 5, la méthode de description de la forme basée sur les ensembles de pixels (EP) s'appuie sur différents paramètres. Il s'agit en particulier du nombre de silhouettes et de leur résolution. Le nombre de silhouettes ayant déjà été discuté, nous proposons des expérimentations qui visent à évaluer, grâce à des mesures précises (voir annexe D), l'influence de la résolution sur deux bases de données différentes.

Expérimentations sur la résolution des silhouettes

La figure 7.14 et les tableaux 7.5 illustrent l'évolution des performances du descripteur par ensemble de pixels en fonction de la résolution des projections. Les expérimentations ont été menées sur les bases *SHREC'07 Watertight* et *SHREC'09 Generic* avec des silhouettes ayant une résolution de $64 * 64$ ou de $256 * 256$ pixels.



(a) *Descripteur EP - SHREC'07 Watertight* (b) *Descripteur EP - SHREC'09 Generic*

FIGURE 7.14 – *Courbes rappel/précision pour le descripteur EP en fonction de la résolution des silhouettes (en pixels).*

(a) *Descripteur EP - SHREC'07 Watertight*

	NN	FT	ST	DCG
64*64	**80**	40.92	55.41	73.54
256*256	78	**41.76**	**56.2**	**73.76**

(b) *Descripteur EP - SHREC'09 Generic*

	NN	FT	ST	DCG
64*64	**75.12**	**36.14**	**47.97**	**69.41**
256*256	73.75	34.72	46.5	68.41

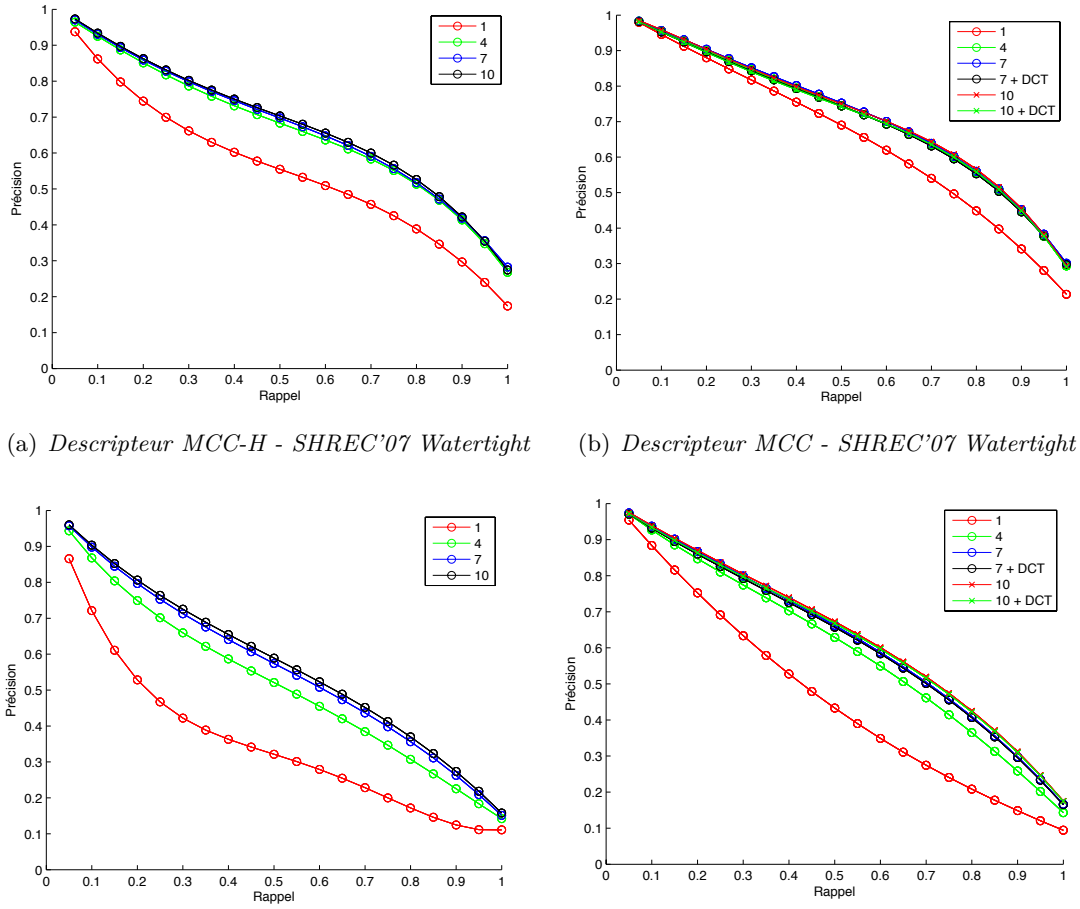
TABLE 7.5 – *Mesures de performances en % pour le descripteur EP en fonction de la résolution des silhouettes (en pixels).*

Ces résultats montrent que la résolution n'a qu'une très faible influence sur les performances de la méthode par ensemble de pixels. Il est intéressant de noter qu'une résolution moins élevée n'est pas forcément synonyme de baisse des performances. Pour conclure, il

est préférable, lors de l'utilisation de ce descripteur d'utiliser une résolution de $64 * 64$ pixels car elle n'influence pas réellement les résultats et permet de réduire la taille de la signature à stocker.

7.4.4 Descripteur par convexités/concavités (MCC-H et MCC)

Nous avons introduit dans le chapitre 5 deux descripteurs basés sur les informations multi-échelles de convexités et de concavités calculées sur les contours des silhouettes. Le premier, utilise un ensemble d'histogrammes (MCC-H) tandis que le second, se base sur une compression des informations à l'aide d'un DCT (MCC). Afin de bien comprendre l'influence des différents paramètres de nos approches sur les performances, nous proposons un ensemble d'expérimentations réalisées sur les bases *SHREC'07 Watertight* et *SHREC'09 Generic*. Lorsque nous ne le précisions pas, les évaluations utilisent 10 échelles ainsi que la distance $L1$. Pour le descripteur MCC, nous utilisons en plus la "déformation temporelle dynamique" associée à un une bande diagonale de 8 pixels.



(a) *Descripteur MCC-H - SHREC'07 Watertight*

(b) *Descripteur MCC - SHREC'07 Watertight*

(c) *Descripteur MCC-H - SHREC'09 Generic*

(d) *Descripteur MCC - SHREC'09 Generic*

FIGURE 7.15 – *Courbes rappel/précision pour les descripteurs MCC-H et MCC en fonction du nombre d'échelles.*

Expérimentations sur le nombre d'échelles

Pour ces deux descripteurs, nous calculons les informations de convexités/concavités sur plusieurs échelles afin de garantir une bonne caractérisation de la forme des contours. Afin d'évaluer l'évolution de la pertinence en fonction du nombre d'échelles, nous présentons dans la figure 7.15 ainsi que dans les tableaux 7.6 un ensemble d'expérimentations. Celles-ci ont été effectuées avec les deux descripteurs proposés (MCC-H et MCC).

(a) *Descripteur MCC-H - SHREC'07 Watertight*

	NN	FT	ST	DCG
1	76	47.41	65.41	77
4	88.25	59.2	73.64	84
7	90.5	59.79	74.07	84.79
10	**91**	**60.22**	**74.54**	**85.23**

(b) *Descripteur MCC - SHREC'07 Watertight*

	NN	FT	ST	DCG
1	94	58.37	68.46	84.87
4	94.5	64.29	75.5	87.34
7	94	64.72	75.99	87.62
7+DCT	94	64.03	75.28	87.15
10	94.75	64.08	75.2	87.2
10+DCT	**95**	**64.8**	**75.96**	**87.71**

(c) *Descripteur MCC-H - SHREC'09 Generic*

	NN	FT	ST	DCG
1	57	30.41	43.64	62.76
4	80.62	45.37	59.83	75.96
7	84.5	49.26	63.92	79
10	**84.12**	**50.28**	**65.01**	**79.76**

(d) *Descripteur MCC - SHREC'09 Generic*

	NN	FT	ST	DCG
1	84.75	40.74	47.3	73.49
4	89.25	53.76	62.57	81.6
7	89.75	56.68	66.32	83.24
7+DCT	89.63	56.28	65.8	82.85
10	90.37	57.05	67.09	83.33
10+DCT	**90.5**	**57.46**	**67.6**	**83.71**

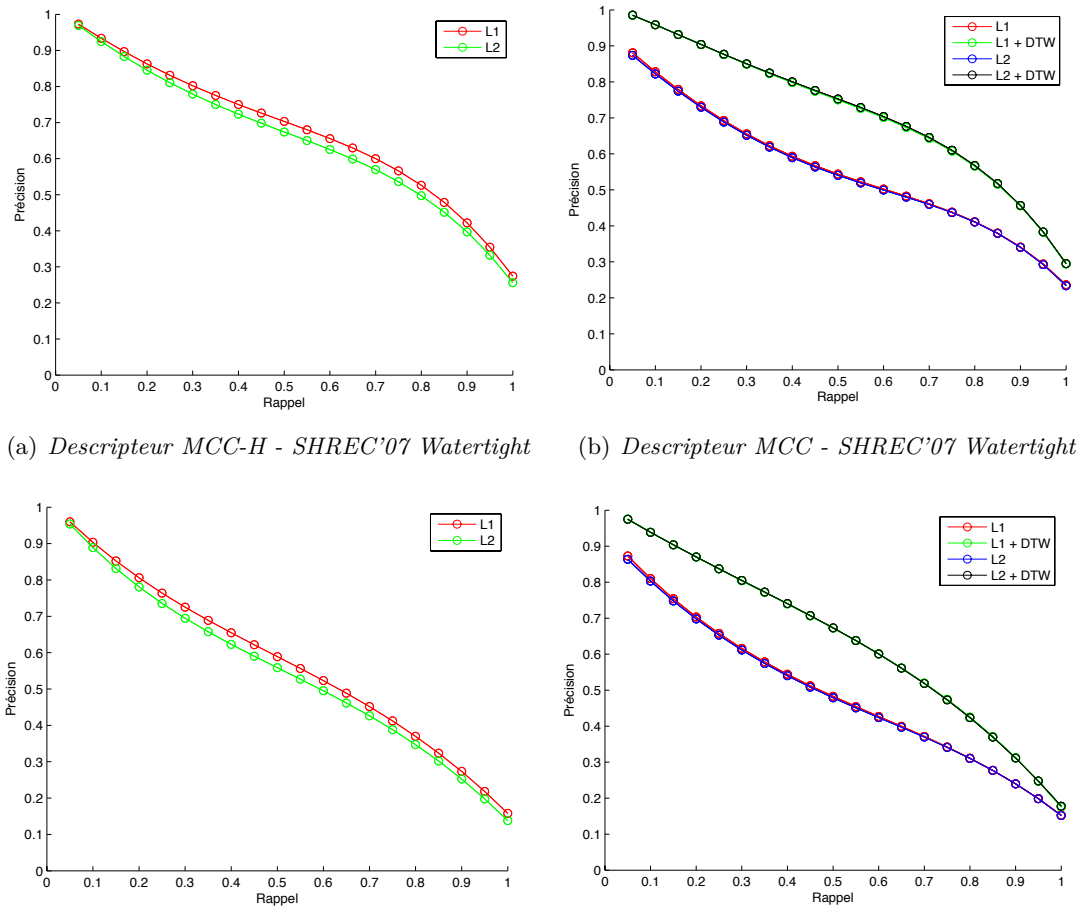
TABLE 7.6 – *Mesures de performances en % pour les descripteurs MCC-H et MCC en fonction du nombre d'échelles.*

Les résultats obtenus montrent que pour les deux descripteurs et quelle que soit la base de données utilisée, les performances augmentent avec le nombre d'échelles. Il est intéressant de noter que l'emploi d'une seule échelle fait baisser considérablement la pertinence globale des résultats. Étant donné que la compression des données à l'aide de la 'DCT" n'influence que peu la pertinence, mais permet de réduire l'espace de stockage, il est conseillé de l'appliquer. Pour conclure, ces expérimentations montrent qu'il est nécessaire d'utiliser l'approche multi-échelles si l'on souhaite garantir de bons résultats de recherche. Afin de décrire efficacement la forme d'un objet 3D, nous utilisons donc 10 échelles. Cependant, si l'on souhaite obtenir une signature plus synthétique, mais tout de même une pertinence raisonnable, il est possible de n'en utiliser que 7.

Expérimentations sur la mesure de dissimilarité

Afin de comparer les informations de convexités/concavités stockées dans nos deux descripteurs, nous avons mis en place des mesures de dissimilarité. Pour la signature ba-

sée sur les histogrammes, nous évaluons les performances des distances L1 et L2 dans la figure 7.16 et les tableaux 7.7. En ce qui concerne le descripteur utilisant la “DCT”, les mêmes figures et tableaux montrent l’intérêt de la “déformation temporelle dynamique” pour mettre en correspondance deux jeux d’informations de convexités/concavités.



(a) *Descripteur MCC-H - SHREC’07 Watertight*

(b) *Descripteur MCC - SHREC’07 Watertight*

(c) *Descripteur MCC-H - SHREC’09 Generic*

(d) *Descripteur MCC - SHREC’09 Generic*

FIGURE 7.16 – *Courbes rappel/précision pour les descripteurs MCC-H et MCC en fonction de la mesure de dissimilarité.*

Ces expérimentations montrent que l’utilisation de la distance L1 ou L2 n’influence que très peu les résultats, et ceci, quelle que soit la base de données. En revanche, la “déformation temporelle dynamique” offre un gain de performances beaucoup plus important. C’est donc raisonnablement vers cette dernière que nous nous sommes tournés pour comparer deux informations de convexités/concavités classiques, tandis que nous utilisons la distance L1 pour les histogrammes.

(a) *Descripteur MCC-H - SHREC'07 Watertight*

	NN	FT	ST	DCG
L1	**91**	**60.22**	**74.54**	**85.23**
L2	90.5	57.91	72.13	83.79

(b) *Descripteur MCC - SHREC'07 Watertight*

	NN	FT	ST	DCG
L1	76	46.72	71.26	77.91
L1+DTW	**95**	**64.8**	**75.96**	**87.71**
L2	74.75	46.34	71.09	77.72
L2+DTW	95.25	65.01	76.08	87.77

(c) *Descripteur MCC-H - SHREC'09 Generic*

	NN	FT	ST	DCG
L1	**84.1**	**50.28**	**65.01**	**79.76**
L2	81.6	47.75	63.06	78.18

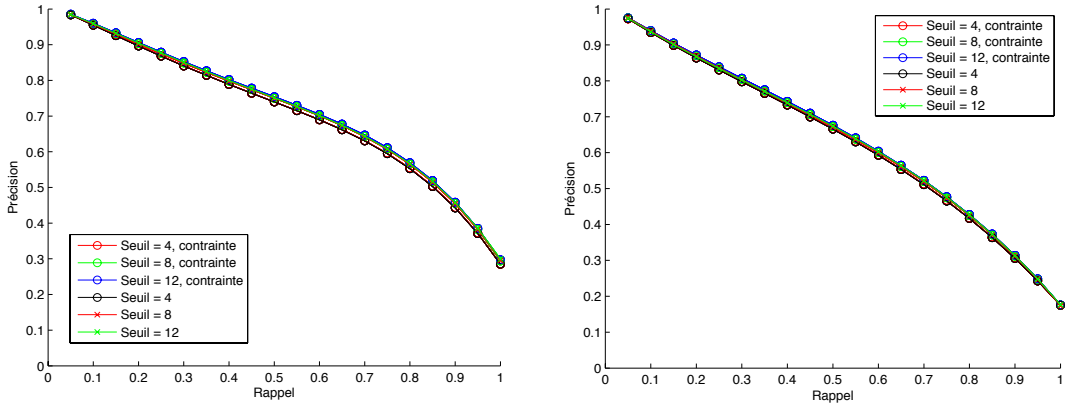
(d) *Descripteur MCC - SHREC'09 Generic*

	NN	FT	ST	DCG
L1	72	42.65	64.2	74.55
L1+DTW	**90.5**	**57.46**	**67.6**	**83.71**
L2	72.5	42.53	64.24	74.61
L2+DTW	90.12	57.5	67.48	83.71

TABLE 7.7 – *Mesures de performances en % pour les descripteurs MCC-H et MCC en fonction de la mesure de dissimilarité.*

Expérimentations sur la "déformation temporelle dynamique"

Le descripteur de convexités et de concavités des contours qui n'est pas basé sur les histogrammes utilise une "déformation temporelle dynamique" pour mettre en correspondance les informations de deux silhouettes. La rapidité du calcul de celle-ci est en partie contrôlée par un seuil permettant de contraindre le chemin de coût minimal autour de la diagonale. D'autre part, une contrainte a été mise en place pour interdire un point du contour A d'être mis en correspondance avec plus de deux points du contour B. Afin d'évaluer l'évolution des performances de ce descripteur en fonction de ces deux paramètres, nous proposons dans la figure 7.17 et les tableaux 7.8 un ensemble d'expérimentations.



(a) *Descripteur MCC - SHREC'07 Watertight* (b) *Descripteur MCC - SHREC'09 Generic*

FIGURE 7.17 – *Courbes rappel/précision pour le descripteur MCC en fonction des paramètres de la "déformation temporelle dynamique".*

(a) *Descripteur MCC - SHREC'07 Watertight*

	NN	FT	ST	DCG
bande diagonale de 4 points avec contrainte	95.25	63.67	75.66	87.17
bande diagonale de 8 points avec contrainte	**95**	**64.8**	**75.96**	**87.71**
bande diagonale de 12 points avec contrainte	94.75	65.25	76.18	87.9
bande diagonale de 4 points sans contrainte	95	63.74	75.63	87.13
bande diagonale de 8 points sans contrainte	95.25	64.91	76.1	87.59
bande diagonale de 12 points sans contrainte	94.75	64.91	76.22	87.8

(b) *Descripteur MCC - SHREC'09 Generic*

	NN	FT	ST	DCG
bande diagonale de 4 points avec contrainte	89.5	56.74	67.48	83.33
bande diagonale de 8 points avec contrainte	**90.5**	**57.46**	**67.6**	**83.71**
bande diagonale de 12 points avec contrainte	90	57.76	67.66	83.84
bande diagonale de 4 points sans contrainte	89.62	56.64	67.52	83.29
bande diagonale de 8 points sans contrainte	90.5	57.35	67.55	83.68
bande diagonale de 12 points sans contrainte	90.62	57.58	67.49	83.63

TABLE 7.8 – *Mesures de performances en % pour le descripteur MCC en fonction des paramètres de la "déformation temporelle dynamique".*

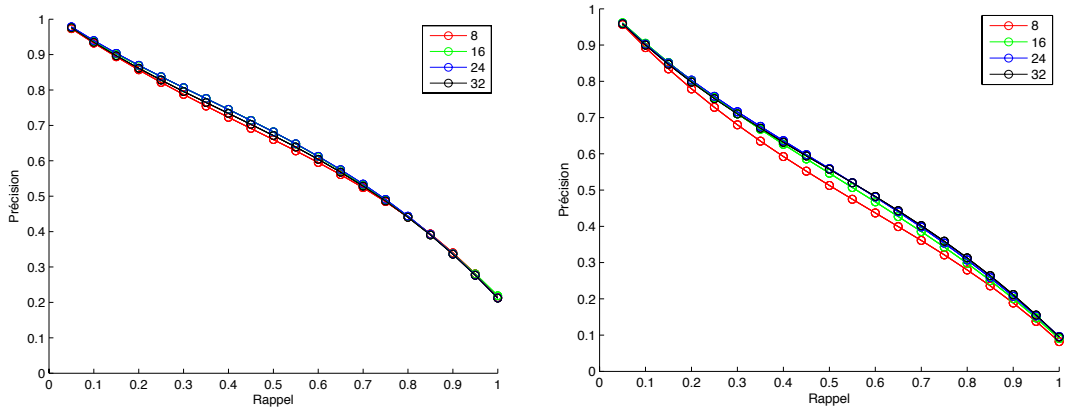
Les résultats montrent que l'élargissement de la bande diagonale, dans laquelle doit rester le chemin de coût minimal, permet d'augmenter la pertinence des résultats. Cependant, il faut noter que plus sa taille augmente, plus le temps de calcul devient important. D'autre part, nous pouvons voir que le fait de limiter un point à être mis en correspondance avec plus de deux points n'a que peu d'influence sur la pertinence globale du descripteur. Afin de garantir un temps de calcul raisonnable tout en gardant une certaine robustesse des résultats, nous utilisons une bande diagonale de 8 points.

7.4.5 Descripteur par cartes de normales (CN)

Le troisième descripteur que nous avons proposé dans le chapitre 5 est basé sur des cartes de normales. De ce fait, plusieurs paramètres entrent en jeu lors de son implémentation. Afin d'optimiser leur choix, nous présentons dans cette section un ensemble d'expérimentations visant à évaluer l'évolution des performances en fonction de ceux-ci.

Expérimentations sur le nombre de cœfficients de Fourier

Le premier paramètre qu'il est nécessaire de régler est le nombre de cœfficients de Fourier utilisé pour caractériser l'allure générale de chaque carte de normales. Afin d'évaluer les performances de notre approche en fonction de ce paramètre, nous proposons dans la figure 7.18 ainsi que dans les tableaux 7.9 des expérimentations effectuées sur les bases *SHREC'07 Watertight* et *SHREC'09 Generic*.



(a) *Descripteur CN - SHREC'07 Watertight* (b) *Descripteur CN - SHREC'09 Generic*

FIGURE 7.18 – *Courbes rappel/précision pour le descripteur MCC en fonction du nombre de cœfficients de Fourier*

(a) *Descripteur CN - SHREC'07 Watertight*

	NN	FT	ST	DCG
8	93	55.51	68.13	83.41
16	93.75	56.79	69.29	84.33
24	**94**	**56.41**	**69.38**	**84.33**
32	92.75	55.71	68.72	83.8

(b) *Descripteur CN - SHREC'09 Generic*

	NN	FT	ST	DCG
8	84.87	45.15	57.91	76.42
16	86.37	47.39	60.02	77.88
24	**86.12**	**48.26**	**60.8**	**78.21**
32	86.25	48.01	60.95	78.08

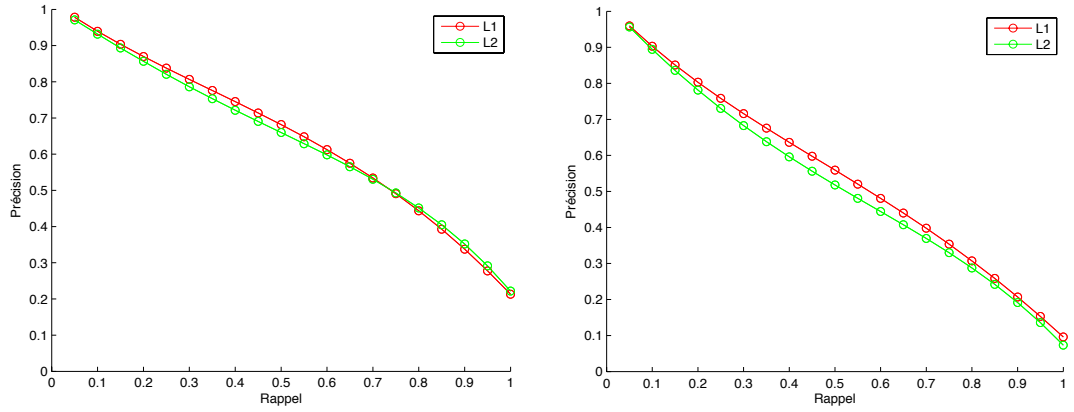
TABLE 7.9 – *Mesures de performances en % pour le descripteur CN en fonction du nombre de cœfficients de Fourier*

Les résultats obtenus montrent que le fait d'augmenter le nombre de cœfficients améliore les performances jusqu'à un certain seuil. Ici, il apparaît que 24 est la limite à ne pas dépasser si nous ne voulons pas dégrader la pertinence des résultats. De ce fait, nous utilisons 24 valeurs pour décrire les cartes de normales, et ceci, quelle que soit la base de données.

Expérimentations sur la mesure de dissimilarité

Afin de comparer deux cartes de normales, nous avons mis en place une mesure de dissimilarité. Afin de choisir la mesure la plus adaptée, nous évaluons les performances des distances L1 et L2 dans la figure 7.19 et les tableaux 7.10.

Les résultats obtenus montrent que la distance L1 garantit une meilleure pertinence de la recherche, et ceci, quelle que soit la base de données. C'est donc celle-ci que nous associons au descripteur par carte de normales.



(a) *Descripteur CN - SHREC'07 Watertight* (b) *Descripteur CN - SHREC'09 Generic*

FIGURE 7.19 – *Courbes rappel/précision pour le descripteur CN en fonction de la mesure de dissimilarité.*

(a) *Descripteur CN - SHREC'07 Watertight*

	NN	FT	ST	DCG
L1	**94**	**56.41**	**69.38**	**84.33**
L2	91.5	55.41	68.59	83.46

(b) *Descripteur CN - SHREC'09 Generic*

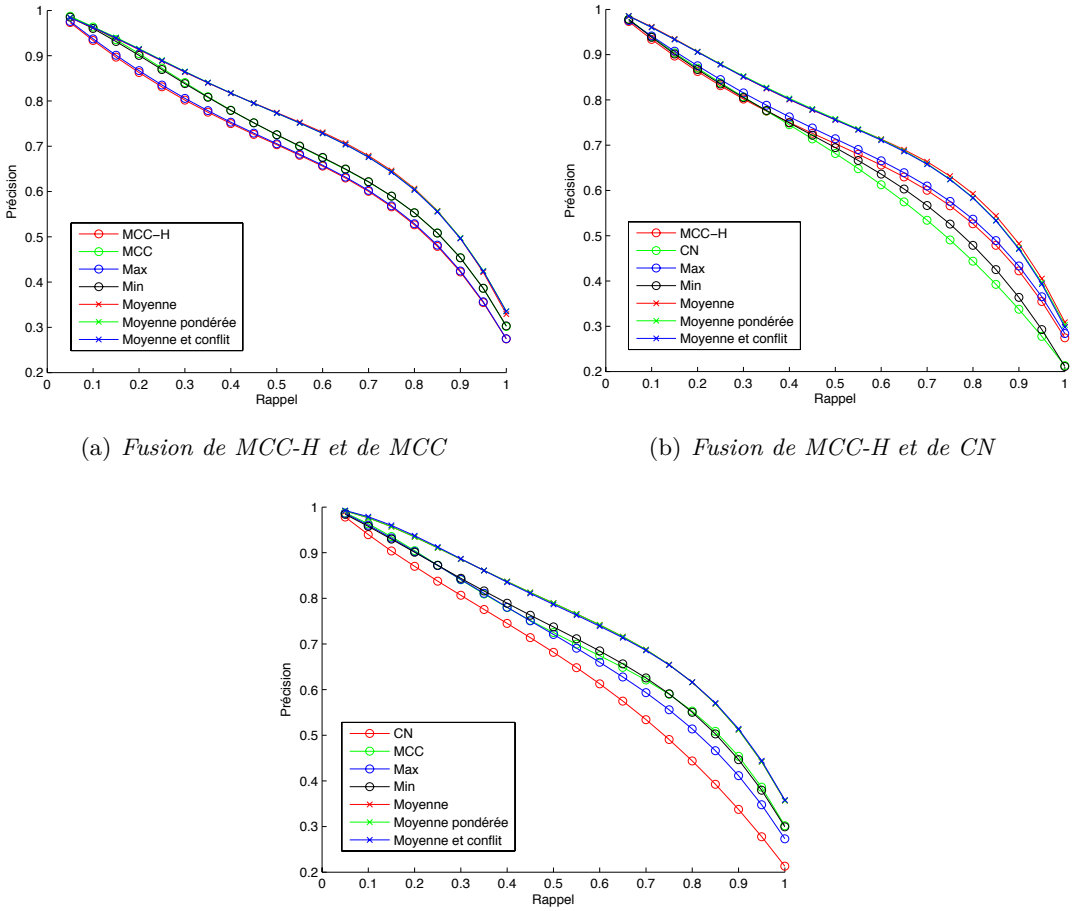
	NN	FT	ST	DCG
L1	**86.12**	**48.26**	**60.8**	**78.21**
L2	85.37	45.64	58.05	76.62

TABLE 7.10 – *Mesures de performances en % pour le descripteur CN en fonction de la mesure de dissimilarité.*

7.4.6 Optimisation par fusion de descripteurs

Nous avons proposé, dans la section 6.5.1 du chapitre 6, une première optimisation permettant d'améliorer la pertinence des résultats en fusionnant les résultats de plusieurs descripteurs. Nous avons en particulier proposé cinq opérateurs d'agrégation, tous basés sur des heuristiques de fusions différentes. Rappelons que nous utilisons les valeurs de confiances suivantes : EP : 0.2, MCC-H : 0.4, MCC et CN : 0.6. Afin d'évaluer chacun d'eux, nous proposons sur la figure 7.20 et les tableaux 7.11 des résultats d'agrégation sur la base *SHREC'07 Watertight*. Notons que des résultats sur la base *Prineton Shape Benchmark* sont visibles dans la section 7.5.1.

Les résultats que nous obtenons montrent que dans chaque cas la fusion permet de gagner en pertinence. En particulier, les opérateurs "par compromis" basés sur la moyenne pondérée ainsi que sur la moyenne et le conflit améliorent nettement les résultats. D'autre part, il est à noter que l'opérateur "pessimiste" *Min* donne toujours des performances supérieures à l'opérateur "Optimiste" *Max*. Finalement, il est plus intéressant d'utiliser le dernier opérateur utilisant une mesure de conflit car il ne nécessite pas de donner une valeur de confiance à chaque source de données.



(a) *Fusion de MCC-H et de MCC*

(b) *Fusion de MCC-H et de CN*

(c) *Fusion de CN et de MCC*

FIGURE 7.20 – *Courbes rappel/précision pour la fusion des résultats de différents descripteurs.*

7.4.7 Optimisation par élagage précoce

Nous avons proposé dans la section 6.5.2 du chapitre 6 une optimisation visant à accélérer la recherche d'objets 2D. Celle-ci est basée sur un élagage précoce contrôlé à l'aide d'un paramètre k. Afin de comprendre l'évolution de la pertinence des résultats en fonction de k, nous proposons dans la figure 7.21 des résultats sur les bases *SHREC'07 Watertight* et *SHREC'09 Generic*. Étant donné que cette optimisation a besoin d'ordonner les descripteurs en fonction de leur complexité, nous utilisons les résultats de la figure 7.12 qui nous donne : $\mathcal{C}(EP) < \mathcal{C}(MCC - H) < \mathcal{C}(CN) < \mathcal{C}(MCC)$.

Les résultats de l'optimisation par élagage précoce montrent l'intérêt de cette méthode. En effet, nous nous rendons compte qu'un descripteur au pouvoir discriminant limité, de type MCC-H ou EP, permet d'éliminer rapidement les objets 3D qui ne sont pas la cible de l'utilisateur. Ainsi, ils laissent le descripteur puissant (CN ou MCC) analyser plus finement les modèles 3D pertinents et uniquement ceux-ci. La figure 7.21 montre qu'une

(a) *Fusion de MCC-H et de MCC*

	NN	FT	ST	DCG
MCC-H	91	60.22	74.54	85.23
MCC	95	64.8	75.96	87.71
Min	94.75	63.55	75.47	87.21
Max	91.5	60.45	74.8	85.55
Moyenne	94.25	66.95	78.89	88.85
Moyenne pondérée	**94.25**	**67.13**	**78.95**	**88.93**
Moyenne et conflit	**94.25**	**66.97**	**78.92**	**88.85**

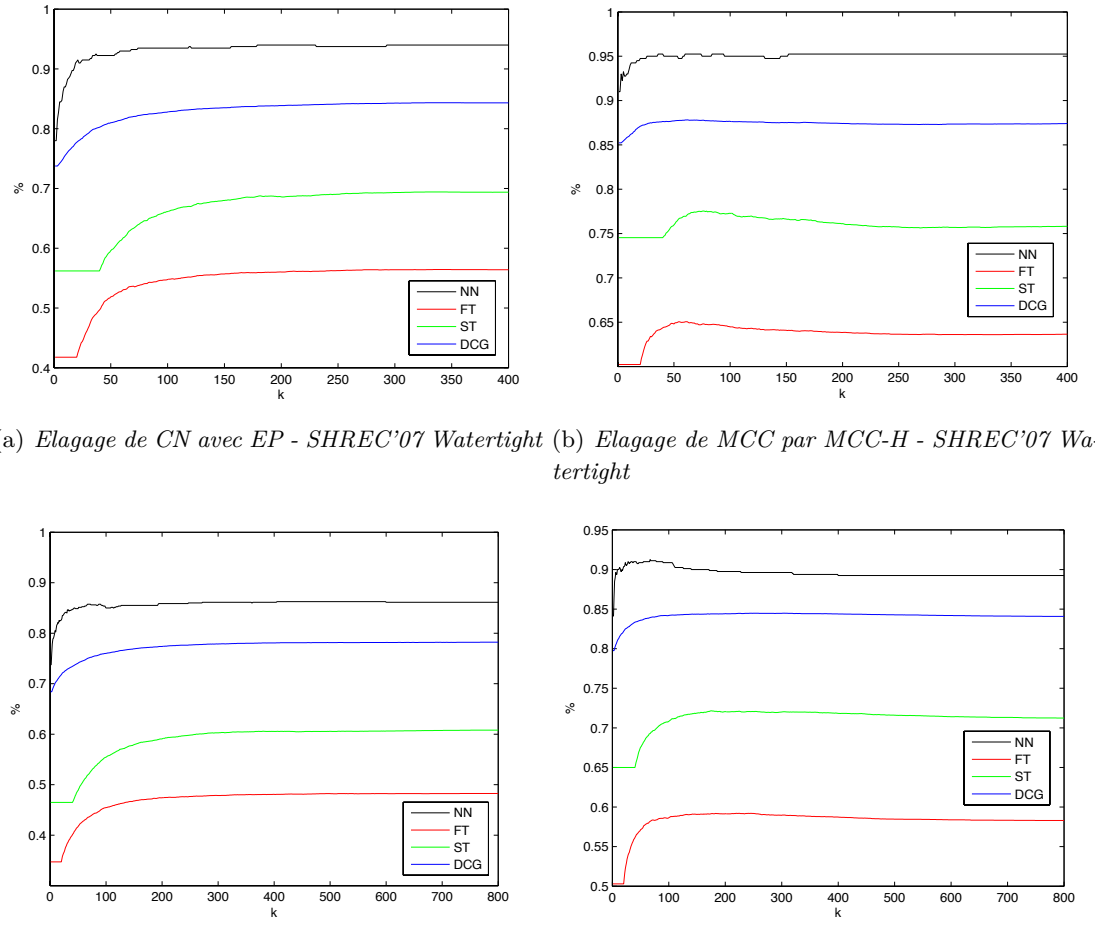
(b) *Fusion de MCC-H et de CN*

	NN	FT	ST	DCG
MCC-H	91	60.22	74.54	85.23
CN	94	56.41	69.38	84.33
Min	94.25	58.29	70.18	84.75
Max	91.25	61.14	75.33	85.93
Moyenne	94.5	65	77.5	88.37
Moyenne pondérée	**95**	**64.8**	**76.84**	**88.25**
Moyenne et conflit	**95**	**64.57**	**76.75**	**88.18**

(c) *Fusion de CN et de MCC*

	NN	FT	ST	DCG
CN	94	56.41	69.38	84.33
MCC	95	64.8	75.96	87.71
Min	95.5	63.74	75.32	87.36
Max	95.25	61.54	74	86.87
Moyenne	97	68.88	79.41	90.06
Moyenne pondérée	**97**	**68.88**	**79.41**	**90.06**
Moyenne et conflit	**97.25**	**69.13**	**79.37**	**90.09**

TABLE 7.11 – *Mesures de performances en % pour la fusion des résultats de différents descripteurs.*



(a) *Elagage de CN avec EP - SHREC'07 Watertight* (b) *Elagage de MCC par MCC-H - SHREC'07 Watertight*

(c) *Elagage de CN avec EP - SHREC'09 Generic* (d) *Elagage de MCC par MCC-H - SHREC'09 Generic*

FIGURE 7.21 – *Évolution du "nearest neighbor (NN)", "first tier (FT)", "second tier (ST)" et "discounted cumulative gain (DCG)" en fonction du paramètre k de l'optimisation par élagage précoce.*

valeur de $k = 75$ est généralement suffisante pour limiter la perte de performance en terme de résultats, et ceci, quelle que soit la base de données. La figure 7.22 montre le gain de temps obtenu avec cette approche d'élagage précoce lorsque nous utilisons respectivement les descripteurs EP et MCC-H pour élaguer CN et MCC pour le test exhaustif de la base *SHREC'09 Generic* (800 objets).

Nous pouvons voir que le gain de temps est quasiment nul pour l'élagage de CN avec EP, dû à leur temps de comparaison très proche. En revanche, l'élagage de MCC avec MCC-H permet d'accélérer de façon considérable la recherche dans la base de données. Ceci vient du fait que le temps de comparaison de MCC-H est très nettement inférieur à celui de MCC (voir figure 7.12).

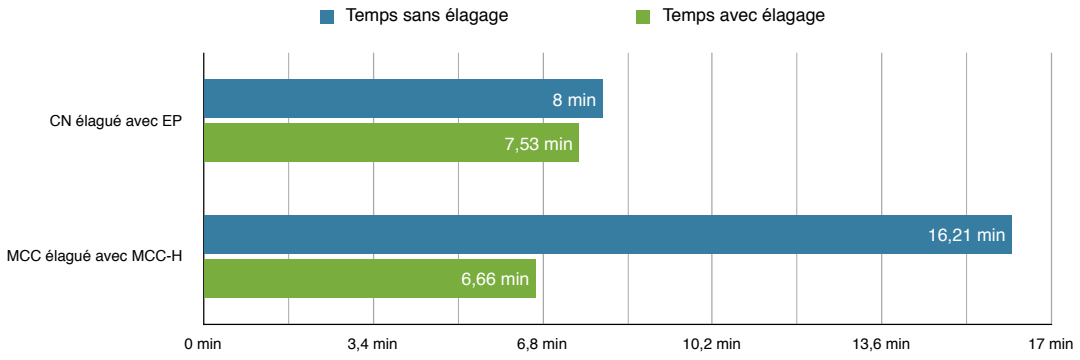


FIGURE 7.22 – *Comparaison du temps de recherche avec et sans élagage précoce. La figure présente le gain de temps avec CN élagué avec EP d'une part, et celui avec MCC élagué avec MCC-H d'autre part.*

7.4.8 Comparaison des nos descripteurs

Étant donné que nous connaissons désormais les paramètres optimaux de nos approches, nous pouvons dresser un tableau récapitulatif des meilleures performances. Les tableaux 7.12 regroupent ces résultats pour l'ensemble de nos descripteurs sur les bases *SHREC'07 Watertight* et *SHREC'09 Generic*, avec et sans optimisations.

Les tableaux récapitulatifs 7.12 montrent la supériorité des approches CN et MCC. Cela vient du fait que la première est une approche "2.5D/3D" capturant, de ce fait, une information de forme plus riche et que la seconde est associée à la "déformation temporelle dynamique" qui lui confère une plus grande adaptabilité.

Si l'on regarde du coté des optimisations, nous pouvons nous rendre compte que les méthodes proposées sont pertinentes. En effet, l'élagage de MCC par MCC-H accélère le temps de recherche sans perdre en robustesse. D'autre part, les résultats de la fusion montrent le grand intérêt de cette optimisation. En effet, dans tous les cas elle permet de rendre les résultats plus pertinents, parfois même de 10% si nous associons MCC-H et CN. Il est intéressant de noter que l'opérateur d'agrégation par conflit que nous avons proposé offre la même pertinence que la moyenne pondérée. Finalement, nous nous rendons compte que les méthodes proposées sont capables de répondre au attentes de l'utilisateur en terme de temps et de robustesse des résultats.

7.4.9 Expérimentation sur des requêtes 2D

Le descripteur par ensemble de pixels ainsi que ceux basés sur les informations de concavités/convexités permettent d'interroger la base de données en utilisant des requêtes de type "photo" ou "dessin au trait". Afin d'évaluer la pertinence de ce type d'approches, nous proposons ici un ensemble d'expérimentations avec des requêtes 2D. Étant donné la supériorité des approches par concavités/convexités, nous utilisons ici un protocole

(a) *SHREC'07 Watertight*

	NN	FT	ST	DCG
EP	78	41.76	56.2	73.76
MCC-H	91	60.22	74.54	85.23
CN	**94**	**56.41**	**69.38**	**84.33**
MCC	**95**	**64.8**	**75.96**	**87.71**
Elagage MCC par MCC-H ($k = 75$)	95	64.78	77.51	87.78
Fusion de CN et MCC (moyenne pondérée)	95	64.8	76.84	88.25
Fusion de MCC-H et CN (moyenne pondérée)	**97**	**68.88**	**79.41**	**90.06**
Fusion de CN et MCC (moyenne + conflit)	95	64.57	76.75	88.18
Fusion de MCC-H et CN (moyenne + conflit)	**97.25**	**69.13**	**79.37**	**90.09**

(b) *SHREC'09 Generic*

	NN	FT	ST	DCG
EP	73.75	34.71	46.5	68.41
MCC-H	84.1	50.28	65.01	79.76
CN	**86.12**	**48.26**	**60.8**	**78.21**
MCC	**90.5**	**57.46**	**67.6**	**83.71**
Elagage MCC par MCC-H ($k = 75$)	91.12	58.32	69.7	84
Fusion de CN et MCC (moyenne pondérée)	91	56.89	70.8	84.2
Fusion de MCC-H et CN (moyenne pondérée)	**93.12**	**62.96**	**75.57**	**87.08**
Fusion de CN et MCC (moyenne + conflit)	90.87	57.18	71.03	84.24
Fusion de MCC-H et CN (moyenne + conflit)	**92.37**	**63.1**	**75.82**	**89.13**

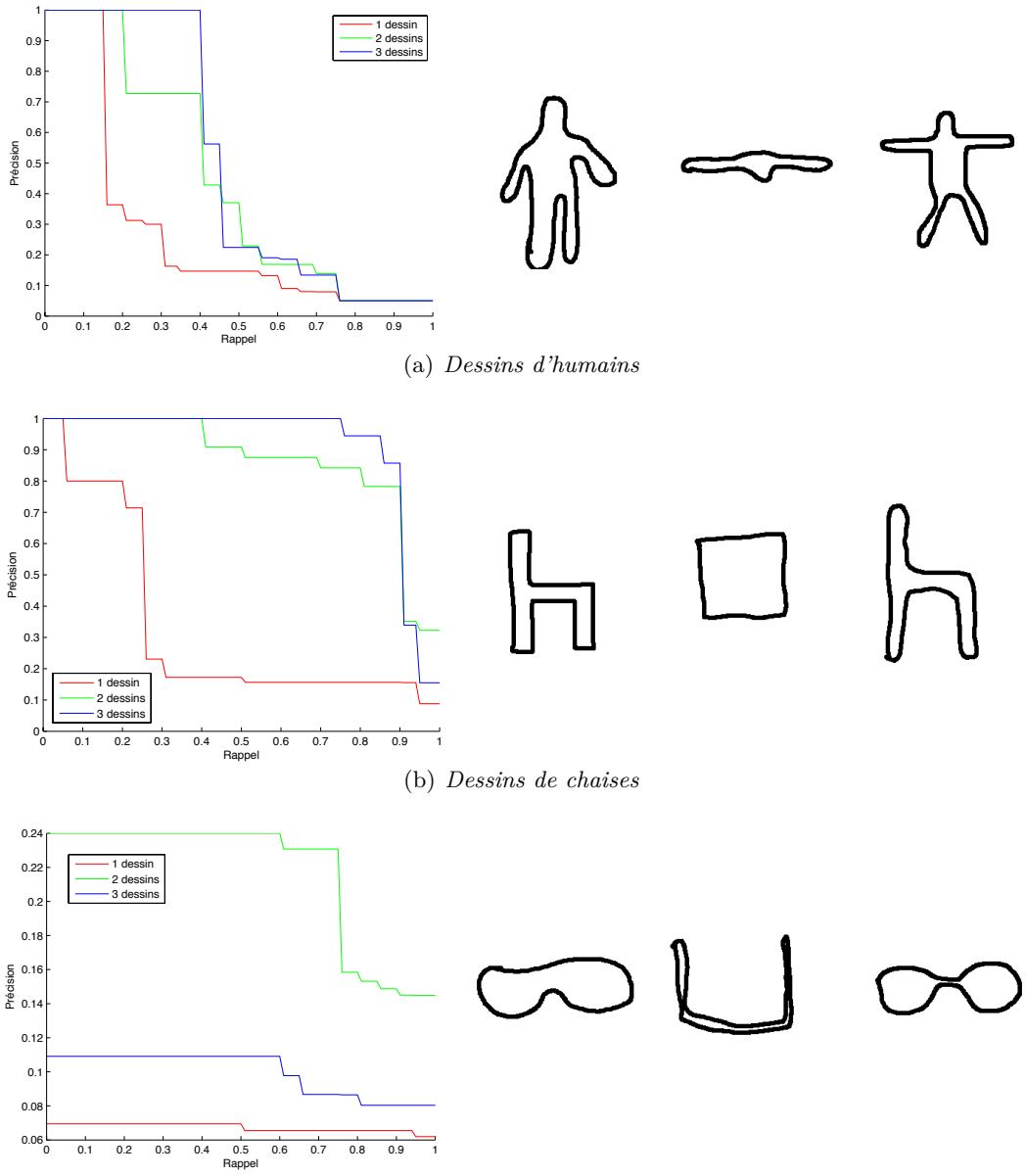
TABLE 7.12 – *Mesures de performances en % pour l'ensemble des descripteurs proposés, avec et sans optimisations.*

expérimental basé sur l'optimisation par élagage précoce avec les descripteurs MCC-H et MCC où le seuil k est fixé à 50.

Expérimentations avec des dessins au trait

Nous avons introduit dans le chapitre 6 un prototype de moteur de recherche d'objets 3D fonctionnant sur terminal mobile. Avec celui-ci, l'utilisateur peut interroger la base de données en utilisant un ou plusieurs dessins au trait. Afin d'évaluer la pertinence des résultats retournés à l'utilisateur, nous proposons dans la figure 7.23 et le tableau 7.13 des expérimentations évaluant la pertinence en fonction du nombre de dessins donné par l'utilisateur.

Pour conclure, les résultats obtenus montrent que l'augmentation du nombre de dessins permet d'améliorer la pertinence des résultats. Cependant, la figure 7.24 montre qu'un seul dessin fournit dans la majorité des cas des résultats suffisants pour l'utilisateur. En effet,



(a) *Dessins d'humains*

(b) *Dessins de chaises*

(c) *Photos de lunettes*

FIGURE 7.23 – *Courbes rappel/précision pour des requêtes de type "dessin au trait" en fonction du nombre d'esquisses fournies par l'utilisateur.*

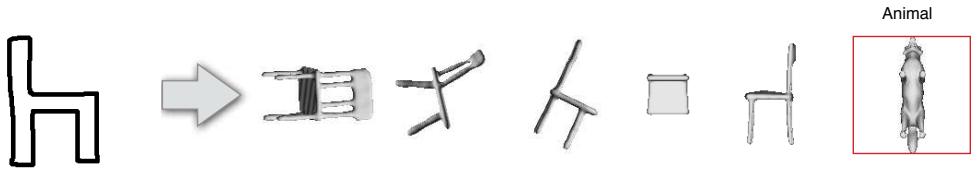
nous pouvons remarquer que les huit premiers résultats répondent totalement à la requête de l'utilisateur.

Expérimentations avec des photos

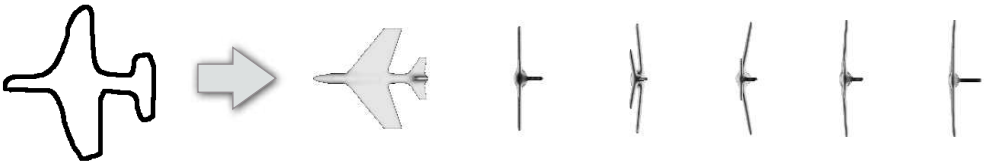
Comme pour les expérimentations précédentes sur les requêtes de type "dessin au trait", nous avons effectué un ensemble de tests avec des requêtes de type "photos". La figure 7.25 et le tableau 7.14 présentent des expérimentations évaluant la pertinence en fonction du nombre de photos donné par l'utilisateur.

	Nombre de dessins	NN	FT	ST	DCG
Humain	1	100	30	30	68.94
	2	100	40	50	79.55
	3	100	45	45	82.57
Chaise	1	100	25	30	67.92
	2	100	80	90	96.37
	3	100	85	90	97.77
Lunette	1	100	15	15	61.03
	2	0	40	50	60.71
	3	100	20	35	70.71

TABLE 7.13 – *Mesures de performances en % pour des requêtes de type "dessin au trait" en fonction du nombre d'esquisses fournies par l'utilisateur.*



(a) *Exemple de résultats pour un dessin d'une chaise*

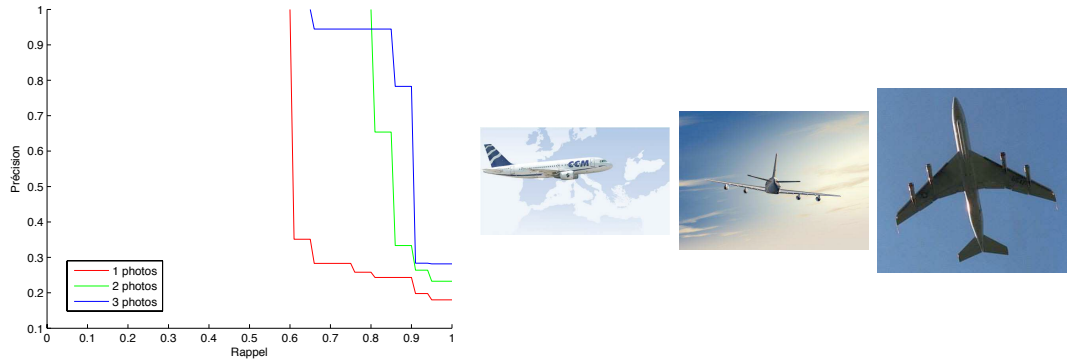


(b) *Exemple de résultats pour un dessin d'un avion*

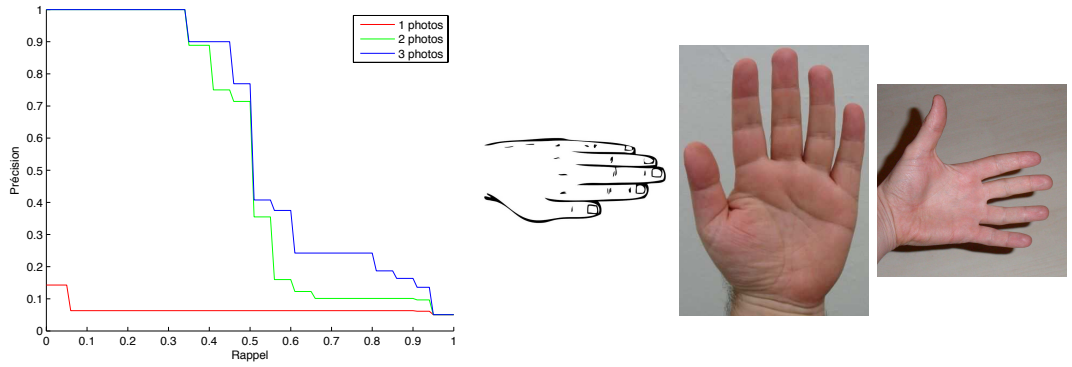
(c) *Exemple de résultats pour un dessin d'une main*

FIGURE 7.24 – *Exemple de résultats pour différentes requêtes de type "dessin au trait". (a), (b) et (c) représentent respectivement une chaise, un avion et une main.*

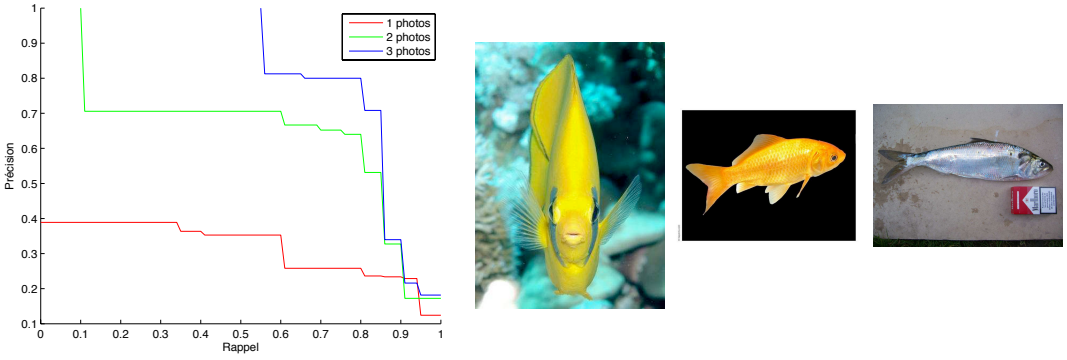
Les résultats montrent clairement que la pertinence des résultats augmente avec le nombre de photos fournies par l'utilisateur. Finalement, la figure 7.26 montre des exemples de résultats obtenus avec des requêtes de type "photos".



(a) *Photos d'avions*



(b) *Photos de mains*



(c) *Photos de poissons*

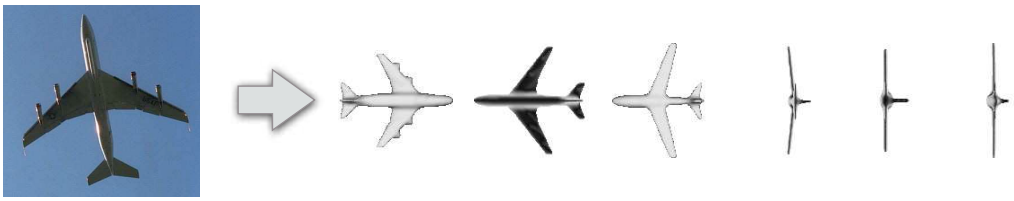
FIGURE 7.25 – *Courbes rappel/précision pour des requêtes de type "photo" en fonction du nombre de clichés fournis par l'utilisateur.*

7.5 Etude comparative

L'ensemble des évaluations que nous avons menées montre la bonne performance de nos différentes approches. Cependant, afin d'avoir une vision objective de nos résultats, nous proposons dans cette section une comparaison de nos méthodes par rapport à celles de la littérature. Pour commencer, nous effectuons en particulier une étude sur la base *Princeton Shape Benckmark* (voir annexe D). Finalement, nous présentons les résultats du concours annuel *SHREC* pour l'année 2009 sur deux bases de données différentes.

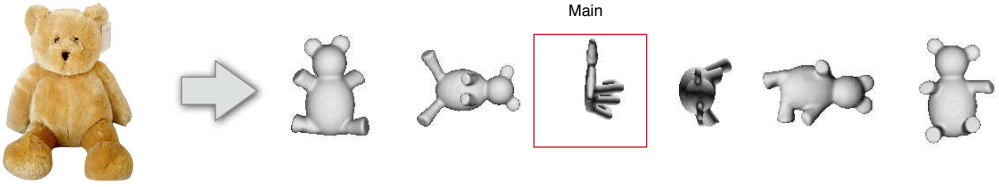
	Nombre de photos	NN	FT	ST	DCG
Avion	1	100	60	65	91.58
	2	100	80	85	96.83
	3	100	85	90	97.68
Main	1	0	5	5	37.46
	2	100	50	55	84.51
	3	100	50	60	87.42
Poisson	1	0	35	60	54.63
	2	100	65	85	83.25
	3	100	80	85	95.37

TABLE 7.14 – *Mesures de performances en % pour des requêtes de type "photo" en fonction du nombre de clichés fournis par l'utilisateur.*



(a) *Exemple de résultats pour une photo d'un avion*

(b) *Exemple de résultats pour une photo d'une main*



(c) *Exemple de résultats pour une photo d'un ourson*

FIGURE 7.26 – *Exemple de résultats pour différentes requêtes de type "photo". (a), (b) et (c) représentent respectivement un avion, une main et un ourson.*

7.5.1 Comparaison avec Princeton

L'évaluation de nos méthodes sur la base de test de *Princeton Shape Benckmark* est présentée dans le tableau 7.15. Notons que ces mesures proviennent de nos propres travaux

et des évaluations de Chaouch [24], Shilane *et al.* [111] et Ansary [7].

	Type	NN	FT	ST	DCG
Fusion de CN et MCC (moyenne pondérée)	**2.5D/3D**	**74.7**	**51.4**	**64.7**	**74.8**
Fusion de MCC-H et CN (moyenne avec conflit)	**2.5D/3D**	**73.9**	**47.5**	**61**	**72.2**
MDLA [24]	2.5D/3D	72.8	48.3	59.9	71.8
MCC	**2D/3D**	**71.1**	**45.2**	**57.2**	**70**
Elagage MCC par MCC-H ($k = 75$)	**2D/3D**	**71.6**	**45**	**55.7**	**69.4**
DLA [24]	2.5D/3D	70.8	43.4	55.6	69.2
CN	**2.5D/3D**	**70.4**	**42.3**	**54.9**	**68.1**
CDLA [24]	2.5D/3D	65.5	37.6	49.7	64.5
LFD [28]	2D/3D	65.7	38	48.7	64.3
MCC-H	**2D/3D**	**62.6**	**38**	**50.5**	**64.5**
EDBA [24]	2.5D/3D	65.4	38.3	49.8	64.1
AVC [35]	2D/3D	62	35.5	45.5	63
3DGA [24]	3D/3D	60.8	34.6	43.7	60.2
EP	**2D/3D**	**59.2**	**32.8**	**43.1**	**60.1**
ESA [24]	2D/3D	57.8	32.6	44.4	60.2
REXT [132]	3D/3D	60.2	32.7	43.2	60.1
DBD [134]	2.5D/3D	59.2	32.9	41.8	58.9
SHD [63]	3D/3D	55.6	30.9	41.1	58.4
GEDT [63]	3D/3D	60.3	31.3	40.7	58.4
SIL [134]	2D/3D	52.8	28.5	38.8	56.3
EXT [107]	3D/3D	54.9	28.6	37.9	56.2
SECSHEL [6]	3D/3D	54.6	26.7	35	54.5
VOXEL [134]	3D/3D	54	26.7	35.5	54.3
SECTORS [6]	3D/3D	50.4	24.9	33.4	52.9
CEGI [59]	3D/3D	42	21.1	28.7	47.9
EGI [51]	3D/3D	37.7	19.7	27.7	47.2
D2 [90]	3D/3D	31.1	15.8	23.5	43.4
SHELLS [6]	3D/3D	22.7	11.1	17.3	38.6

TABLE 7.15 – *Mesures de performances en % de nos approches (en gras) et des méthodes existantes de la littérature. Nous présentons en particulier nos quatre descripteurs ainsi que les résultats obtenus avec nos deux optimisations.*

La comparaison sur la base de *Princeton Shape Benchmark* montre la robustesse de nos deux principales approches, à savoir CN et MCC. Notre méthode par ensemble de pixels reste bien classée, malgré sa simplicité, et MCC-H offre des résultats légèrement en

dessous de la méthode LFD. Nous pouvons noter que l'optimisation par élagage précoce atteint des performances similaires à MCC tout en gagnant en vitesse (grâce à l'élagage). Finalement, l'optimisation basée sur l'agrégation de résultats offre des résultats très pertinents qui surclassent, de 1 à 3 % en fonction des mesures, toutes les autres méthodes.

Il est intéressant de noter que notre méthode MCC offre des résultats très proches de MDLA. Celle-ci utilise 20 vues et des images de profondeurs (2.5D) pour décrire la forme d'un objet, tandis que nous utilisons une méthode "2D/3D" basé sur seulement 6 vues. Notons aussi que les méthodes "2.5D/3D" ou "2D/3D" donnent de meilleurs résultats que les approches "3D/3D". Finalement, nous pouvons remarquer que les méthodes les plus robustes (MDLA, MCC, DLA, CN) utilisent toutes une caractérisation multi-vues.

Pour conclure, nous pensons que la force de nos approches vient de notre méthode d'alignement, de la caractérisation multi-vues de la forme et pour MCC de la "déformation temporelle dynamique" appliquée sur des données multi-échelles.

7.5.2 Concours SHREC

Nous avons participé au cours des années 2007, 2008 et 2009 au concours annuel *SHREC*. Celui-ci évaluant de manière totalement objective plusieurs approches, il nous a semblé pertinent de fournir l'un des résultats. Le tableau 7.16 présente les résultats sur la base *Shrec'09 Watertight* originalement proposés par Hartveldt *et al.* [46].

	Précision FT	Rappel FT	Précision ST	Rappel ST
Elagage de MCC avec MCC-H ($k = 200$) - 9 vues	**81**	**54**	**51**	**68**
Compound-SID-CMVD 3	77	52	52	70
Compound-SID-CMVD 2	76	51	51	68
Elagage de MCC avec MCC-H ($k = 50$) - 9 vues	**74**	**49**	**48**	**64**
Compound-SID-CMVD 1	74	49	48	64
MR-SPRH-UDR 1	74	49	48	64
BF-SIFT 1	72	48	48	64
Elagage de MCC avec MCC-H ($k = 50$) - 3 vues	**71**	**48**	**45**	**60**
Compact Multi View Descriptor 1	69	46	47	62
MCC - 9 vues	**68**	**46**	**45**	**61**
ERG 2	61	41	40	53
ERG 1	56	37	36	49
Bag Of Words 1	29	19	17	23
Concentric Bag Of Words 2	25	17	16	21

TABLE 7.16 – *Résultats, en terme de précision et rappel (FT : 10 et ST : 20), du concours annuel SHREC sur la base Shrec'09 Watertight.*

Notons que notre approche est placée en première position. Le lecteur pourra trouver les autres résultats dans les évaluations faites par Kalyanaraman et Ramani [58], Muthuganapathy et Ramani [73], Biasotti et Attene [12], J.Hartveldt *et al.* [55] et Godil *et al.* [45].

7.6 Conclusion

Ce chapitre a présenté les expérimentations qui ont guidé nos choix dans le développement des différentes contributions de notre thèse. Grâce à ces différents tests, nous avons pu paramétrer nos approches pour qu'elles offrent des résultats pertinents tout en limitant l'espace de stockage nécessaire. Nous nous sommes en particulier arrêtés sur les méthodes de normalisation, le nombre de vues, les distances de dissimilarité ainsi que sur des paramètres propres à chaque méthode.

Étant donné que les trois descripteurs proposés dans le chapitre 5 ne garantissent pas l'invariance envers les transformations géométriques, nous avons mis en place une nouvelle méthode d'alignement d'objets 3D. Les résultats expérimentaux proposés dans ce chapitre ont clairement montré que notre approche permet de définir une pose cohérente pour les modèles d'une base de données, et ceci, de manière plus robuste que les méthodes classiques utilisant une analyse en composantes principales.

D'autre part, ce chapitre a permis de confronter nos quatre méthodes entre elles ainsi qu'avec des approches existantes de la littérature. Les résultats ont montré la supériorité de l'approche basée sur les convexités/concavités ainsi que les bonnes performances des cartes de normales. Pour le premier descripteur, c'est l'utilisation de la "déformation temporelle dynamique" qui a permis l'obtention d'une telle pertinence. Pour le second, il s'agit plutôt du caractère "2.5D/3D" de l'approche qui lui confère de bons résultats, car pouvant décrire plus finement la forme des objets 3D.

Les optimisations présentées dans le chapitre 6 permettent d'accélérer la recherche d'objets 3D ainsi que la pertinence des résultats. D'une part, les expérimentations menées sur l'élagage précoce ont montré qu'il était facile de combiner deux descripteurs dans le but d'accélérer la recherche. D'autre part, la fusion des résultats de deux approches permet dans la plupart des cas d'améliorer significativement la pertinence des résultats pour un coût qui n'est cependant pas nul, car nécessitant le calcul d'au moins deux listes de résultats.

Pour conclure, la comparaison sur la base de *Princeton Shape Benchmark* a montré l'intérêt de nos descripteurs par rapport à la littérature actuelle. Finalement, nous pouvons voir la pertinence de notre méthode basée sur les informations de convexités/concavités grâce aux résultats objectifs produits par le concours *SHREC*.

CHAPITRE 8

Conclusions et perspectives

Dans cette thèse, nous nous sommes intéressés aux problématiques associées à l'indexation et à la recherche d'objets 3D dans les bases de données. Nos objectifs se sont portés sur le développement de nouvelles approches permettant d'interroger une bibliothèque de modèles 3D à partir de requête en deux, ou en trois dimensions. En particulier, nous avons cherché à réduire le temps nécessaire à la recherche sans pour autant diminuer la pertinence des résultats. Pour cela, nous avons apporté plusieurs contributions à la chaîne de traitement classique, qui vont de la normalisation des objets 3D à leur description, en passant par leur recherche.

Conclusions

Normalisation et estimation de l'alignement

Nous avons proposé dans un premier temps une méthode permettant de normaliser en position, en échelle et en rotation un objet en trois dimensions. Nos méthodes d'indexation étant basées sur des approches "2.5/3D" et "2D/3D", caractérisant la forme à partir de projections, nous avons élaboré un traitement bipartite qui leur est adapté. La première phase introduit l'utilisation de la sphère minimale englobante pour définir la position et la taille du modèle tout en garantissant une résolution maximale des silhouettes. Dans un second temps, nous avons élaboré un estimateur qui se fait le juge de la qualité de plusieurs poses d'un même modèle 3D, dans le but de l'aligner. Celui-ci est basé sur un critère de minimisation de l'aire des projections obtenues à partir des sommets d'un octaèdre. Ainsi, nous avons été capables de normaliser et d'aligner correctement les objets d'une base de données d'objets 3D.

Les caractéristiques principales de notre contribution sur cette étape de prétraitements sont les suivantes :

- **Adaptée aux méthodes à base de projection :** L'utilisation de la sphère minimale englobante garantit une bonne qualité des silhouettes 2D, en terme de résolution, dans les approches multi-vues. Ainsi, elle est particulièrement bien adaptée aux approches "2.5/3D" et "2D/3D". De ce fait, pour ces méthodes, il est plus intéressant de définir le centre et l'échelle d'un objet 3D avec celle-ci, plutôt qu'avec une boite englobante, par exemple.

- **Proche de la perception humaine :** L'estimateur que nous avons proposé a été conçu en accord avec des résultats existants en psychologie cognitive qui vise à comprendre la représentation mentale des objets 3D. Grâce à ce lien fort avec la perception humaine, notre méthode d'alignement est bien adaptée aux requêtes 2D de type "dessin au trait". De plus, cet estimateur à l'avantage de choisir, a priori, la pose du modèle parmi plusieurs alignements, permettant ainsi de réduire l'espace de stockage nécessaire aux signatures.

- **Efficace pour normaliser et aligner des objets 3D :** Les expérimentations que nous avons menées montrent l'intérêt de notre normalisation par rapport aux méthodes de la littérature actuelle. D'autre part, l'estimateur que nous avons proposé permet d'améliorer la robustesse des résultats par rapport aux approches classiques utilisant l'analyse en composantes principales. Ainsi, les descriptions de forme sont plus pertinentes offrant, de ce fait, une meilleure séparation des classes d'objets.

Indexation d'objets 3D

La recherche d'objets 3D basée sur le contenu est une tâche complexe qui nécessite de décrire les propriétés intrinsèques de chaque modèle. Après une étude de l'état de l'art, nous nous sommes penchés sur les approches "2.5/3D" et "2D/3D" multi-vues qui offrent de nombreuses possibilités de requête. Nous avons en particulier proposé trois nouveaux descripteurs caractérisant chacun, différemment, la forme d'un objet. Nous avons pour cela utilisé des informations de silhouettes, de convexités et de concavités des contours, mais aussi d'orientations de la surface du modèle 3D.

Les principales contributions que nous avons apportées à l'indexation d'objet 3D sont les suivantes :

- **Approche ensembliste :** Nous avons introduit un premier descripteur qui décrit la forme d'une silhouette à l'aide d'un ensemble de pixels. Celui-ci permet d'appliquer des opérations ensemblistes sur les signatures, afin de les comparer. Cette représentation intuitive a fait l'objet d'une compression sans perte visant à réduire l'espace de stockage et à optimiser le temps de comparaison nécessaire à la recherche.

- **Convexités et concavités des contours :** Dans un second temps, nous avons utilisé des informations de convexités et de concavités pour décrire le contour des projections de nos objets 3D. Ces mesures ont permis de définir deux descripteurs apparentés, basés sur une compression par histogrammes et par DCT. Afin de caractériser plus finement les contours, nous utilisons une mesure multi-échelles des informations de convexités/concavités.

- **Carte de normales :** Nous avons proposé un dernier descripteur de type "2.5D/3D". En effet, celui-ci utilise, en plus des projections de l'objet 3D, l'orientation de sa surface. Ces informations sont regroupées dans une carte de normales que nous

décrivons à l'aide de cœfficients de Fourier. Ainsi, la signature est synthétique tout en garantissant une faible perte des caractéristiques de forme. Par son aspect 2.5D ce descripteur rend néanmoins difficile les requêtes à partir de dessins ou de photos.

Recherche d'objets 3D

Nous avons proposé dans cette troisième partie des méthodes permettant l'interrogation d'une base de données. Pour cela, nous avons proposé, pour chacun de nos descripteurs, des mesures de dissimilarité qui leur sont adaptées. De plus, nous nous sommes penchés sur l'optimisation du temps de recherche, mais aussi sur l'amélioration de la pertinence des résultats à travers une optimisation par fusion des résultats et par élagage précoce. Finalement, nous avons prouvé la faisabilité de notre processus de recherche en développant un prototype fonctionnant sur un terminal mobile.

Les points forts de nos contributions sur la recherche d'objets 3D sont :

- **Des mesures de dissimilarité adaptées :** Afin de comparer nos différents descripteurs de forme, nous avons, pour chacun d'entre eux, défini une mesure de dissimilarité. Notre première signature étant basée sur une représentation ensembliste, nous avons tout naturellement utilisé un opérateur de différence pour comparer deux descripteurs. La mise en correspondance des informations de convexités/concavités, qui n'utilisent pas les histogrammes, est basée sur une "déformation temporelle dynamique". Grâce à l'élasticité que celle-ci amène, ce descripteur est particulièrement performant sur les bases d'objets "articulés". Finalement, les descripteurs par histogrammes de convexités/concavités et par carte de normales utilisent une distance L1. Celle-ci permet de rendre rapide la comparaison de ces signatures.

- **Optimisation du temps et de la pertinence :** L'utilisateur souhaitant toujours des résultats pertinents en un minimum de temps, nous avons proposé deux optimisations. La première utilise des opérateurs d'agrégation pour fusionner les résultats issus de deux descripteurs, tandis que la seconde permet d'éliminer rapidement les objets 3D ne répondant pas à la recherche de l'utilisateur. Grâce à ces optimisations, nous pouvons renvoyer rapidement les résultats que souhaite l'utilisateur.

- **Pertinence des résultats :** Les différentes expérimentations que nous avons menées montrent que notre processus d'interrogation est robuste. En effet, malgré l'utilisation de méthodes "2.5/3D" et "2D/3D", nous avons obtenu des résultats plus que satisfaisants par rapport aux autres approches de la littérature. En particulier, les méthodes par convexités/concavités et par carte de normales ont montré leur force à décrire la forme des objets 3D. Finalement, la faisabilité de notre méthode a été illustrée par notre prototype de recherche fonctionnant sur terminal mobile.

Perspectives

Au cours de cette thèse, nous avons développé et mis au point de nouvelles techniques d'indexation et de recherche d'objets 3D. Étant donné que nos diverses contributions ont été guidées par des choix, souvent basés sur des contraintes, mais parfois arbitraires, des perspectives diverses et variées s'offrent à nous.

Pour la normalisation et l'estimation de l'alignement

- **Recherche de la pose naturelle :** L'approche d'alignement que nous avons proposée permet de définir la pose d'un objet 3D de façon robuste. Cependant, même si celle-ci essaye d'approcher la perception humaine, elle ne permet pas de comprendre la fonctionnalité de l'objet. Afin d'améliorer cette méthode, il pourrait être intéressant de classer, avant cette étape d'estimation de la pose, les objets selon qu'ils contiennent, ou non, des possibilités d'alignements privilégiées. Pour cela, nous pourrions travailler sur une méthode proche de celle proposée par Pu et Ramani [99] qui vise à trouver les surfaces de contacts "virtuelles" d'un objet 3D. Ainsi, cette classification permettrait d'adapter la méthode d'alignement à chaque type d'objets.

- **Réduction du nombre de permutations des axes :** La méthode d'estimation de la pose n'étant pas robuste envers l'ordre et le signe des axes du repère 3D, nous calculons les 48 permutations possibles de ceux-ci. Comme l'ont montré les expérimentations à ce sujet, cette étape semble indispensable pour garantir des résultats robustes. Afin de limiter le nombre de cas à traiter, nous pourrions étudier les valeurs propres fournies par l'analyse en composantes principales et ainsi, associer un degré de confiance à chaque axe. Grâce à celles-ci, nous pourrions dans certains cas ne calculer qu'un sous-ensemble des 48 permutations. Il faut cependant noter qu'un tel traitement nécessiterait d'appliquer les permutations aux objets 3D de la base de données et non plus à la requête, augmentant de ce fait le volume de stockage nécessaire.

Pour les descripteurs de forme

- **Ajout d'élasticité dans la différence symétrique :** Les expérimentations menées sur le descripteur à base d'ensemble de pixels ont montré que la différence symétrique était trop simple pour offrir des résultats très pertinents. Afin d'améliorer cette approche, il serait intéressant d'ajouter un peu d'élasticité dans la mesure de dissimilarité. Nous pourrions par exemple utiliser une distance de type EMD ou faire de la mise en correspondance par bloc plutôt que par pixel. Ces techniques pourraient aussi faire l'objet d'accélération sur carte graphique (calcul sur GPU).

- **Remplacer la "déformation temporelle dynamique" par une métrique :** La distance que nous utilisons pour comparer les informations de convexités et de concavités est basée, pour l'un de nos descripteurs, sur la "déformation temporelle

dynamique". Même si cette approche permet d'améliorer significativement les résultats, elle ne garantit pas d'obtenir toujours une métrique. De ce fait, il est impossible d'accélérer la recherche par des techniques de pivots. Afin de résoudre ce problème, il serait intéressant de tester des méthodes d'appariement robuste définies comme étant des métriques. Nous pensons en particulier aux approches par noyaux dépendant du contexte proposées par Sahbi *et al.* [106].

– **Décrire les cartes de normales par des séquences d'états :** La comparaison de nos approches avec celles de la littérature a montré que l'approche de Chaouch [24] basée sur les images de profondeurs donnait de très bons résultats. Nous pensons que ces performances proviennent, en partie, du codage de ces images par des séquences d'états qu'il est possible de mettre en correspondance avec des techniques de "programmation dynamique". Nous proposons donc de tester ce type d'approche sur nos cartes de normales pour voir le gain de performances que nous pourrions obtenir.

Pour la recherche d'objets 3D

– **Définir une structure d'index pour "passer à l'échelle" :** Les expérimentations que nous avons menées sur le temps nécessaire à la recherche d'objets 3D, à l'aide de nos descripteurs, montrent que ceux-ci ne passent pas à l'échelle. En effet, l'interrogation d'une base de données comportant plusieurs milliers d'objets nécessiterait un temps trop important pour l'utilisateur. Afin de résoudre ce problème, il serait intéressant de structurer le stockage des descripteurs à l'aide d'un arbre de recherche. La question importante serait alors le choix du critère permettant de structurer nos différentes signatures. Nous pourrions aussi tester les techniques de pivot ou évaluer des méthodes de recherche par classe plutôt que par objet.

– **Ajouter le flux vidéo comme type de requête :** Nous avons proposé dans cette thèse trois types de requêtes qui vont des objets 3D aux photos, en passant par les dessins. Il pourrait cependant être intéressant de proposer un quatrième type basé sur les flux vidéo. Ainsi, il serait possible de placer un objet, une photo ou un dessin devant une caméra numérique afin d'interroger la base de données. Pour cela, il serait nécessaire de mettre en place des techniques permettant la séparation de l'objet et de son arrière-plan.

– **Évaluer la méthode sur des bases de dessins :** Finalement, les expérimentations que nous avons menées sur les requêtes 2D de type dessin et photo ont été limitées, ceci car nous ne possédions pas de collections de dessins en corrélation avec nos bases de données d'objets 3D. Ainsi, il serait judicieux de constituer une bibliothèque de dessins et de photos représentant nos objets 3D. Celle-ci permettrait alors d'approfondir nos tests sur les requêtes 2D. Dans un esprit similaire, il serait intéressant d'évaluer l'ergonomie ainsi que la pertinence du moteur de recherche embarqué dans notre prototype dans des conditions d'utilisations réelles.

ANNEXE A

Etat de l'art sur les mesures de dissimilarité

La mesure de dissimilarité permet, au moyen d'une distance d, de calculer la dissimilarité entre deux signatures d'objets 3D représentées par des vecteurs. Soit $\mathcal{P}$ un ensemble de points, et d sa fonction associée, telle que $d : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}^+$. Alors, d peut vérifier les propriétés suivantes définies par Copson [30] :

1. *L'identité* : $\forall x \in \mathcal{P},\ d(x,x) = 0$
2. *La positivité* : $\forall (x,y) \in \mathcal{P} \times \mathcal{P} | x \neq y,\ d(x,y) > 0$
3. *La symétrie* : $\forall (x,y) \in \mathcal{P} \times \mathcal{P},\ d(x,y) = d(x,y)$
4. *L'inégalité triangulaire* : $\forall (x,y,z) \in \mathcal{P} \times \mathcal{P} \times \mathcal{P},\ d(x,z) \leqslant d(x,y) + d(y,z)$

Par définition, d est appelée métrique si toutes les propriétés sont vérifiées, tandis que d est appelée semi-métrique si l'inégalité triangulaire n'est pas respectée. D'après Barros *et al.* [10], l'inégalité triangulaire, qui signifie que la distance pour aller de x à z est inférieure ou égale à la distance pour aller de x à z en passant par un autre endroit, améliore les résultats d'une recherche basée sur le contenu.

Étant donné qu'une multitude de distances d peuvent être utilisées pour comparer deux descriptions d'objets 3D, nous introduisons dans cette section les distances les plus couramment utilisées dans la littérature.

Norme de Minkowski

Aussi appelée norme L_n, elle permet de calculer une distance entre deux vecteurs $v_1 = (x_1,\ x_2,\ ...,\ x_i)$ et $v_2 = (y_1,\ y_2,\ ...,\ y_i)$. la norme de Minkowski est définie de la manière suivante :

$$L_n = \sqrt[n]{\sum_{i=1}^{n} |x_i - y_i|^n} \tag{A.1}$$

Les utilisations les plus courantes de cette norme sont les distances Euclidienne, de Manhattan et de Tchebychev qui correspondent, respectivement, aux normes L_1, L_2 et L_∞ (voir figure A.1).

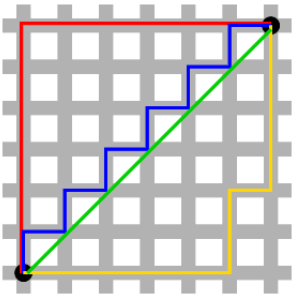


FIGURE A.1 – *Représentation de la distance de Manhattan en rouge, jaune et bleu et de la distance Euclidienne en vert.*

Distance de Hausdorff

La distance de Hausdorff permet de définir un critère de dissimilarité entre deux formes. Considérons le point de v_1 le plus éloigné de v_2 d'une part, et le point de v_2 le plus éloigné de v_1. Alors, la distance de Hausdorff est la plus grande valeur des deux. On la formule de la manière suivante :

$$d_h(v_1, v_2) = max\{\sup_{p \in v_1} \inf_{q \in v_2} (d_{pq}), \sup_{q \in v_2} \inf_{p \in v_1} (d_{pq})\} \tag{A.2}$$

où d_{pq} représente la distance (on peut choisir L_1 ou L_2) entre p et q.

Distance du χ^2

La loi du χ^2 est une méthode formelle permettant de calculer une distance entre deux histogrammes P et Q de n composantes. Elle permet d'apprécier le degré d'adéquation, ou de ressemblance, entre deux distributions et permet de ce fait de savoir si elles proviennent du même jeu de données. Elle est formulée par Hetzel *et al.* [48] de la manière suivante :

$$\chi^2(P, Q) = \sum_{i=1}^{n} \frac{(p_i - q_i)^2}{q_i} \tag{A.3}$$

Distance de Bhattacharyya

Définie par Thacker *et al.* [119], cette distance permet de comparer deux histogrammes normalisés, P et Q de n composantes, afin d'en donner une mesure de dissimilarité. Elle est formulée comme suit :

$$d_b(P, Q) = 1 - \sum_{i=1}^{n} \sqrt{p_i}\sqrt{q_i} \tag{A.4}$$

Distance EMD

La distance "EMD", ou *Earth Mover's Distance*, permet de résoudre les problèmes dis : de "transports" entre histogrammes. Il s'agit en fait de calculer la quantité de "travail" nécessaire pour passer de l'histogramme H_1 à l'histogramme H_2. Intuitivement, supposons que le premier histogramme correspond à des tas de terre disposés aléatoirement dans un espace, tandis que le second correspond à des trous dans le même espace. Alors, la distance "EMD" correspond au travail minimum, nécessaire pour remplir les trous avec la terre. On comprend bien que pour définir cette mesure, il faut au préalable définir une distance d dans l'espace des histogrammes. Pour cette mesure, la taille des deux histogrammes peut-être différente.

D'après Rubner *et al.* [105], la distance "EMD" peut être définie comme suit. Soit deux histogrammes $P = \{(p_1, w_{p1})\}, ..., \{(p_m, w_{pm})\}$ et $Q = \{(q_1, w_{q1})\}, ..., \{(q_n, w_{qn})\}$, P contentant m composantes p_i de poids w_{pi} et Q contenant n composantes q_j de poids w_{pj}. Alors, le problème revient à trouver le transport $F = f_{ij}$ qui minimise le travail suivant :

$$\mathcal{W}(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij} \tag{A.5}$$

où d_{ij} est la distance entre deux composantes i et j et avec les contraintes suivantes :

$$\begin{aligned}
&f_{ij} \geqslant 0 \ avec \ 1 \leqslant i \leqslant m \ 1 \leqslant j \leqslant n \\
&\sum_{j=1}^{n} f_{ij} \leqslant w_{pi} \ avec \ 1 \leqslant i \leqslant m \\
&\sum_{i=1}^{m} f_{ij} \leqslant w_{qj} \ avec \ 1 \leqslant j \leqslant n \\
&\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = min(\sum_{i=1}^{m} w_{pi}, \sum_{j=1}^{n} w_{qj})
\end{aligned} \tag{A.6}$$

La première contrainte permet de limiter le transport à une direction, de P vers Q. Les deux suivantes, quant à elles, interdisent qu'une composante de P ne déplace, ou qu'une composante de Q ne reçoive, plus que son poids. Finalement, la dernière contrainte force à déplacer le maximum de composantes possibles. La distance "EMD" peut alors s'écrire comme :

$$EMD(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}} \tag{A.7}$$

Le facteur de normalisation est introduit afin d'éviter de favoriser les signatures de dimension moins élevée lorsque celles-ci sont de tailles différentes.

ANNEXE B

Calcul de la sphère minimale englobante

Étant donné un ensemble fini de points noté $\mathcal{P}$ dans $\mathbb{R}^d$, nous cherchons la sphère $\mathcal{S}$ (en dimension d) telle que :

$$\mathcal{S}(c, r) = \{p \in \mathbb{R}^d,\ ||p - c|| \leq r\} \tag{B.1}$$

où $c \in \mathbb{R}^d$ est son centre et $r \in \mathbb{R}^+$ est son rayon. Pour un ensemble de points $\mathcal{T}$ et un point $c \in \mathbb{R}^d$, nous notons $\mathcal{S}(c, P)$ la sphère telle que :

$$\mathcal{S}(c, max_{p \in \mathcal{P}}||p - c||) \tag{B.2}$$

Et notons l'ensemble de points $\mathcal{T}$ à une distance r du centre c, "support", avec $\mathcal{T} \subseteq \mathcal{P}$. En notant $\partial\mathcal{S}$ l'enveloppe, ou surface, de la sphère $\mathcal{S}$, nous avons :

$$\begin{aligned} &\mathcal{T} \in \partial\mathcal{S}(c, \mathcal{P}) \\ &\mathcal{P} \subset \mathcal{S}(c, \mathcal{P}) \end{aligned} \tag{B.3}$$

Finalement, notons $convex(\mathcal{T})$ l'enveloppe convexe de $\mathcal{T}$, c'est-à-dire l'ensemble des barycentres à cœfficients positifs ou nuls de $\mathcal{T}$. La sphère minimale englobante d'un ensemble fini de points $\mathcal{P} \in \mathbb{R}^d$ est alors définie comme étant la sphère de rayon minimum contenant l'ensemble des points de $\mathcal{P}$. L'existence de cette sphère est unique Welzl [138] :

Lemme B.1. ***Fischer et Gartner [37]*** *Soit $\mathcal{T} \in \partial\mathcal{S}(c, \mathcal{P})$. Alors $\mathcal{S}(c, \mathcal{P})$ est la sphère minimale englobante si et seulement si $c \in convex(\mathcal{T})$.*

De plus :

Lemme B.2. ***Fischer et Gartner [37]*** *Si $\mathcal{T} \in \partial\mathcal{S}(c, \mathcal{P})$ avec $c \in convex(\mathcal{T})$. Alors il est possible de définir c comme barycentre avec :*

$$c = \sum_{p \in \mathcal{T}} \lambda_p p,\ \sum_{p \in \mathcal{T}} \lambda_p = 1, \lambda_p \geq 0, \forall p \in \mathcal{T} \tag{B.4}$$

Voir Fischer et Gartner [37] pour les preuves.

Algorithme (voir aussi figure B.1)

L'algorithme de calcul de la sphère minimale englobante est alors simple. Partant d'une sphère ayant pour centre c n'importe quel point $p \in \mathcal{S}$, et pour rayon r la distance maximale entre c et tout les points $p \in \mathcal{P}$. Nous allons réduire cette sphère en itérant deux étapes : **déplacement du centre** c et **retrait, éventuel, d'un point du support** $\mathcal{T}$, jusqu'à atteindre la terminaison définit par le Lemme B.1.

Déplacement du centre c : (Si et seulement si $c \notin convex(\mathcal{T})$)

On déplace c vers le centre de gravité de $\mathcal{T}$ jusqu'à ce que la sphère réduite touche un nouveau point $p \in \mathcal{P}$, dans ce cas, nous insérons alors p dans $\mathcal{T}$.

Retrait, éventuel, d'un point du support $\mathcal{T}$: (Si et seulement si $\exists p \in \mathcal{T},\ \lambda_p < 0$)

On retire le point $p \in \mathcal{T}$ qui à un cœfficient $\lambda_p < 0$ (Lemme B.2).

Finalement, l'algorithme complet s'écrit comme suit :

Algorithme B.1 *Calcul de la sphère minimale englobante*

1: **Algorithme :**
2:
3: $c :=$ n'importe quel point de $\mathcal{P}$ (Lemme B.1)
4: $\mathcal{T} := p$, ou p est le point le plus distant de c appartenant à $\mathcal{P}$
5: **Tant Que** $c \notin convex(\mathcal{T})$ **Faire**
6: **Si** $\exists p \in \mathcal{T},\ \lambda_p < 0$ **Alors**
7: nous retirons un point p de $\mathcal{T}$ avec $\lambda_p < 0$ (Lemme B.2)
8: **Fin Si**
9: nous déplacons c vers le centre de gravité de $\mathcal{T}$
10: **Si** le déplacement atteint un point p de $\mathcal{P}$ **Alors**
11: $\mathcal{T} := \mathcal{T} \cup \{p\}$
12: **Fin Si**
13: **Fin Tant Que**
14: **return** $\mathcal{S}(c, \mathcal{T})$

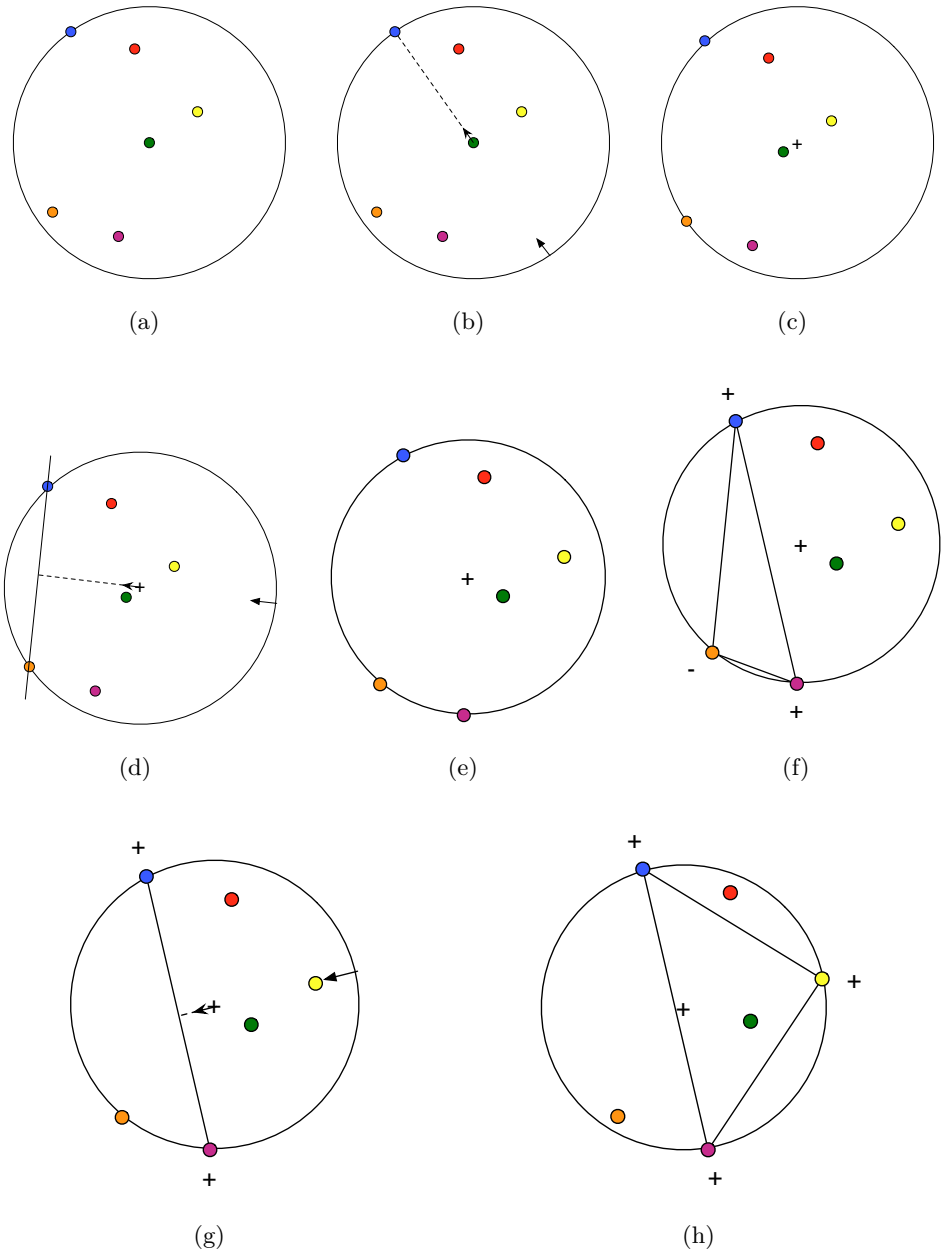


FIGURE B.1 – *Exemple du déroulement de l'algorithme de la sphère minimale englobante en deux dimensions. (a) Nous initialisions la sphère en prenant comme centre un point au hasard et comme rayon, le point le plus distant de ce centre qui définit aussi le support. (b) Nous réduisons la sphère en déplaçant le centre vers le support. (c) Le centre se déplace et le support comporte désormais deux points. (d) Nous réduisons la sphère en déplaçant le centre vers le nouveau support. (e) Le centre se déplace et le support comporte désormais trois points. (f) Même si le support comporte trois points, la condition d'arrêt n'est pas atteinte car le centre n'appartient pas à l'enveloppe convexe du support. Nous supprimons donc le points ayant un cœfficient négatif. (g) Nous réduisons la sphère en déplaçant le centre vers le support réduit. (h) Le centre se déplace et le support comporte désormais trois points. La condition d'arrêt est atteinte avec le centre de la sphère qui appartient à l'enveloppe convexe du support.*

ANNEXE C

Méthode de double rendu

Lors de test de performance avec des silhouettes en basse résolution, $r < 256 * 256$ pixels, nous avons fait face à des résultats décevants. En effet, la pertinence du descripteur "MCC"(*Multiscale Closed Contour*), sur la base de test de *Princeton Shape Bnechmark*, se dégrade rapidement lorsque la résolution diminue, et ce de manière anormale.

Afin de comprendre ces mauvaises performances, nous avons étudié les différences présentes entre des silhouettes obtenues par projection avec $r = 256*256$ et $r = 64*64$ pixels. La principale observation a été l'apparition d'un grand nombre de composantes connexes. Celles-ci n'étant pas présentent sur le rendu avec $r = 256 * 256$ pixels ni sur l'objet 3D initial, nous en avons conclu que le problème venait de la technique permettant l'obtention des silhouettes par projection. Nous proposons donc un nouveau mode de rendu pour pallier à ce problème en basse résolution.

Le nouveau mode de prise de vue

Dans notre mode de capture original, nous utilisions un rendu en face pleine géré par *OpenGL* (voir figure C.1).

(a) *Résolution* $r = 256 * 256$ *pixels*

(b) *Résolution* $r = 64 * 64$ *pixels*

FIGURE C.1 – *La même silhouette obtenue en face pleine par projection avec une résolution de* $r = 256 * 256$ *(a) et de* $r = 64 * 64$ *pixels (b).*

Nous pouvons voir sur cet exemple l'apparition de nombreuses composantes connexes sur l'image 64x64 pixels. Ceci est dû au mode de rendu face pleine d'*OpenGL* qui n'affiche pas certaines d'entres elles à cause de leurs petites tailles. Pour pallier à ce problème, nous

allons donc utiliser le mode fil de fer d'*OpenGL*. Nous utilisons donc un double rendu : face pleine et fil de fer afin de pallier au problème des petites faces qui ne sont pas affichées. Nous obtenons le résultat visible sur la figure C.2

(a) *Résolution* $r = 256 * 256$ *pixels*

(b) *Résolution* $r = 64 * 64$ *pixels*

FIGURE C.2 – *La même silhouette obtenue en face pleine et fil de fer par projection avec une résolution de* $r = 256 * 256$ *(a) et de* $r = 64 * 64$ *pixels (b).*

Résultats sur la base de *Princeton Shape Benchmark*

Nous allons présenter ici les résultats obtenus avec l'ancienne et la nouvelle méthode de prise de vue. Afin de bien se rendre compte de l'importance de ce nouveau mode de capture, nous présentons pour chaque base de données le pourcentage de silhouettes comportant plusieurs composantes connexes ainsi que le nombre moyen de celle-ci et le "DCG" (voir l'annexe D) pour des images avec $r = 64 * 64$ et $r = 256 * 256$ pixels et ceci dans les deux modes de prises de vues présentés.

Résultats pour la base de données de test ("Princeton Test")

Les figures C.3, C.4 et C.5 présentent respectivement le pourcentage de silhouettes comportant plus d'une composante connexe, le nombre moyen de composantes connexes dans une silhouette et le "DCG" global.

Résultats pour la base de données d'entrainement ("Princeton Train")

Les figure sC.6, C.7 et C.8 présentent respectivement le pourcentage de silhouettes comportant plus d'une composante connexe, le nombre moyen de composantes connexes dans une silhouette et le "DCG" global.

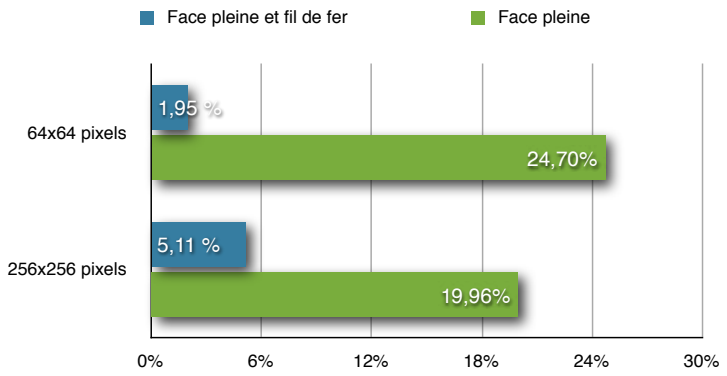


FIGURE C.3 – *Pourcentage de silhouettes ayant plusieurs compostantes connexes pour "Princeton Test".*

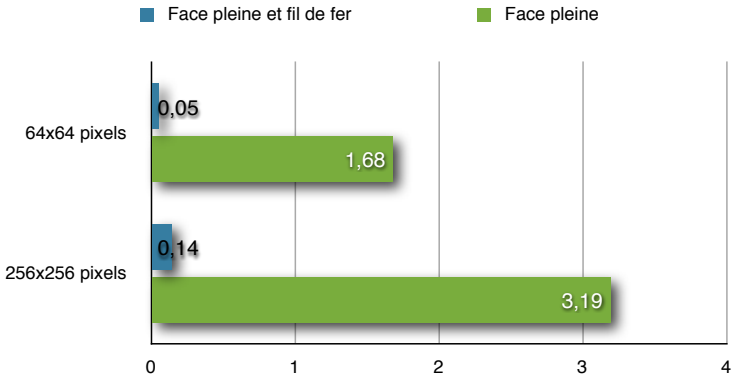


FIGURE C.4 – *Nombre moyen de compostantes connexes dans une silhouette pour "Princeton Test".*

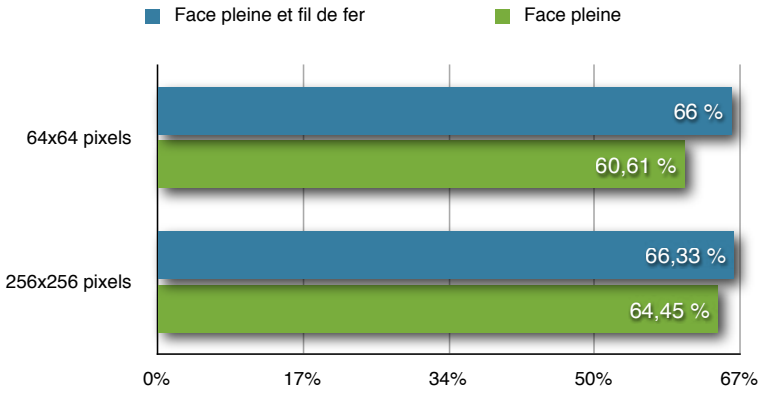


FIGURE C.5 – *DCG pour une requête exhaustive pour "Princeton Test".*

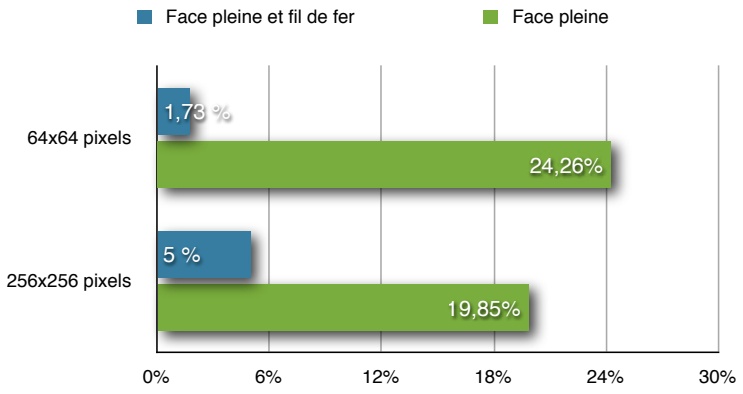


FIGURE C.6 – *Pourcentage de silhouettes ayant plusieurs compostantes connexes pour "Princeton Train".*

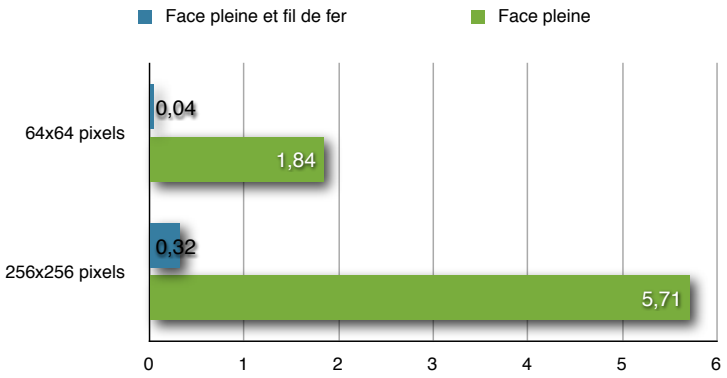


FIGURE C.7 – *Nombre moyen de compostantes connexes dans une silhouette pour "Princeton Train".*

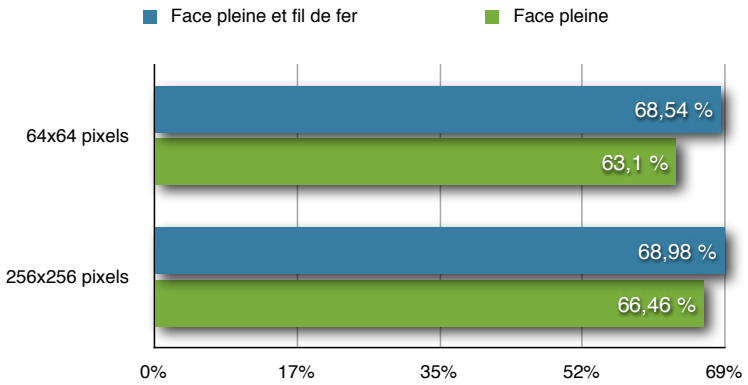


FIGURE C.8 – *DCG pour une requête exhaustive pour "Princeton Train".*

Conclusion

Nous voyons bien sur les bases de données d'objets de *Princeton Shape Benchmark*, que le nouveau mode de prise de vue combinant le mode face pleine et le mode fil de fer diminue le nombre de composantes connexes et augmente ainsi le "DCG" conformément à nos attentes. Ainsi, nous avons mis en place une technique de rendu palliant aux artefacts qui apparaissent à de basses résolutions.

ANNEXE D

Bases de données et estimation de la pertinence

Base de données d'objets 3D

Les bases d'objets 3D sont les briques essentielles pour évaluer un processus d'indexation et de recherche de modèles 3D. Ce qui différencie ces bases est le type d'objets qu'elles comportent ainsi que leurs nombres. Il en existe deux types :

- Les bases dites "généralistes" qui regroupent des objets de tous horizons : Web, musées, industries automobiles, jeux...
- Les bases dites "spécialisées" qui ne rassemblent que des modèles 3D d'un seul type (exemple : modèles de visages) ou d'un domaine donné (exemple : modèles de musée) ou encore répondant à une fonction (exemple : modèles articulés).

Nous pouvons finalement retrouver différentes bases constituées au fils des ans par des instituts de recherche ou par des industriels. Mais la plus grande de toutes reste la collection de modèles 3D stockée par *Google 3D Warehouse*[1] qui comporte quelques millions d'objets créés par les utilisateurs du logiciel *Google SketchUp*[2]. Cette base est interrogeable au moyen de mots-clés. Une liste, non exhaustive, de base d'objets 3D peut être trouvée dans le tableau D.1.

La création d'une nouvelle méthode d'indexation passe avant tout par la nature des objets qu'elle permet de traiter. En effet, il est clair que les moyens mis en œuvre pour reconnaître un visage diffèrent de ceux permettant de retrouver une pièce manufacturée. Dans ce sens, et afin de fournir à l'utilisateur des outils appropriés à son type de recherche, il est nécessaire de définir les classes d'objets pris en compte par la méthode d'indexation. Dans cette optique nous avons choisi d'orienter nos recherches sur les objets généralistes articulés ou non.

En particulier, nous utilisons trois bases pour évaluer notre processus d'indexation et de recherche. Il s'agit de deux bases de données du concours "SHREC" (voir section D) qui sont : *SHREC'07 Watertight* [80] et *SHREC'09 Generic* [83] d'une part, ainsi que de la base *Princeton Shape Benchmark* [110] d'autre part.

1. http://www.google.fr/3dwarehouse/
2. http://sketchup.google.com/intl/fr/

Base d'objets 3D	Nombre d'objets	Type d'objets
NTU Database [127]	10911	objets généralistes
Princeton Shape Benchmark Database [110]	1814	objets généralistes
SHREC'09 Generic Database [83]	800	objets généralistes
ITI Database [53]	544	objets généralistes
MPEG7 Database [131]	227	objets généralistes
SHREC'08 Watertight Database [81]	1500	objets articulés
SHREC'07 Watertight Database [80]	400	objets articulés
SHREC'09 Watertight Database [82]	200	objets articulés
York University Database [120]	5000	visages
SHREC'07 Face Database [79]	1515	visages
Renault Semantic Database [109]	5000	objets de CAO
PES Benchmark Database [100]	865	objets de CAO
Sculpteur Database [98]	720	objets de musée

TABLE D.1 – *Liste non exhaustive de bases de données 3D avec le nombre et le type d'objets.*

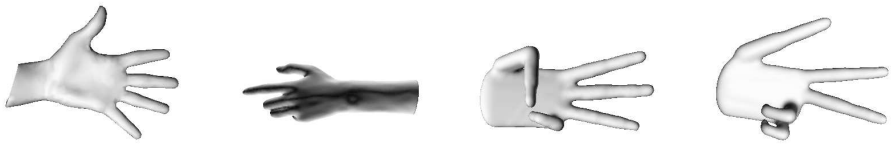
SHREC'07 Watertight

La base de données *SHREC'07 Watertight* provient du concours annuel "SHREC" et regroupe les objets "articulés" de l'édition 2007 (voir section D). Elle est composée de 400 modèles répartis équitablement dans 20 classes permettant ainsi une évaluation juste pour chaque classe. Chacune d'entre elles comporte, en général, un même objet déformé au niveau des articulations. Il existe cependant des classes non articulées, comme par exemple les "avions" et les "poissons".

À cause du caractère spécialisé de cette base, il est indispensable de mettre en place une méthode prenant en compte, lors de la description des objets, les déformations articulaires. Cette particularité est généralement obtenue par des approches par squelette ou par la mise en œuvre d'une élasticité dans la mesure de dissimilarité. La figure D.1 montre des exemples d'objets de cette base.

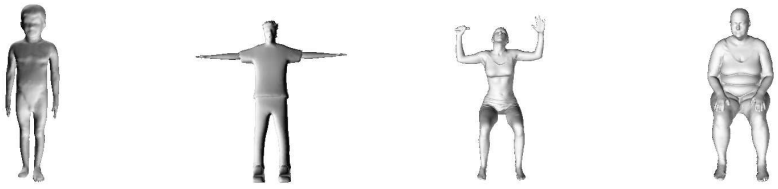
SHREC'09 Generic

La base *SHREC'09 Generic* provient, elle aussi, du concours annuel "SHREC" et regroupe, quant à elle, les objets de la section "générique" de l'édition 2009 (voir section D). Elle est constituée de 800 objets 3D répartis, de façon égale, dans 40 classes, ce qui permet comme pour la base précédente (voir paragraphe D), une évaluation cohérente sur l'ensemble des classes. Cette classification est assez proche de celle de la base *Princeton Shape Benchmark* (voir paragraphe D) organisée par rapport à la sémantique des objets plutôt que sur leur forme.



(a) *Classe "main" (articulée)*

(b) *Classe "lunette" (articulée)*

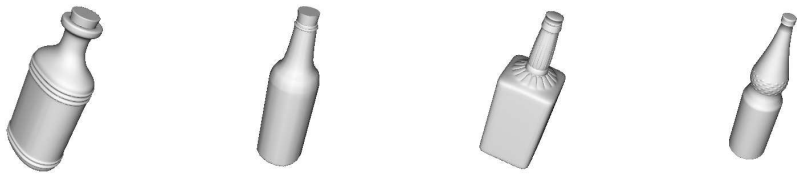


(c) *Classe "humain" (articulée)*

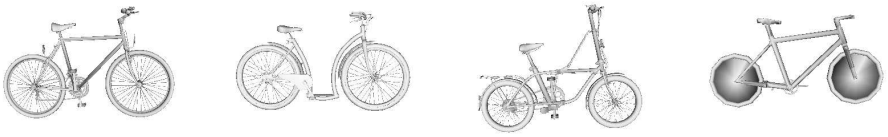
(d) *Classe "table" (non articulée)*

FIGURE D.1 – *Exemple de seize objets de quatre classes de la base Princeton Shape Benchmark SHREC'07 Watertight. (a), (b), (c) et (d) illustrent respectivement les classes "table", "vase", "buste" et "humain".*

Il est intéressant de noter que les classes choisies sont très fines. Par exemple, la classe "moto" est différente de la classe "vélo" tout comme les classes "avion monoplan" et "avion biplan". Finalement, grâce à la rigueur de sa constitution, cette base est idéale pour tester et évaluer un processus complet d'indexation et de recherche d'objets 3D. La figure D.2 montre un exemple d'objets provenant de celle-ci.

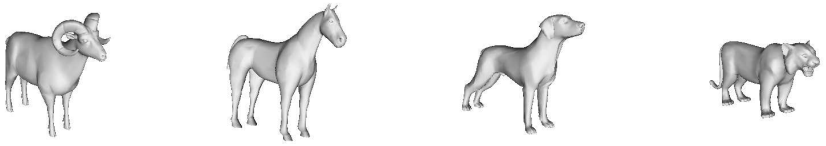


(a) *Classe "bouteille"*



(b) *Classe "bicyclette"*

(c) *Classe "instrument de musique"*



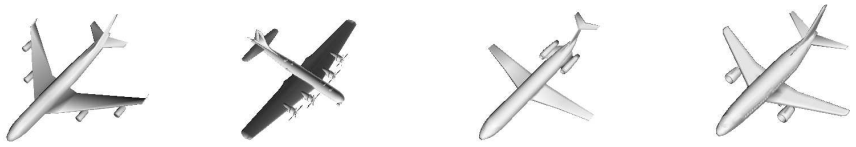
(d) *Classe "quadrupède"*

FIGURE D.2 – *Exemple de seize objets de quatre classes de la base Princeton Shape Benchmark SHREC'09 Generic. (a), (b), (c) et (d) illustrent respectivement les classes "avion", "chaise", "main" et "visage".*

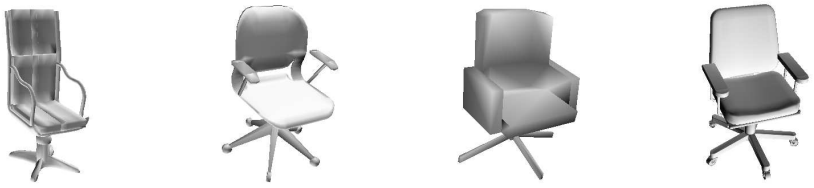
Princeton Shape Benchmark

La base de données *Princeton Shape Benchmark* est une base d'objets 3D mise à disposition par l'équipe du *Princeton Shape Retrieval and Analysis Group (PSRAG)*. Elle a été crée dans le but de permettre aux différentes équipes de recherche en indexation 3D, à travers le monde, de tester leurs approches sur une base commune. Elle est décomposée en deux parties, l'une pour l'entraînement : *Train Database* et l'autre pour l'évaluation : *Test Database*. Chacune d'entre elles comporte 907 objets 3D généralistes répartis, inégalement, dans 90 classes pour la première et 92 pour la seconde.

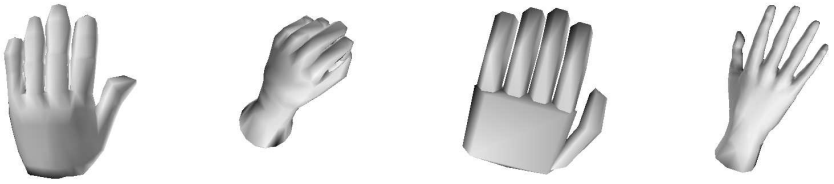
Il est intéressant de noter que ce regroupement en classes a été fait sur la sémantique des objets 3D plus que sur leur forme. En effet, il est possible de retrouver dans la classe "Escalier" des escaliers droits ou en colimaçons. Ce choix peut être handicapant lorsque l'on veut évaluer une méthode d'indexation par le contenu. Néanmoins, cette base offre une grande variété d'objets répartis dans un grand nombre de classes permettant une évaluation satisfaisante. D'autre part, sa grande notoriété dans le monde de la recherche permet une comparaison par rapport aux autres équipes du domaine. La figure D.3 montre des exemples d'objets qui la composent.



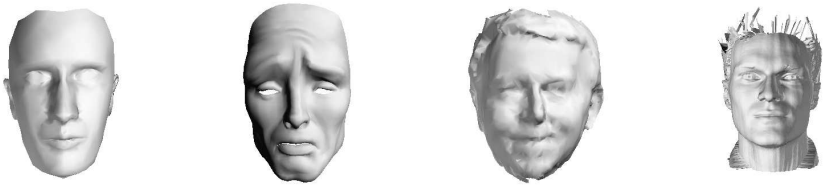
(a) *Classe "avion commerciaux"*



(b) *Classe "chaise de bureau"*



(c) *Classe "main"*



(d) *Classe "visage"*

FIGURE D.3 – *Exemple de seize objets de quatre classes de la base Princeton Shape Benchmark [110]. (a), (b), (c) et (d) illustrent respectivement les classes "avion", "chaise", "main" et "visage".*

Autres bases

D'autres bases ont été utilisées pour l'évaluation de nos résultats, en particulier lors du concours annuel "SHREC" (voir section D). Cependant, celles-ci n'étant que peu utilisées par le reste de la communauté scientifique, il ne nous a pas semblé nécessaire de les faire figurer dans ce manuscrit. Il s'agit de bases d'objets de type "CAO" et de bases de modèles articulés. Des évaluations de notre méthode sur ces bases peuvent être retrouvées dans les résultats analysés par Kalyanaraman et Ramani [58], Muthuganapathy et Ramani [73], Biasotti et Attene [12] et J.Hartveldt *et al.* [55].

Estimation de la pertinence

Afin d'estimer la pertinence des résultats qu'offre notre approche, deux moyens se sont offerts à nous. Le premier est l'évaluation des résultats grâce à des mesures statistiques et des courbes, présentées dans le paragraphe D, qui nous ont permis de valider numériquement la pertinence de la méthode proposée. Le second est la participation au cours des années 2007, 2008 et 2009 au concours "SHREC" (voir paragraphe D) qui nous a offert la chance de comparer nos résultats avec ceux d'autres équipes de recherches à travers le monde.

Mesures de la pertinence

Pour évaluer une méthode d'indexation et de recherche d'objets 3D, il est indispensable de se référer à la vérité terrain, définit par l'association "objet-classe", aussi appelée classification, des bases utilisées. Celle-ci n'est pas unique, car constituée par un humain qui estime de manière subjective la ressemblance entre deux modèles 3D, mais tente de se rapprocher d'une "vérité". Grâce à cette association, il devient possible d'évaluer les résultats obtenus par une approche.

Afin de noter une recherche utilisant un processus d'indexation, il est nécessaire de donner, pour chaque objet de la bibliothèque, une distance par rapport à la requête. Ainsi il possible d'organiser les objets dans un ordre décroissant de similarité, en fonction de la distance obtenue pour chacun. L'estimation de la pertinence des résultats a pour but de donner un score, généralement en pourcentage, de cette liste, et ceci, en considérant tous les objets de la base comme étant des requêtes possibles.

Dans ce manuscrit, nous utilisons quatre mesures statistiques et une analyse des performances, par courbe. Cette variété des moyens d'estimation permet de bien comprendre le comportement de notre approche pour des requêtes issues de classes diverses et variées. Afin de ne pas biaiser les résultats, la requête est toujours absente de la liste retournée par le processus de recherche.

Nearest Neighbor (NN)

C'est une mesure de type plus proche voisin. Elle permet de savoir si l'objet $\mathcal{O}_1$, situé en première position dans la liste de résultats, appartient à la même classe C que la requête. Le *nearest neighbor* d'une requête $\mathcal{O}_r$ est donné par :

$$NN(\mathcal{O}_r) = \begin{cases} 1 & si\ C(\mathcal{O}_1) = C(\mathcal{O}_r) \\ 0 & sinon \end{cases} \tag{D.1}$$

avec $C(\mathcal{O})$, la classe de l'objet $\mathcal{O}$. Le *nearest neighbor* pour un ensemble de n requêtes est donné par la moyenne des valeurs pour chaque requête :

$$NN = \frac{1}{n}\sum_{i=1}^{n} NN(\mathcal{O}_i) \tag{D.2}$$

First tier (FT)

C'est une mesure de type k plus proches voisins. Elle permet de connaître le pourcentage d'objets de la même classe que la requête, qui ont été retrouvés dans les k premières places. Ici, $k = |C(\mathcal{O}_r) - 1|$, c'est à dire la cardinalité de la classe de la requête. Le *first tier* d'une requête $\mathcal{O}_r$ est donné par :

$$FT(\mathcal{O}_r) = \frac{1}{k}\sum_{i=1}^{k} f(\mathcal{O}_i) \tag{D.3}$$

avec :

$$f(\mathcal{O}_i) = \begin{cases} 1 & si\ C(\mathcal{O}_i) = C(\mathcal{O}_r) \\ 0 & sinon \end{cases} \tag{D.4}$$

Le *first tier* pour n requêtes est donné par :

$$FT = \frac{1}{n}\sum_{i=1}^{n} FT(\mathcal{O}_i) \tag{D.5}$$

Cette mesure est très restrictive. En effet, pour obtenir un score de 100 %, il est nécessaire de retrouver l'intégralité des objets de la classe dans les k premières positions de la liste de résultats.

Second tier (ST)

Cette mesure est très proche du *first tier*, elle admet une marge de 50 % d'erreurs, en ne recherchant plus les objets de la classe dans les k premières positions de la liste de résultats, mais dans les $2k$ ($k = |C(\mathcal{O}_r)| - 1$). Ce qui donne, toujours avec la même fonction f définit dans l'équation D.4 :

$$ST(\mathcal{O}_r) = \frac{1}{k}\sum_{i=1}^{2k} f(\mathcal{O}_i) \tag{D.6}$$

Le *second tier* pour n requêtes est donné par :

$$ST = \frac{1}{n}\sum_{i=1}^{n} ST(\mathcal{O}_i) \tag{D.7}$$

C'est une mesure de pertinence qui permet d'évaluer très rapidement une méthode, et de la comparer par rapport à une autre, en admettant une certaine marge d'erreur dans les résultats. Cette mesure statistique est équivalente au *Bulls Eye Percentage Score* adopté par la norme MPEG7 (voir Zaharia et Preteux [143]).

Discounted Cumulative Gain (DCG)

C'est une mesure statistique qui permet de donner plus de poids aux objets de la même classe que la requête, qui apparaissent au début de la liste de résultats qu'à ceux qui sont à la fin. Plus les objets pertinents sont retrouvés tôt dans la liste, plus le score augmente (voir Kalervo et Jaana [57]). Ce qui donne, pour l'objet requête $\mathcal{O}_r$ et toujours avec la même fonction f définit dans l'équation D.4 :

$$DCG(\mathcal{O}_r) = \frac{DCG_n(\mathcal{O}_r)}{1 + \sum_{i=2}^{|C(\mathcal{O}_r)|-1} \frac{1}{log_2(i)}} \tag{D.8}$$

où n est le nombre d'objets dans la base et avec :

$$DCG_j(\mathcal{O}_r) = \begin{cases} f(\mathcal{O}_1) & si\ j = 1 \\ DCG_{j-1} + \frac{f(\mathcal{O}_j)}{log_2(j)} & sinon \end{cases} \tag{D.9}$$

Le *discounted cumulative gain* pour un ensemble de n requêtes est donné par :

$$ST = \frac{1}{n}\sum_{i=1}^{n} ST(\mathcal{O}_i) \tag{D.10}$$

Il est utile de noter que le calcul de cette mesure nécessite d'avoir la liste complète des résultats, et non plus les k ou $2k$ premiers objets comme pour le *first* et *second tier*.

Courbe Rappel/Précision

Pour une estimation plus fine des résultats, nous utilisons la notion de courbe *rappel/précision*. Elle se base sur deux critères. Le premier est le *rappel* : c'est le rapport du nombre d'objets pertinents (de la bonne classe) retrouvé au rang k sur le nombre d'objets pertinents disponible (la taille de la classe). Il s'agit donc de la proportion d'objets de la classe retrouvés au rang k. La seconde est la *précision* : c'est la proportion de documents pertinents retrouvés au rang k. Ce qui nous donne :

$$\begin{aligned} rappel_k(\mathcal{O}_r) &= \frac{1}{|C(\mathcal{O}_r)| - 1}\sum_{i=1}^{k} f(\mathcal{O}_i) \\ précision_k(\mathcal{O}_r) &= \frac{1}{k}\sum_{i=1}^{k} f(\mathcal{O}_i) \end{aligned} \tag{D.11}$$

On retrouve régulièrement ces deux notions dans la littérature car elles reflètent le point de vue de l'utilisateur : si la précision est faible, l'utilisateur sera insatisfait, car il devra perdre du temps à lire des informations qui ne l'intéressent pas. Si le rappel est faible, l'utilisateur n'aura pas accès à une information importante qu'il souhaitait avoir. Finalement, un processus d'indexation et de recherche parfait doit avoir une précision et un rappel de un, mais ces deux mesures sont souvent en compétition et une très forte précision ne peut être obtenue qu'au prix d'un rappel faible et vice-versa.

Pour construire la courbe, il suffit de calculer le *rappel* ainsi que la *précision* pour k allant de 1 à n, où n est le nombre d'objets de la base de test.

> **Remarque D.1.** *À cause de la formulation des courbes rappel/précision introduite précédemment, l'allure générale du graphique peut ne pas suivre la forme décroissante classique. Afin de contourner ce problème, on applique fréquemment la polarisation sur la courbe. Le traitement est le suivant : soit j et k deux points de rappel tels que $j < k$. Si au point j, la précision est inférieure à la précision au point k alors on augmente la précision de j à celle de k. Il est aussi possible, pour de meilleurs résultats, de faire une interpolation linéaire entre i et k, avec $i < j < k$, plutôt que de faire une simple augmentation.*

> **Remarque D.2.** *Il est intéressant de noter que les mesures nearest neighboor, first tier et Second tier représentent des points caractéristiques de la courbe rappel/précision. En effet, on a : $NN(\mathcal{O}_r) = précision_1(\mathcal{O}_r)$, $FT(\mathcal{O}_r) = rappel_{|C(\mathcal{O}_r)|-1}(\mathcal{O}_r)$ et $FT(\mathcal{O}_r) = rappel_{2*|C(\mathcal{O}_r)|-1}(\mathcal{O}_r)$.*

SHREC

Nous venons de présenter différentes mesures permettant d'évaluer numériquement et visuellement la méthode proposée dans cette thèse. Cependant, afin de comparer de manière juste nos travaux par rapport à d'autres approches de la littérature, nous avons participé au cours des années 2007, 2008 et 2009 au concours *3D SHape REtrieval Contest* "SHREC". Celui-ci est organisé, chaque année, dans le cadre du réseau d'excellence *AIM@-SHAPE*[3] afin d'évaluer les méthodes d'indexation et de recherches d'objets 3D sur différentes bases de données, généralistes ou non.

Ce concours se décompose en deux phases :

- La première est le développement, par chaque participant, d'une méthode fonctionnelle pour un type précis d'objets, dont il est possible de télécharger des exemples. En effet, il existe des bases d'objets génériques, articulés, de visages, de CAO... Cette étape dure environ un mois afin que chaque équipe ait le temps de tester sa méthode sur la base.

3. http://www.aimatshape.net/event/SHREC

– Dans un second temps, une série de requêtes est envoyée aux participants afin qu'ils déterminent les objets ressemblants à celle-ci. Évidemment, les équipes ne connaissent pas la vérité terrain de la base et de la requête. Dans cette phase, les participants doivent renvoyer la liste des objets classés dans un ordre décroissant de pertinence, et ceci, pour chaque requête, dans un temps imparti qui varie de un à sept jours.

Les résultats sont ensuite rendus publics et présentés lors d'une conférence ou d'un Workshop sur le sujet (voir en particulier les études comparatives de Veltkamp et Haar [130] et Jayantia *et al.* [54]). Dans les années passées, d'autres évaluations ont eu lieu, et en particulier celle faite par le *Princeton Shape Retrieval and Analysis Group* avec la base *Princeton Shape Benchmark*. Nous nous comparerons aussi avec ces résultats présentés par Shilane *et al.* [111].

Bibliographie

[1] Tomasz Adamek et Noel O'Connor : A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Transactions on Circuits and Systems for Video Technology : Special Issue on Audio and Video Analysis for Multimedia Interactive Services*, 14(5):742–753, 2004. 11, 94, 97, 101, 103, 128

[2] Tomasz Adamek, Noel O'Connor, Gareth Jones et Noel Murphy. : An integrated approach for object shape registration and modeling. *In SIGIR 2005 - 28th Annual International ACM SIGIR Conference, Workshop on Multimedia Information Retrieval*, 2005. 11, 94, 97, 101, 103

[3] Tomasz Adamek, Noel O'Connor et Noel Murphy. : Multi-scale representation and optimal matching of non-rigid shapes. *In CBMI 2005 - 4th International Workshop on Content-Based Multimedia Indexing*, 2005. 11, 94, 97, 101, 103

[4] Ceyhun Burak Akgul : *3d Shape Similarity Descriptors And Similarity Learning.* Thèse de doctorat, 2008. 59

[5] Ceyhun Burak Akgul, Bulent Sankur, Yucel Yemez et Francis Schmitt : Density-based 3d shape descriptors. *In EURASIP Journal on Applied Signal Processing, Special Issue on Search and Retrieval of 3d Content and Associated Knowledge Extraction and Propagation*, 2006. 59

[6] Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel et Thomas Seidl : 3d shape histograms for similarity search and classification in spatial databases. *In SSD '99 : Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pages 207–226, 1999. 40, 43, 46, 188

[7] Tarik Filali Ansary : *Model retrieval using 2d characteristic views.* Thèse de doctorat, 2006. 57, 188

[8] Tarik Filali Ansary, Jean-Philippe Vandeborre, Said Mahmoudi et Mohamed Daoudi : A bayesian framework for 3d models retrieval based on characteristic views. *In 3DPVT '04 : Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 139–146. IEEE Computer Society, 2004. 57

[9] Jurgen Assfalg, Alberto Del Bimbo et Pietro Pala : Retrieval of 3d objects using curvature maps and weighted walkthroughs. *In ICIAP*, pages 348–353. IEEE Computer Society, 2003. 51

[10] J. Barros, J. French, W. Martin, P. Kelly et M. Cannon : Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. *In SPIE*, volume 2670, pages 392–403, 1996. 197

[11] Richard Bellman : *Dynamic Programming*. Dover Publications, March 2003. 127

[12] Silvia Biasotti et Marco Attene : Shape retrieval contest 2008 : Stability of watertight models. *IEEE international conference on shape modeling and applications*, 0:217–218, 2008. 38, 190, 216

[13] Silvia Biasotti, Simone Marini, Michela Mortara, Giuseppe Patane, Michela Spagnuolo et Bianca Falcidieno : 3d shape matching through topological structures. *Lecture Notes in Computer Science*, pages 194–203, 2003. 50

[14] Silvia Biasotti, Simone Marini, Michela Spagnuolo et Bianca Falcidieno : Subpart correspondence by structural descriptors of 3d shapes. *Computer-Aided Design*, 38(9):1002–1019, 2006. 59

[15] Irving Biederman : Recognition-by-components : A theory of human image understanding. *In Psychological Review*, volume 94, pages 115–147, 1987. 70, 120

[16] Alberto Del Bimbo et Pietro Pala : Content-based retrieval of 3d models. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2:20–43, 2006. 37

[17] B. Bustos, D. A. Keim, D. Saupe, T. Schreck et D. V. Vranic : Automatic selection and combination of descriptors for effective 3d similarity search. *In IEEE International Workshop on Multimedia Content-based Analysis and Retrieval (MCBAR 2004)*, Miami, Florida, dec 2004. 58

[18] Benjamin Bustos, Daniel Keim et Tobias Schreck : A pivot-based index structure for combination of feature vectors. *In SAC '05 : Proceedings of the 2005 ACM symposium on Applied computing*, pages 1180–1184, 2005. 134

[19] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck et Dejan V. Vranic : An experimental comparison of feature-based 3d retrieval methods. pages 215–222. IEEE Computer Society, 2004. 37, 52

[20] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck et Dejan V. Vranic : Using entropy impurity for improved 3d object similarity search. pages 1303–1306. IEEE, 2004. 58

[21] Benjamin Bustos, Gonzalo Navarro et Edgar Chávez : Pivot selection techniques for proximity searching in metric spaces. *Pattern Recognition Letters*, 24(14):2357–2366, 2003. 134

[22] N. Canterakis : 3d zernike moments and zernike affine invariants for 3d image analysis and recognition. *In In 11th Scandinavian Conf. on Image Analysis*, pages 85–93, 1999. 44

[23] Hoel Le Capitaine : *Opérateurs d'agrégation pour la mesure de similarite - Application à l'ambiguïté en reconnaissance de formes.* Thèse de doctorat, Université de La Rochelle, 2009. 135, 136

[24] Mohamed Chaouch : *Recherche par le contenu d'objets 3D.* Thèse de doctorat, Telecom ParisTech, 2009. 33, 43, 54, 58, 106, 127, 160, 162, 163, 188, 195

[25] Mohamed Chaouch et Anne Verroust-Blondet : Enhanced 2d/3d approaches based on relevance index for 3d-shape retrieval. *In Shape Modeling International'06*, Matsushima, jun 2006. 54, 106

[26] Mohamed Chaouch et Anne Verroust-Blondet : Enhanced silhouette and depth-buffer based approaches for 3d-shape retrieval. *In SHREC2006, 3d Shape Retrieval Contest*, jun 2006. 54, 106

[27] Mohamed Chaouch et Anne Verroust-Blondet : Alignment of 3d models. *Graphical Models*, 71(2):63–76, March 2009. 33, 160, 162, 163

[28] Ding-Yun Chen : *Three-Dimensional Model Shape Description and Retrieval Based on LightField Descriptors.* Thèse de doctorat, 2003. 56, 92, 188

[29] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen et Ming Ouhyoung : On visual similarity based 3d model retrieval. *In EUROGRAPHICS*, Granada, Spain, sep 2003. 56, 92

[30] E. Copson : *Metric spaces.* Cambridge University Press, 1968. 131, 197

[31] P.R. Cromwell : *Polyhedra.* Cambridge University Press, 1999. 34

[32] M. Crucianu, M. Ferecatu et N. Boujemaa : Relevance feedback for image retrieval : a short survey. *In State of the Art in Audiovisual content-based retrieval, information universal access and interaction including data models and languages*, 2004. 134

[33] P. Daras et A. Axenopoulos : A 3d shape retrieval framework supporting multimodal queries. *International Journal of Computer Vision*, jul 2009. 58

[34] J.L. Dugelay, A. Baskurt et M. Daoudi : *3D object processing : Compression, indexing and watermarking.* John Wiley & Sons, Apr 2008. 37

[35] Tarik Filali-Ansary, Mohamed Daoudi et Jean-Philippe Vandeborre : A bayesian 3d search engine using adaptive views clustering. *In IEEE Transactions On Multimedia*, jan 2007. 57, 188

[36] Tarik Filali-Ansary, Jean-Philippe Vandeborre et Mohamed Daoudi : On 3d retrieval from photos. *In The 3rd IEEE International Symposium on 3D Data Processing , Visualization and Transmission (IEEE 3DPVT'06)*, 2006. 57

[37] Kaspar Fischer et Bernd Gartner : The smallest enclosing ball of balls : Combinatorial structure and algorithms. *In International Journal of Computational Geometry and Applications (IJCGA)*, volume 14, pages 341–378, 2004. 201

[38] Kaspar Fischer, Bernd Gartner et Martin Kutz : Fast smallest-enclosing-ball computation in high dimensions. *In ESA : Annual European Symposium on Algorithms*, 2003. 66

[39] Hongbo Fu, Daniel Cohen-Or, Gideon Dror et Alla Sheffer : Upright orientation of man-made objects. *In SIGGRAPH '08 : ACM SIGGRAPH 2008 papers*, pages 1–7, 2008. 32

[40] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz et David Dobkin : Modeling by example. *ACM Trans. Graph*, 23(3):652–663, 2004. 59

[41] Thomas A. Funkhouser, Patrick Min, Michael M. Kazhdan, Joyce Chen, Alex Halderman, David P. Dobkin et David P. Jacobs : A search engine for 3d model. *ACM Transactions on Graphics*, 22(1):83–105, jan 2003. 46

[42] Nikhil Gagvani et Deborah Silver : Parameter-controlled volume thinning. *CVGIP : Graph. Models Image Process.*, 3:149–164, 1999. 48

[43] Ran Gal et Daniel Cohen-Or : Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph*, 25(1):130–150, 2006. 59

[44] Bernd Gärtner : Fast and robust smallest enclosing balls. *In ESA '99 : Proceedings of the 7th Annual European Symposium on Algorithms*, pages 325–338, 1999. 66

[45] Afzal Godil, Helin Dutagaci, Ceyhun Burak Akgül, Apostolos Axenopoulos, Benjamin Bustos, Mohamed Chaouch, Petros Daras, Takahiko Furuya, Sebastian Kreft, Zhouhui Lian, Thibault Napoleon, Athanasios Mademlis, Ryutarou Ohbuchi, Paul L. Rosin, Bülent Sankur, Tobias Schreck, Xianfang Sun, Masaki Tezuka, Anne Verroust-Blondet, M. Walter et Yücel Yemez : Shrec'09 track : Generic shape retrieval. *In 3DOR*, pages 61–68, 2009. 38, 190

[46] J. Hartveldt, Michela Spagnuolo, Apostolos Axenopoulos, Silvia Biasotti, Petros Daras, Helin Dutagaci, Takahiko Furuya, Afzal Godil, Xiaolan Li, Athanasios Mademlis, Simone Marini, Thibault Napoleon, Ryutarou Ohbuchi et Masaki Tezuka : Shrec'09 track : Structural shape retrieval on watertight models. *In 3DOR*, pages 77–83, 2009. 38, 189

[47] Carlos Hernandez : *Stereo and Silhouette Fusion for 3D Object Modeling from Uncalibrated Images Under Circular Motion.* Thèse de doctorat, Telecom ParisTech, 2004. 20

[48] G. Hetzel, B. Leibe, P. Levi et B. Schiele : 3d object recognition from range images using local feature histograms. *In IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume 2, pages 394–399, 2001. 198

[49] Masaki Hilaga, Yoshihisa Shinagawa, Taku Komura et Tosiyasu L. Kunii : Topology matching for fully automatic similarity estimation of 3d shapes. *In SIGGRAPH*, pages 203–212, 2001. 49

[50] Berthold K. P. Horn : Obtaining shape from shading information. *In Shape From Shading*, pages 123–173, 1975. 20

[51] Berthold K. P. Horn : Extended gaussian images. jul 1983. 39, 46, 188

[52] Thomas Hurtut : *Analyse et recherche d'œuvres d'art 2D selon le contenu pictural.* Thèse de doctorat, Telecom ParisTech, 2008. 134

[53] Informatics et telematics institute : Iti database. http://3d-search.iti.gr/3DSearch. 212

[54] Subramaniam Jayantia, Yagnanarayanan Kalyanaramana, Natraj Iyera et Karthik Ramani : Developing an engineering shape benchmark for cad models. *Computer-aided design, Shape similarity detection and search for CAD/CAE applications*, 38:939–953, September 2006. 220

[55] J.Hartveldt, M.Spagnuolo, A.Axenopoulos, S.Biasotti, P.Daras, H.Dutagaci, T.Furuya, A.Godil, X.Li, A.Mademlis, S.Marini, T.Napoleon, R.Ohbuchi et M.Tezuka : Shrec'09 track : structural shape retrieval on watertight models. *30th International conference on EUROGRAPHICS 2009, workshop on 3D object retrieval, Munich, Germany*, March 2009. 190, 216

[56] Pu Jiantao, Liu Yi, Xin Guyu, Zha Hongbin, Liu Weibin et Yusuke Uehara : 3d model retrieval based on 2d slice similarity measurements. *In 3DPVT '04 : Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 95–101, 2004. 51

[57] Järvelin Kalervo et Kekäläinen Jaana : Ir evaluation methods for retrieving highly relevant documents. *In SIGIR '00 : Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48, 2000. 218

[58] Yagna Kalyanaraman et Karthik Ramani : Shape retrieval contest for cad models 2007 - evaluations. https://engineering.purdue.edu/PRECISE/shrec/results2007. 190, 216

[59] S. B. Kang et K. Ikeuchi : The complex egi : A new representation for 3-d pose determination. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(7):707–721, 1993. 40, 188

[60] Kazhdan, Chazelle, Dobkin, Funkhouser et Rusinkiewicz : A reflective symmetry descriptor for 3d models. *ALGRTHMICA : Algorithmica*, 38, nov 2003. 46, 47

[61] Michael Kazhdan, Thomas Funkhouser et Szymon Rusinkiewicz : Symmetry descriptors and 3d shape matching. *In SGP '04 : Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 115–123, 2004. 46

[62] Michael M. Kazhdan : *Shape Representations and Algorithms for 3d Model Retrieval*. Thèse de doctorat, 2004. 46

[63] Michael M. Kazhdan, Thomas A. Funkhouser et Szymon Rusinkiewicz : Rotation invariant spherical harmonic representation of 3d shape descriptors. *In SGP '03 : Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 156–165, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 45, 46, 188

[64] Jan J. Koenderink et Andrea J. van Doorn : Surface shape and curvature scales. *Image Vision Comput.*, 10:557–565, 1992. 40

[65] M. Kortgen, G.-J. Park, M. Novotni et R. Klein : 3d shape matching with 3d shape contexts. *In The 7th Central European Seminar on Computer Graphics*, apr 2003. 43

[66] Elwyn C. Leek : Effects of stimulus orientation on the identification of common polyoriented objects. *In Psychonomic bulletin and review*, volume 5, pages 650–658, 1998. 70, 71, 120

[67] Patrick Lemaire, éditeur. *Psychologie cognitive*. De Boeck Université, 2005. 10, 70

[68] D. G. Lowe : Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 2004. 54

[69] David Marr, éditeur. *Vision : A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York, NY, USA, 1982. 8, 70, 120

[70] Kevin McGuinness et Noel E. O'Connor : A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, February 2010. 10, 118

[71] Nimrod Megiddo : Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. *In SFCS '82 : Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 329–338, November 1982. 66

[72] Farzin Mokhtarian et Alan K. Mackworth : A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992. 97

[73] Ramanathan Muthuganapathy et Karthik Ramani : Shape retrieval contest 2008 : Cad models. *IEEE international conference on shape modeling and applications*, 0:221–222, 2008. 38, 190, 216

[74] C. Myers, L. Rabiner et A. Rosenberg : Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *Acoustics, Speech, and Signal Processing, IEEE Transactions on Signal Processing*, 28(6):623–635, 1980. 128

[75] Thibault Napoléon et Hichem Sahbi : Content-based 3d object retrieval using 2d views. *In IEEE International Conference on Image Processing*, pages 1437–1440, November 2009. 5, 113

[76] Thibault Napoléon et Hichem Sahbi : From 2d silhouettes to 3d object retrieval : Contributions and benchmarking. *EURASIP Journal on Image and Video Processing*, 2010, 2010. 5, 61

[77] Thibault Napoléon et Hichem Sahbi : Sketch-driven mental 3d object retrieval. *In Three-Dimensional Image Processing (3DIP) and Applications*, volume 7526, jan 2010. 5, 113

[78] S. B. Needleman et C. D. Wunsch : A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, March 1970. 127

[79] Center for geometry imaging Network of Excellence Project AIM@SHAPE, Utrecht university et virtual environments : Shrec'07 3d face models database. http://give-lab.cs.uu.nl/SHREC/shrec2007/. 212

[80] IMATI-CNR shape modeling group Network of excellence project AIM@SHAPE : Shrec'07 watertight models database, . http://watertight.ge.imati.cnr.it/. 211, 212

[81] IMATI-CNR shape modeling group Network of excellence project AIM@SHAPE : Shrec'08 stability on watertight models database, . http://shrec.ge.imati.cnr.it/shrec08_stability.html. 212

[82] IMATI-CNR shape modeling group Network of excellence project AIM@SHAPE : Shrec'09 structural shape retrieval database, . http://shrec.ge.imati.cnr.it/SHREC_2009_-_Structural_Shape_Retrieval/Home.html. 212

[83] national institute of standards Network of excellence project AIM@SHAPE et technology : Shrec'09 generic database. http://www.itl.nist.gov/iad/vug/sharp/benchmark/shrecGeneric/data.html. 211, 212

[84] Marcin Novotni et Reinhard Klein : 3d zernike descriptors for content based shape retrieval. *In The 8th ACM Symposium on Solid Modeling and Applications*, jun 2003. 44

[85] Marcin Novotni et Reinhard Klein : Shape retrieval using 3d zernike descriptors. *In Computer Aided Design 2004*, volume 36, pages 1047–1062, 2004. 44

[86] Ryutarou Ohbuchi, Takahiro Minamitani et Tsuyoshi Takein : Shape-similarity search of 3d models by using enhanced shape functions. *In Theory and Practice of Computer Graphics 2003 (TP.CG. 03)*, pages 97–104, Birmingham, UK, jun 2003. 39

[87] Ryutarou Ohbuchi, Masatoshi Nakazawa et Tsuyoshi Takei : Retrieving 3d shapes based on their appearance. *In* Nicu Sebe, Michael S. Lew et Chabane Djeraba, éditeurs : *MIR '03 : Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 39–45, New York, NY, USA, 2003. ACM. 53

[88] Ryutarou Ohbuchi, Kunio Osada, Takahiko Furuya et Tomohisa Banno : Salient local visual features for shape-based 3d model retrieval. *In Shape Modeling International*, pages 93–102, 2008. 54

[89] Ryutarou Ohbuchi, Tomo Otagiri, Masatoshi Ibato et Tsuyoshi Takei : Shape-similarity search of three-dimensional models using parameterized statistics. pages 265–275. IEEE Computer Society, 2002. 42

[90] R. Osada, T. Funkhouser, B. Chazelle et D. Dobkin : Matching 3d models with shape distributions. *In* Bob Werner, éditeur : *Proceedings of the International Conference on Shape Modeling and Applications (SMI-01)*, pages 154–166, Los Alamitos, CA, may 2001. IEEE Computer Society. 38, 188

[91] Robert Osada, Thomas Funkhouser, Bernard Chazelle et David Dobkin : Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002. 38

[92] Papadakis Papadakis, Ioannis Pratikakis, Theoharis Theoharis, Georgios Passalis et Stavros Perantonis : 3d object retrieval using an efficient and compact hybrid shape descriptor. *In Eurographics Workshop on 3D Object Retrieval*, pages 9–16, 2008. 58, 69

[93] Eric Paquet et Marc Rioux : Nefertiti : A query by content software for three-dimensional models databases management. pages 345–352. IEEE Computer Society, 1997. 38

[94] Eric Paquet et Marc Rioux : The mpeg-7 standard and the content-based management of three-dimensional data : A case study. pages 375–380, 1999. 43

[95] Eric Paquet, Marc Rioux, Anil Murching, Thumpudi Naveen et Ali Tabatabai : Description of shape information for 2-d and 3-d objects. *Signal Processing : Image Communication*, 16:103–122, 2000. 29

[96] Geogios Passalis, Theoharis Theoharis et Ioannis A. Kakadiaris. : Ptk : A novel depth buffer-based shape descriptor for three-dimensional object retrieval. *Visual Comput*, 23:5–14, 2007. 52

[97] Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz et Thomas A. Funkhouser : A planar-reflective symmetry transform for 3d shapes. *ACM Trans. Graph*, 25(3):549–559, 2006. 27, 32, 33, 47

[98] Sculpteur R&D project (semantic et cntent-based multimedia exploitation for European benefit) : Sculpteur database. http://www.sculpteurweb.org/. 212

[99] J.T Pu et K. Ramani : An approach to drawing-like view generation from 3d models. *In ASME IDETC/CIE 2005, 25th Computers and Information in Engineering (CIE) Conference*, Long Beach, California, USA, sep 2004. 194

[100] Center for information sciences in engineering Purdue university : Purdue engineering shape benchmark database. http://shapelab.ecn.purdue.edu/Benchmark.aspx. 212

[101] Philip T. Quinlan et Glyn W. Humphreys : Perceptual frames of reference and two-dimensional shape recognition : further examination of internal axes. *In Perception*, volume 22, pages 1343–1364, 1993. 70

[102] Julien Ricard : *Indexation et recherche d'objets 3d a partir de requetes 2d et 3d.* Thèse de doctorat, 2005. 48

[103] Julien Ricard, David Coeurjolly et Atilla Baskurt : Art extension for description, indexing and retrieval of 3d objects. pages 79–82, 2004. 48

[104] Julien Ricard, David Coeurjolly et Atilla Baskurt : Generalizations of angular radial transform for 2d and 3d shape retrieval. *Pattern Recognition Letters*, 26 (14):2174–2186, 2005. 48

[105] Y. Rubner, C. Tomasi et L. J. Guibas : A metric for distributions with applications to imge databases. *In IEEE International Conference on Computer Vision (ICCV'98)*, Bombay, India, 1998. 199

[106] Hichem Sahbi, Jean yves Audibert, Jaonary Rabarisoa et Renaud Keriven : Robust matching and recognition using context-dependent kernels. *In ICML '08 : Proceedings of the 25th international conference on Machine learning*, pages 856–863, 2008. 195

[107] Dietmar Saupe et Dejan V. Vranic : 3d model retrieval with spherical harmonics and moments. *In* Bernd Radig et Stefan Florczyk, éditeurs : *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, volume 2191 de *Lecture Notes in Computer Science*, pages 392–397. Springer, 2001. 44, 188

[108] Steven Seitz et Charles Dyer : Photorealistic scene reconstruction by voxel coloring. *In CVPR'97 : Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, page 1067, 1997. 20

[109] Indexation et Compression 3D) Semantic project (Service d'Echange et de MANIPULATION TATOUAGE : Renault semantic database. http://liris.cnrs.fr/semantic-3d/. 48, 212

[110] Princeton shape RETRIEVAL et analysis GROUP : Princeton shape benchmark database. http://shape.cs.princeton.edu/benchmark/. 41, 45, 48, 52, 54, 55, 57, 59, 211, 212, 215

[111] Philip SHILANE, Patrick MIN, Michael M. KAZHDAN et Thomas A. FUNKHOUSER : The princeton shape benchmark. *IEEE international conference on shape modeling and applications*, 0:167–178, 2004. 37, 38, 188, 220

[112] Yoshihisa SHINAGAWA, Tosiyasu L. KUNII et Yannick L. KERGOSIEN : Surface coding based on morse theory. *IEEE Comput. Graph. Appl.*, 5:66–78, 1991. 49

[113] T. F. SMITH et M. S. WATERMAN : Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, March 1981. 127

[114] Steve SULLIVAN et Jean PONCE : Automatic model construction, pose estimation and object recognition from photographs using triangular splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1091–1096, 1998. 20

[115] H. SUNDAR, D. SILVER, N. GAGVANI, S. DICKINSON et D. Silver Y : Skeleton based shape matching and retrieval. *In SMI '03 : Proceedings of the Shape Modeling International 2003*, pages 130–139, 2003. 48, 60

[116] James J. SYLVESTER : A question on the geometry of situation. *In Quarterly Journal of Pure and Applied Mathematics*, pages 1–79, 1857. 66

[117] Johan W. H. TANGELDER et Remco C. VELTKAMP : Polyhedral model retrieval using weighted point sets. *In International Journal of Image and Graphics*, volume 3, pages 209–229, 2002. 42

[118] J.W.H. TANGELDER et R.C. VELTKAMP : A survey of content based 3d shape retrieval methods. *Shape Modeling Applications, 2004. Proceedings*, pages 145–156, June 2004. 37

[119] N. A. THACKER, F. AHEARNE et P. I. ROCKETT : The bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 32(4):363–368, 1997. 198

[120] Department of computer science The university of YORK : York university database. http://www-users.cs.york.ac.uk/~nep/research/3Dface/tomh/3DFaceDatabase.html. 212

[121] THOMAS et Philip SHILANE : Partial matching of 3d shapes with priority-driven search. *In Symposium on Geometry Processing*, jun 2006. 59

[122] Julien TIERNY : *Reeb graph based 3D shape modeling and applications.* Thèse de doctorat, Université des Sciences et Techonologies de Lille, 2009. 49

[123] Julien TIERNY, Jean-Philippe VANDEBORRE et Mohamed DAOUDI : 3d mesh skeleton extraction using topological and geometrical analyses. *In 14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006, ACM/SIGGRAPH sponsored)*, pages 85–94, Taipei, Taiwan, oct 2006. 49

[124] Julien TIERNY, Jean-Philippe VANDEBORRE et Mohamed DAOUDI : Invariant high-level reeb graphs of 3d polygonal meshes. *In 3rd IEEE International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'06)*, Chapel Hill, North Carolina, USA, jun 2006. 49

[125] Tony TUNG : *Indexation 3d de bases de donnees d'objets par graphes de Reeb ameliores.* Thèse de doctorat, Telecom ParisTech, 2005. 50

[126] Tony TUNG et Francis SCHMITT : The augmented multiresolution reeb graph approach for content-based retrieval of 3d shapes. *International Journal of Shape Modeling*, 11(1):91–120, 2005. 50

[127] National Taiwan UNIVERSITY : National taiwan university database. `http://3d.csie.ntu.edu.tw/~dynamic/database/index.html`. 212

[128] N. VAJRAMUSHTI, I. A. KAKADIARIS, T. THEOHARIS et G. PAPAIOANNOU : Efficient 3d object retrieval using depth images. *MIR '04 : Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 189–196, 2004. 53

[129] Remco C. VELTKAMP et Frank B. HAAR : Shrec2007 : 3d shape retrieval contest. Rapport technique, Department of Information and Computing Sciences, Utrecht University, 2007. 38

[130] Remco C. VELTKAMP et Frank B. HAAR : Shape retrieval contest (shrec) 2008. *IEEE international conference on shape modeling and applications*, 0:215–216, 2008. 38, 220

[131] MPEG7 video GROUP : Mpeg7 database. 212

[132] D. V. VRANIC : An improvement of rotation invariant 3d shape descriptor based on functions on concentric spheres. *In IEEE International Conference on Image Processing (ICIP 2003)*, volume 3, pages 757–760, September 2003. 46, 188

[133] D. V. VRANIC et D. SAUPE : Description of 3d-shape using a complex function on the sphere. *In IEEE International Conference on Multimedia and Expo (ICME 2002)*, pages 177–180, Lausanne, Switzerland, aug 2002. 46

[134] Dejan V. VRANIC : *3d Model Retrieval.* Thèse de doctorat, Leipzig University, 2004. 27, 28, 29, 30, 31, 33, 44, 45, 52, 53, 76, 106, 188

[135] Dejan V. Vranic et D. Saupe : 3d shape descriptor based on 3d fourier transform. *In* K. Fazekas, éditeur : *EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001)*, pages 271–274, Budapest, Hungary, sep 2001. 29, 47, 76

[136] Dejan V. Vranic, D. Saupe et J. Richter : Tools for 3d-object retrieval : Karhunen-loeve transform and spherical harmonics. *In* J.-L. Dugelay et K. Rose, éditeurs : *IEEE 2001 Workshop Multimedia Signal Processing*, pages 293–298, Cannes, France, oct 2001. 29, 46, 76

[137] Dingwen Wang, Jiqi Zhang, Hau-San Wong et Yuanxiang Li : 3d model retrieval based on multi-shell extended gaussian image. *In VISUAL*, pages 426–437, 2007. 40

[138] Emo Welzl : Smallest enclosing disks (balls and ellipsoids). *In Results and new trends in computer science*, pages 359–370, 1991. 66, 201

[139] Ding yun Chen et Ming Ouhyoung : A 3d model alignment and retrieval system, oct 2002. 56, 92

[140] Titus Zaharia et Françoise Prêteux : 3d versus 2d/3d shape descriptors : A comparative study. *In in SPIE Conf. on Image Processing : Algorithms and Systems*, volume 1, 2004. 37

[141] Titus Zaharia et Francoise Preteux : Hough transform-based 3d mesh retrieval. *In SPIE Conference 4476 on Vision Geometry X*, pages 175–185, San Diego, CA, aug 2001. 47, 76, 77

[142] Titus Zaharia et Francoise Preteux : Indexation de maillages 3d par descripteurs de forme. *In Actes 13Ëme CongrËs Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA'2002)*, pages 48–57, Angers, France, jan 2002. 40

[143] Titus Zaharia et Francoise J. Preteux : 3d-shape-based retrieval within the mpeg-7 framework. *In Nonlinear Image Processing and Pattern Analysis XII*, volume 4304, pages 133–145, 2001. 218

[144] Titus Bogdan Zaharia : *Indexation de videos et de maillages 3d dans le contexte MPEG-7*. Thèse de doctorat, 2001. 40, 47

[145] F. Zernike : Diffraction theory of the cut procedure and its improved form, the phase contrast method. *Physica*, 1:689–704, 1934. 56

[146] D. S. Zhang et G. Lu : Shape-based image retrieval using generic fourier descriptor. *In Signal Processing : Image Communication*, volume 17, pages 825–848, November 2002. 53

Publications

Article de journal

T. Napoléon et H. Sahbi : From 2D Photography Silhouettes to 3D Object Retrieval : Contributions and Benchmarking, *EURASIP Journal on Image and Video Processing* - Juin 2010

Articles de conférences internationales

T. Napoléon et H. Sahbi : Sketch Driven Mental 3D Object Retrieval, *Electronic Imaging Conference, 3D Image Processing (3DIP) and Applications*, Janvier 2010

T. Napoléon et H. Sahbi : Content-based 3D object retrieval using 2D views, *IEEE International Conference on Image Processing* - Novembre 2010

A. Godil, H. Dutagaci, C. Akgul, A. Axenopoulos, B. Bustos, M. Chaouch, P. Daras, T. Furuya, S. Kreft, Z. Lian, T. Napol ?eon, A. Mademlis, R. Ohbuchi, P. L. Rosin, B. Sankur, T. Schreck, X. Sun, M. Tezuka, A. Verroust-Blondet, M. Walter, Y. Yemez : SHREC'09 Track : Generic Shape Retrieval, *Eurographics Workshop on 3D Object Retrieval, ACM SIGGRAPH* - Mars 2009

J. Hartveldt, M. Spagnuolo, A. Axenopoulos, S. Biasotti, P. Daras, H. Dutagaci, T. Furuya, A. Godil, X. Li, A. Mademlis, S. Marini, T. Napoleon, R. Ohbuchi, M. Tezuka : SHREC'09 Track : Structural Shape Retrieval on Watertight Models, *Eurographics Workshop on 3D Object Retrieval, ACM SIGGRAPH* - Mars 2009

T. Napoleon, T. Adamek, F. Schmitt, N.E. O'Connor : Multi-view 3D retrieval using multi- scale contour representation, *SMI 2008. IEEE International Conference on Shape Modeling and Applications* - Juin 2008

T. Napoleon, T. Adamek, F. Schmitt, N.E. O'Connor : Multi-view 3D retrieval using silhouette intersection and multi-scale contour representation, *SMI 2007. IEEE International Conference on Shape Modeling and Applications* - Juin 2007

Rapport technique

T. Napoléon, H. Sahbi : From 2D photography to 3D object retrieval : contributions and benchmarking, *Rapport interne Télécom ParisTech* - Février 2009