

SPIM

Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques

U N I V E R S I T É D E B O U R G O G N E

Appariement de formes, recherche par forme clef

■ BILAL MOKHTARI



SPIM

Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques

U N I V E R S I T É D E B O U R G O G N E

N° X | X | X

THÈSE présentée par

BILAL MOKHTARI

pour obtenir le

Grade de Docteur de
l'Université de Bourgogne
et de

l'Université de Biskra

Spécialité : **Informatique**

Appariement de formes, recherche par forme clef

Unité de Recherche :
Laboratoire Électronique, Informatique et Image

Composition du jury:

OLIVIER MONGA	Rapporteur	Professeur, IRD, Bondy, France
SAMY AIT-AOUDIA	Rapporteur	Professeur, ENSI, Alger, Algérie
SEBTI FOUFOU	Examineur	Professeur, univ. Bourgogne, France
MOHAMED CHAWKI BATOUCHE	Examineur	Professeur, univ. Constantine, Algérie
DOMINIQUE MICHELUCCI	Co-directeur	Professeur, univ. Bourgogne, France
KAMAL EDDINE MELKEMI	Co-directeur	MCF classe A, univ. Biskra, Algérie
MOHAMED CHAWKI BABAHENINI	Invité	MCF classe A, univ. Biskra, Algérie

SOMMAIRE

Introduction	1
1 Descripteurs de formes, appariement et recherche par forme clef	3
1.1 Introduction	3
1.2 Similarité d'objets et recherche par forme clef	3
1.2.1 Descripteurs de formes	4
1.2.1.1 Descripteurs géométriques	5
1.2.1.2 Descripteurs structurels et topologiques	8
1.2.1.3 Descripteurs par prise de vues	10
1.2.2 Dissimilarité des objets : calcul de distance	11
1.3 Distribution de forme et distances statistiques	13
1.3.1 Descripteurs de distribution de forme	13
1.3.2 La géométrie de diffusion et la décomposition spectrale	16
1.3.3 Distances entre distribution : distances statistiques	22
1.4 Conclusion	23
2 Quelques concepts utiles	25
2.1 Introduction	25
2.2 Représentation des objets	25
2.2.1 Les objets 2D	25
2.2.2 Les objets 3D	27
2.2.3 Modèles polygonaux : maillages	28
2.2.4 Représentation volumique discrète des objets 3D	29
2.3 Bases de données	30
2.4 Mesurer la performance du processus de recherche par forme clef	32
2.5 Conclusion	33
3 Décomposition d'objets en squelette d'ellipsoïdes	35
3.1 Introduction	35
3.2 Décomposition d'objets par l'algorithme des nuées dynamiques	35

3.2.1	Algorithme des nuées dynamiques	35
3.2.2	Partitionnement optimal	37
3.2.2.1	Un exemple 1D	39
3.2.2.2	Le nombre k optimal	39
3.3	Matrice de covariance et squelette d'ellipsoïdes	41
3.3.1	Matrice de covariance	41
3.3.2	Partitionnement final	42
3.3.3	Les k -partitions initiales	43
3.4	Classification d'une base de données	46
3.5	Appariement de formes	46
3.5.1	Vecteur caractéristique d'une classe	46
3.5.2	Calcul de distance : couplage optimal dans un graphe biparti	47
3.6	Résultats	47
3.7	Conclusion	49
4	Descripteurs et distribution de formes	51
4.1	Introduction	51
4.2	Notations	51
4.3	Descripteurs de formes	53
4.3.1	Descripteur de la géométrie de diffusion (GD)	53
4.3.2	Le diamètre local (LFD)	55
4.3.3	Le descripteur de l'énergie discrète (DE)	55
4.3.4	Descripteur de la normale en un sommet (VND)	56
4.3.5	La distribution locale de la forme (LSD)	56
4.3.6	Le descripteur de distribution de température (TD)	57
4.3.7	Descripteur de distribution (D2)	57
4.3.8	Le descripteur de l'enveloppe convexe (CH)	57
4.4	Mesure de dissimilarité : distance	58
4.5	Conclusion	59
I	Annexe	61
4.A	La décomposition spectrale des maillages 3D	63
4.A.1	Coordonnées différentielles et matrice Laplacienne	63
4.A.2	Matrice Laplacienne d'une marche aléatoire	64
4.A.3	Les valeurs et les vecteurs propres d'une matrice Laplacienne	64

4.B	Marche aléatoire et géométrie de diffusion	66
4.B.1	Chaîne de Markov	66
4.B.2	Chaîne de Markov d'un maillage 3D	68
5	La fusion des descripteurs par la règle du produit	71
5.1	Introduction	71
5.2	Descripteurs de formes utilisés et dissimilarité de formes	71
5.3	La fusion des descripteurs	72
5.3.1	Fusion précoce et règles de fusion	73
5.3.2	Le descripteur proposé : S	75
5.4	Résultats expérimentaux	76
5.4.1	Paramètres	77
5.4.2	Résultats	77
5.4.3	Résultats de comparaison	80
5.4.4	Pré-traitement et temps de calcul	82
5.5	Conclusion	83
6	Combinaison des mesures de dissimilarité	85
6.1	Introduction	85
6.2	Vue d'ensemble	86
6.2.1	Descripteurs de formes utilisés	86
6.2.2	Règles de fusion	87
6.3	Notre approche	87
6.3.1	Notations	87
6.3.2	Description de notre approche	88
6.4	Résultats	89
6.4.1	Descripteurs de formes et bases de données 3D utilisées	89
6.4.2	Résultats et comparaisons	92
6.5	Conclusion	95
7	Le clonage améliore la reconnaissance	97
7.1	Introduction	97
7.2	Contexte et principes	98
7.2.1	L'amélioration du processus de recherche par forme clef	98
7.2.2	L'utilisation du hasard, et du bruit	98
7.2.3	Les algorithmes génétiques	100

7.2.4	Principe de notre méthode	101
7.3	Clonage de formes et algorithme génétique	102
7.3.1	Initialisation de l'algorithme génétique : clonage de formes	102
7.3.2	Le calcul des paramètres ϵ et P	106
7.3.3	Notre algorithme génétique	107
7.3.3.1	Population de l'algorithme algorithme	107
7.3.3.2	Opérateurs de l'algorithme génétique	108
7.3.3.3	La qualité d'un clone : son aptitude	110
7.4	Algorithme génétique distribué pour le clonage de formes	113
7.4.1	Description de l'algorithme distribué proposé	113
7.4.2	La structure du MAS-SR-GA	115
7.5	Résultats expérimentaux	116
7.5.1	Résultats	116
7.5.2	Comparaison de l'algorithme génétique et du lissage	117
7.5.3	Comparaison des résultats	119
7.5.4	Pré-traitement et temps de calcul	120
7.6	Conclusion	121

INTRODUCTION

Cette thèse porte sur l'appariement de formes et la recherche par forme clef. Les formes sont décrites par des maillages triangulaires. A terme ces recherches s'appliqueront aux moteurs de recherche par forme clef sur internet.

En effet, la reconnaissance, l'appariement de formes, et la recherche par forme clef sont des tâches cruciales pour l'informatique graphique, la vision par ordinateur, les dépôts de formes, le contrôle de qualité, la robotique, etc. Actuellement, avec l'avancement des technologies qui produisent et manipulent des modèles 3D, la classification des modèles 3D est devenue cruciale dans différents domaines d'applications nécessitant d'identifier des modèles dans une large collection de modèles 3D.

Les maillages triangulaires sont les représentations des formes 3D les plus utilisées, car les plus simples et les plus commodes. Cependant, la structure des maillages n'a pas la régularité des images 2D ou 3D, ce qui complique l'analyse des formes. Lors de l'analyse de formes rigides représentées par des maillages triangulaires, il faut considérer des propriétés résistantes aux translations, aux rotations, et aux changements d'échelle. Pour des objets non rigides, il faut considérer des propriétés résistantes aux changements de posture, voire aussi aux changements topologiques : il faut reconnaître une personne qu'elle soit assise, debout, couchée, avec ou sans les mains dans les poches. Cette contrainte est essentielle pour garantir la performance de la reconnaissance automatique des formes. Plusieurs recherches ont été consacrées aux descripteurs discriminants de formes. Ils sont utilisés pour analyser, représenter, et ensuite reconnaître et distinguer les formes.

Cette thèse décrit plusieurs contributions au domaine de l'appariement de formes et la recherche par forme clef. La première contribution consiste en une extension de l'algorithme des nuées dynamiques, appelé aussi *k-means*, appliquée à la classification d'objets 3D. L'idée de cette méthode est de trouver pour un objet donné, représenté sous forme de nuage de points (pixels ou voxels) la meilleure décomposition en différentes parties ou classes ou ellipsoïdes. Le calcul de la similarité entre deux formes A et B, décomposées chacune en k ellipsoïdes, est équivalent au calcul d'un couplage optimal dans un graphe biparti avec $2k$ sommets. La méthode est simple, et donne de meilleurs résultats pour des objets rigides que la méthode originale.

La deuxième contribution est la fusion de deux descripteurs, l'un local, l'autre global, par la règle du produit. Pour caractériser une forme 3D, nous utilisons des caractéristiques locales basées sur les courbures pour définir un premier descripteur S_1 , et des caractéristiques globales basées sur la géométrie de la diffusion – par exemple la diffusion de la chaleur – pour définir un second descripteur, S_2 . Nous fusionnons ensuite ces deux descripteurs S_1 et S_2 en un seul, S , par la règle dite du produit. C'est théoriquement la fusion la plus simple et la plus efficace dans notre cas ; nous le confirmons expérimentalement en comparant les résultats en utilisant cette règle avec ceux obtenus en utilisant d'autres règles de fusion telles que : la somme, le minimum, le maximum, la moyenne.

La troisième contribution est basée sur la fusion des mesures de dissimilarité. Elle considère le graphe complet dont les sommets sont les formes de la base ou la requête. Les arêtes sont étiquetées par plusieurs distances, une par descripteur ; ensuite cette méthode calcule par programmation linéaire la combinaison convexe des distances qui maximise soit la somme des longueurs des plus courts chemins entre la requête et les objets de la base de données, soit la longueur du plus court chemin entre la requête et l'objet comparé à la requête.

La quatrième et la dernière contribution explore une nouvelle génération de méthodes utilisées dans la recherche par formes clés. Elle consiste à perturber la requête avec un algorithme génétique pour la rapprocher des formes de la base de données, pour un descripteur donné ; cette méthode est massivement parallèle, et une architecture multi-agent est proposée. L'algorithme génétique fait évoluer plusieurs générations de clones d'une requête donnée. Les clones sont visuellement identiques à la requête. Le meilleur clone est celui qui minimise une fonction distance (aptitude) ; nous considérons plusieurs distances. Nous comparons les différents résultats obtenus en utilisant les différentes fonctions d'aptitude.

Notons que les descripteurs de formes utilisés par nos contributions concernent un type particulier de descripteurs de formes, appelé *distribution de formes*. Ce type de descripteur considère les formes comme des dés généralisés. Par exemple, la surface frontière de chaque forme 3D est échantillonnée de façon uniforme par n points, et chaque point échantillon est étiqueté avec sa distance au centre de gravité de la forme. Pour chaque forme, ce vecteur de distances $d_i, i = 1, \dots, n$ est converti en un histogramme de $b = \sqrt{n}$ tiroirs. La distance, au sens de dissemblance, entre deux formes est définie comme la distance entre leurs histogrammes. L'appellation de distance est commode mais abusive, car l'inégalité triangulaire peut être violée. Donc, les descripteurs utilisés décrivent les objets 3D sous forme de distributions de valeurs calculées en chaque sommet, ou en chaque sommet échantillon, sur la surface de l'objet.

L'ensemble des approches réalisées dans le cadre de cette thèse montre une grande efficacité, et une grande compétitivité relativement aux autres méthodes existantes.

DESCRIPTEURS DE FORMES, APPARIEMENT ET RECHERCHE PAR FORME CLEF

1.1/ INTRODUCTION

La recherche par forme clef consiste à trouver dans une base de données (dans le cas idéal le web) les objets similaires à un objet donné, appelé *requête*, en tenant compte de l'apparence extérieure des objets.

Quelque soit la méthode utilisée, le processus de recherche est basé principalement sur deux concepts : un descripteur de forme, et une mesure de dissimilarité entre les formes. Un descripteur de forme calcule la signature de la forme, permettant de distinguer cette forme de toutes les autres. Cette description permet par la suite d'apparier les formes en utilisant une mesure de dissimilarité, en l'occurrence une distance. Par conséquent, les différents objets sont appariés avec la requête, et triés selon leur distance à la requête. Les objets les plus proches de la requête sont ceux avec une distance minimale.

Dans ce premier chapitre, nous donnons un aperçu général sur l'appariement de formes et la recherche par forme clef. Nous présentons donc les différentes notions de base, ainsi que les méthodes existantes utilisées dans le domaine de l'appariement de formes. Dans la Section 1.2, nous présentons le concept de l'appariement de formes, les descripteurs de formes, et le calcul de dissimilarité entre les formes : un état de l'art des descripteurs de forme existants, ainsi que les différentes catégories de descripteurs est présentée dans la Section 6.2.1 ; le principe pour mesurer la dissimilarité des formes est décrit dans la Section 4.4.

La Section 1.3 définit un cas particulier de descripteurs de formes, appelés descripteurs de distribution de forme. C'est le type de descripteurs utilisé par l'ensemble des contributions de cette thèse.

1.2/ SIMILARITÉ D'OBJETS ET RECHERCHE PAR FORME CLEF

La reconnaissance et la classification artificielle d'objets consistent à déterminer la nature d'un objet donné, et la catégorie d'objets à laquelle il est susceptible d'appartenir. Pour un objet donné, appelé *requête*, cela est réalisé en parcourant une collection d'objets

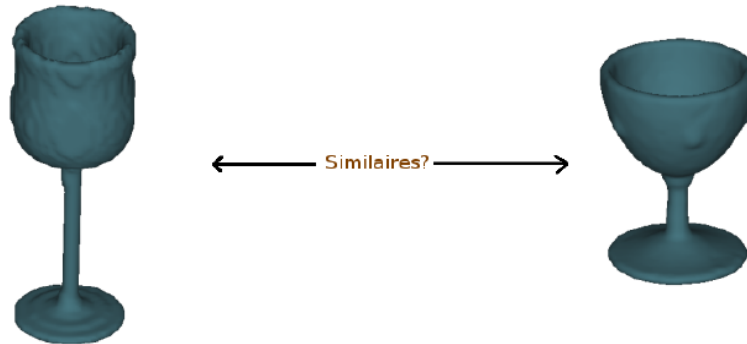


FIGURE 1.1 – Ces deux objets sont-ils similaires ?

(une base de données), dans le but d'identifier et désigner les objets les plus proches (similaires). Les objets sont comparés à la requête, et listés du plus proche (similaire) au moins proche (dissimilaire).

La comparaison de deux objets A et B consiste à mesurer à quel point ils sont similaires (Figure 1.1), en tenant compte de leur apparence. Une distance entre des caractéristiques mesurables associées à ces objets est calculée. Ces caractéristiques sont calculées en utilisant des descripteurs de formes. Les caractéristiques peuvent être des primitives simples telles que la surface de l'objet, son volume, etc.

Un descripteur de formes permet donc de caractériser et de décrire les objets en tenant compte de son apparence extérieure. C'est l'outil principal sur lequel s'appuient les méthodes d'appariement et de recherche par forme clef.

1.2.1/ DESCRIPTEURS DE FORMES

Afin de déterminer la ressemblance ou la similarité des objets, il est tout d'abord primordial de trouver une représentation simplifiée permettant de caractériser ces objets. Cette représentation de formes, appelé *descripteur de formes*, permet d'extraire des éléments caractéristiques à partir de l'objet. Ces caractéristiques sont facilement mesurables et utilisables pour calculer la similarité des objets. Elles peuvent être des primitives géométriques simple tels que le volume de l'objet, sa surface, les courbures aux différents points de la surface de l'objet, la distribution des distances entre les points de la surface, etc.

En effet, il est très facile pour n'importe quel être humain de savoir que les deux objets illustrés dans la Figure 1.1 sont similaires. Toutefois, ce n'est pas toujours évident de reproduire le même raisonnement par logiciel. Pour que le logiciel puisse accomplir cette tâche de la façon la plus efficace possible, il est primordial de considérer une description de formes qui prenne en considération toutes les représentations possibles d'un objet (changement de postures, transformations géométriques, topologiques, etc).

Dans la littérature, nous pouvons distinguer trois grandes familles de descripteurs de formes 3D : les descripteurs géométriques, les descripteurs structuraux et topologiques, et les descripteurs par prise de vues. Le schéma dans la Figure 1.2 résume les différents types de descripteurs de forme.

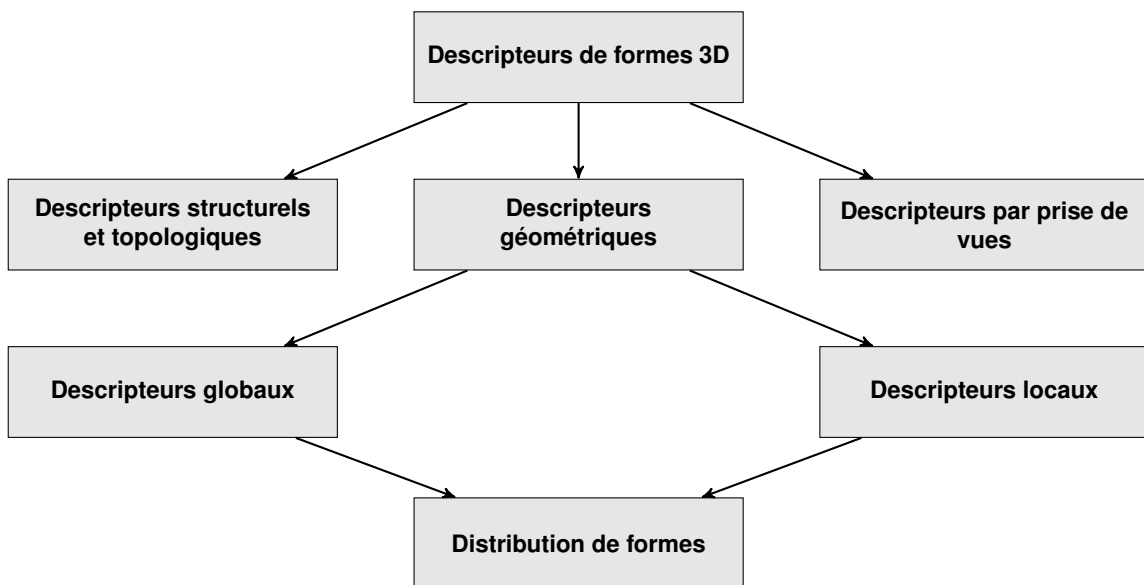


FIGURE 1.2 – Types de descripteurs de formes 3D.

1.2.1.1/ DESCRIPTEURS GÉOMÉTRIQUES

La première famille de descripteurs considère les caractéristiques géométriques de la forme. Ces caractéristiques peuvent être locales ou globales.

Les descripteurs géométriques globaux considèrent des propriétés qui prennent en compte la totalité de la forme. Dans la littérature, diverses méthodes d'appariement de formes basées sur des descripteurs globaux ont été présentées. Le descripteur de forme introduit dans [Belongie et al., 2000], appelé contexte de forme (en Anglais *shape context*), permet de décrire chaque point sur la surface d'un objet par rapport à tous les autres points, en utilisant les coordonnées log-polaires. Deux formes sont similaires si elles ont le même contexte de forme. Ce descripteur de forme a été initialement utilisé pour des objets 2D. Des généralisations du descripteur de contexte de forme pour les objets 3D ont été proposées par la suite [Körtgen et al., 2003, Kokkinos et al., 2012, Grundmann et al., 2008]. La Figure 1.3 montre des points (en violet) extraits à partir de la surface d'un objet 3D pour calculer son contexte de forme. Chaque point extrait est relié à tous les autres (segment bleus) afin de calculer la distribution en terme des coordonnées log-polaires.

Le contexte de forme est un descripteur de formes très simple et a connu un grand succès. Toutefois, le descripteur de contexte de forme dans sa version originale n'est pas adapté aux objets articulés. Cela a donné naissance à une variété de versions améliorées de ce descripteur, appliquées principalement à des objets 3D [Thayananthan et al., 2003, Grundmann et al., 2008, Zhang et al., 2003, Ling et al., 2007]. A titre indicatif, le descripteur présenté dans [Ling et al., 2007] utilise une distance géodésique¹ pour le calcul du contexte de forme. L'utilisation de la distance géodésique rend le descripteur résistant aux changements de postures des objets articulés ou flexibles.

1. la distance géodésique est la longueur du plus court chemin, soit à l'intérieur de la forme 2D, soit sur la surface de la forme 3D, entre deux points donnés.

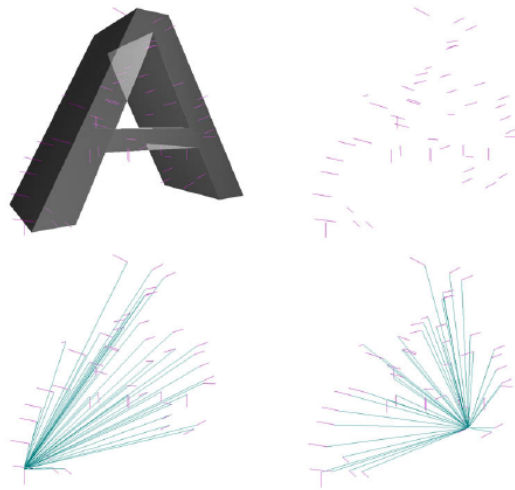


FIGURE 1.3 – Le contexte de forme pour un objet 3D [Körtgen et al., 2003].

Le descripteur de symétrie proposé dans [Kazhdan et al., 2002] permet de décrire les propriétés globales d'un objet 3D. Le descripteur permet de calculer la symétrie d'un objet volumique 3D (décrit par des voxels), en tenant en compte des plans passant par son centre de gravité. Le descripteur de symétrie est calculé en utilisant la transformée de Fourier. La figure 1.4 illustre des objets 3D ainsi que leur descripteur de symétrie.

Les moments invariants et les descripteurs de Fourier [Kim et al., 2000, Zhang et al., 2002, Folkers et al., 2002] font aussi partie des descripteurs globaux, permettant de décrire efficacement la surface de l'objet. D'autres descripteurs globaux ont été introduits dans [Berg et al., 2005, Lv et al., 2009, Bernier et al., 2003, Basri et al., 1998, Dao et al., 2006a].

Les descripteurs locaux mesurent des caractéristiques locales extraites à partir des régions ou des zones de la surface de l'objet [Laga et al., 2011, Buch et al., 2016, Taati et al., 2011]. Une région peut être le voisinage d'un sommet sur la surface d'un objet polygonal par exemple. Ce type de descripteurs de forme permet de décrire localement les objets. Cela s'avère très utile dans certaines situations, comme la reconnaissance des parties d'un objet [Gal et al., 2006, Funkhouser et al., 2006]. De plus, ces descripteurs sont résistants aux bruits.

Les descripteurs locaux utilisent les courbures [Douros et al., 2002, Zou et al., 2015], les points d'intérêt [Mikolajczyk et al., 2005], les vecteurs normaux [Thürner et al., 1998], etc. Les descripteurs de courbures [Koenderink et al., 1992, Meyer et al., 2003] mesurent par exemple la courbure moyenne, ou Gaussienne en un sommet sur la surface d'un modèle représenté sous forme d'un maillage ou d'un nuage de points. Douros et al. [Douros et al., 2002] calculent les courbures moyenne et Gaussienne pour chaque point sur la surface d'un objet représenté par un nuage de points. La surface est transformée ensuite en approximations locales calculées en chaque point en utilisant une représentation implicite de la surface $z = f(x, y)$. Le descripteur local décrit dans [Zaharia et al., 2002] utilise les courbures principales de la surface de l'objet pour définir l'indice de la forme (en Anglais *shape index*) [Koenderink et al., 1992]. L'indice de la

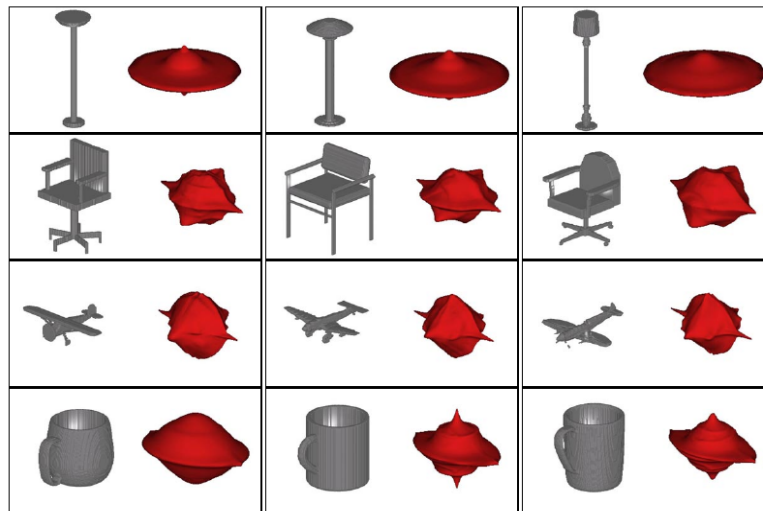


FIGURE 1.4 – Des exemples de modèles 3D avec leur descripteur de symétrie [Kazhdan et al., 2002].

forme pour un point p de la surface est donné par :

$$I(p) = \frac{1}{2} - \frac{1}{\pi} \arctan \frac{k_1 + k_2}{k_1 - k_2}, \quad \arctan(x) \in [0, \pi] \quad (1.1)$$

où k_1 et k_2 sont les courbures principales au point p . L'indice de la forme est un nombre positif qui donne une indication sur le type de la courbure (concavité, convexité, etc).

D'autres descripteurs locaux extraient les points d'intérêt à partir de la surface du modèle [Pratikakis et al., 2010, Knopp et al., 2010]. Il s'agit de déterminer les points porteurs d'informations parmi tous les points de la surface de l'objet. Il existe différentes techniques d'extraction des points d'intérêt telles que le détecteur de Harris [Harris et al., 1988], SIFT [Lowe, 1999], SURF [Funayama et al., 2012] etc. Pratikakis et al. [Pratikakis et al., 2010] présentent un descripteur de forme basé sur la détection des points d'intérêt en utilisant le détecteur de Harris. Leur méthode permet de définir proprement le voisinage d'un point sur la surface de l'objet, en tenant compte de la structure de la surface. Un exemple des points d'intérêt extraits en utilisant ce descripteur est illustré dans la Figure 1.5. Les descripteurs de points d'intérêt sont largement utilisés dans la reconnaissance des images 2D [Murala et al., 2012, Blum et al., 2012, Tang et al., 2009, Li et al., 2013].

Certains descripteurs [Lee et al., 2005, Song et al., 2014] mesurant la saillance des régions sur la surface du modèle font aussi partie des descripteurs locaux. Les descripteurs de saillance permettent de localiser les régions les plus importantes sur la surface de l'objet. Le descripteur de saillance présenté dans [Lee et al., 2005] mesure la saillance des régions d'un maillage 3D en utilisant la moyenne Gaussienne pondérée des courbures moyennes.

Une généralisation du descripteur de contexte de forme décrit précédemment a été proposée par Kokkos et al. [Kokkinos et al., 2012] pour décrire des surfaces courbes. Ce descripteur généralise à des surfaces 3D les coordonnées log-polaires utilisées initialement avec des images 2D. Le contexte de formes est donc défini dans le voisinage local de chaque point sur la surface de l'objet (Figure 1.7). Ce descripteur mesure les propriétés intrinsèques de l'objet 3D, et il est invariant par les transformations isométriques des

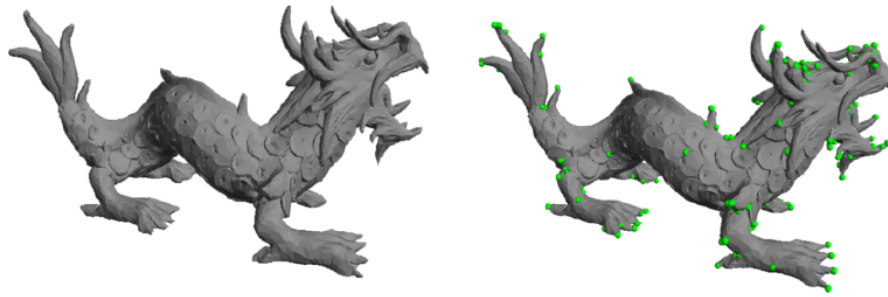


FIGURE 1.5 – Un exemple de descripteur de points d'intérêt [Pratikakis et al., 2010]. Les points en vert représentent les points d'intérêt.

objets articulés.

D'autres descripteurs locaux basés sur des primitives géométriques simples telles que la position d'un point par rapport au plan tangent, la longueur des courbures, etc, sont présentés dans la littérature. Une étude exhaustive de ces descripteurs est présentée dans [Heider et al., 2011].

1.2.1.2/ DESCRIPTEURS STRUCTURELS ET TOPOLOGIQUES

Ces descripteurs permettent de décrire un objet en terme de ses composantes (parties), en montrant comment ces dernières sont reliées entre elles. Ils utilisent des propriétés structurelles ou topologiques de l'objet. Cette catégorie de descripteurs comporte les méthodes de graphes [Fang et al., 2011, Torres et al., 2004], celles des squelettes [Chang et al., 2011, Chao et al., 2011], et les graphes de Reeb [Tung, 2005, Chen et al., 2002, Pascucci et al., 2007, Hilaga et al., 2001].

Ces descripteurs fournissent une représentation de forme très intuitive interprétée facilement. Dans certains cas, ils permettent de reconnaître plus efficacement les objets que d'autres types de descripteurs (géométriques par exemple). Par exemple, une table avec un plateau circulaire ou un plateau rectangulaire est facilement reconnaissable quand on considère des propriétés topologiques. Néanmoins le processus de calcul de similarité considérant des descripteurs structuraux et topologiques est difficilement mis en place.

Les descripteurs de formes basés sur la représentation par graphes sont principalement appliqués à des systèmes CAO. Habituellement, ils sont utilisés avec des représentations par des modèles de solides 3D, tels que les modèles B-Rep (*Boundary Representation*) et CSG (*Constructive solid geometry*). Ils sont difficilement utilisables avec des formes naturelles, comme les êtres vivants par exemple. El-Mehalawia et al. [El-Mehalawi et al., 2003] proposent une méthode d'appariement de modèles mécaniques basée sur un descripteur topologique. Le descripteur utilisé est basé sur la construction d'un graphe décrivant les différentes composantes d'une pièce mécanique. La dissimilarité entre les modèles est calculée par la mise en correspondance de leurs graphes associés.

Le descripteur MSG (*Model Signature Graphs*) proposé dans [McWherter et al., 2001] permet de transformer un modèle 3D de type B-Rep en un graphe décrivant des propriétés structurelles et sémantiques du modèle. Les sommets du graphe sont les facettes du modèle B-Rep ; les attributs d'un sommet sont représentés sous forme d'un vecteur ca-

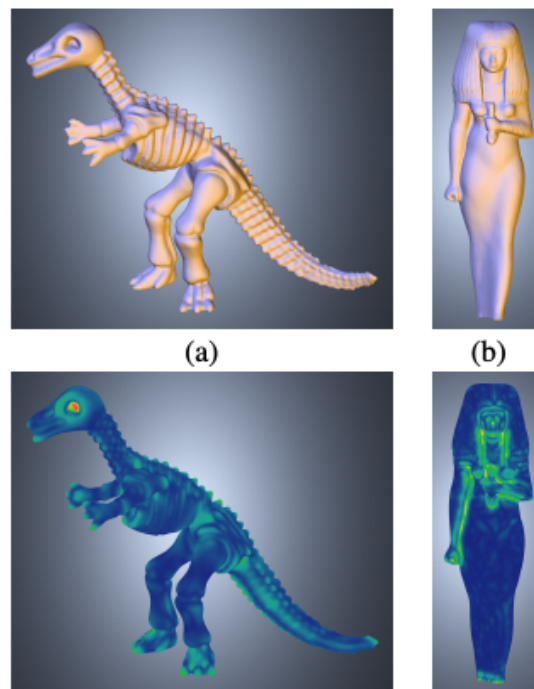


FIGURE 1.6 – La saillance de deux modèles 3D [Lee et al., 2005]. Les régions saillantes sont représentées avec des couleurs proches du rouge.

ractéristique contenant le type de la facette, sa taille, etc. Une arête dans le graphe existe si les deux facettes du modèle sont contiguës. Les attributs d'une arête peuvent encoder diverses informations telles que la concavité ou la convexité de la facette, la valeur de sa courbure, etc. La dissimilarité entre deux objets B-Rep représentés chacun par son graphe est calculée entre les vecteurs caractéristiques des sommets et des arêtes des graphes.

Les graphes de Reeb sont les descripteurs de formes structurels les plus utilisés. La représentation par graphe de Reeb permet de décrire la topologie des objets sous forme d'un squelette. Le squelette est construit en partitionnant l'objet en régions définissant la connectivité topologique de la surface d'un modèle. Le partitionnement est appliqué en utilisant une fonction donnée. Le sommet d'un graphe de Reeb représente une composante (région) du modèle, et les arêtes relient les composantes contiguës. L'exemple dans la Figure 1.8 montre un exemple d'un graphe de Reeb obtenu à partir d'un objet 3D, en considérant les valeurs d'une fonction calculée en chaque sommet sur la surface de l'objet. Typiquement, cette fonction est la coordonnée z du sommet.

La plupart des descripteurs de graphe de Reeb existants sont basés sur leur extension en graphe de Reeb multirésolution. L'idée de ce type de graphe de Reeb consiste à construire une série de graphes de Reeb à plusieurs niveaux de détails pour un objet donné (Figure 1.9). L'un des premiers descripteurs de graphe de Reeb multirésolution a été introduit par Hilaga et al. [Hilaga et al., 2001]. Cette méthode calcule la dissimilarité entre des objets 3D en tenant compte de la topologie des objets. Elle utilise des graphes de Reeb multirésolution en considérant la distance géodésique pour définir les graphes de Reeb à chaque niveau de détails. L'utilisation de la fonction basée sur la distance géodésique assure l'invariance aux articulations.

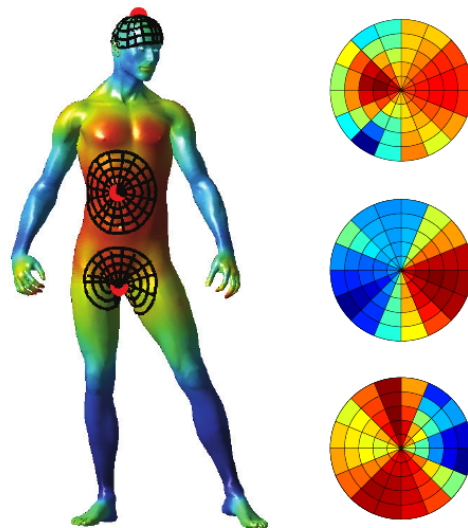


FIGURE 1.7 – Le contexte de forme d'un modèle 3D utilisant les coordonnées log-polaires dans le voisinage d'un point [Kokkinos et al., 2012].

Afin d'améliorer la précision de la méthode décrite précédemment, une extension de cette méthode a été proposée dans [Tung, 2005]. En effet, la méthode originale ne prend en compte que la topologie de la forme, ce qui est dans certains cas insuffisant pour décrire les objets efficacement. Dans la nouvelle méthode, TUNG et al. [Tung, 2005] considèrent à la fois la topologie du modèle, et la géométrie du graphe associé au modèle. Cela améliore considérablement la précision du processus de recherche.

La représentation par squelette est un autre type de descripteurs structurels (Figure 1.10). Un exemple de descripteurs de squelette est donné dans [Sundar et al., 2003]. Ce descripteur permet de transformer la géométrie et la topologie d'un modèle volumique 3D en squelette. La dissimilarité entre deux formes est calculée entre leurs squelettes associés.

Une autre méthode de squelette est présentée dans [Brennecke et al., 2004]. Afin de définir la dissimilarité des objets, la méthode identifie les parties communes dans leurs squelettes. D'autres descripteurs de squelettes sont présentés dans [Brennecke et al., 2004, Li et al., 2014, Dickinson et al., 2014].

1.2.1.3/ DESCRIPTEURS PAR PRISE DE VUES

Ces descripteurs sont calculés à partir de plusieurs points de vues de l'objet. La dissimilarité entre objets 3D est calculée entre des silhouettes 2D générées à partir des modèles 3D. Le descripteur LFD (*Light Field Descriptor*) présenté dans [Chen et al., 2003] est un des premiers descripteurs basés sur la prise de vues pour des modèles 3D. Les résultats expérimentaux ont démontré que deux modèles 3D sont similaires, s'ils sont identiques de tous les points de vues. Le descripteur LFD permet de générer différentes silhouettes 2D à partir d'un modèle 3D, par l'application d'une centaine de projections orthogonales de l'objet. Les silhouettes 2D générées sont calculées en utilisant une combinaison des moments de Zernike [Khotanzad et al., 1990] et la transformée de Fourier [Bracewell, 1965], décrivant respectivement l'intérieur et le contour de la silhouette.

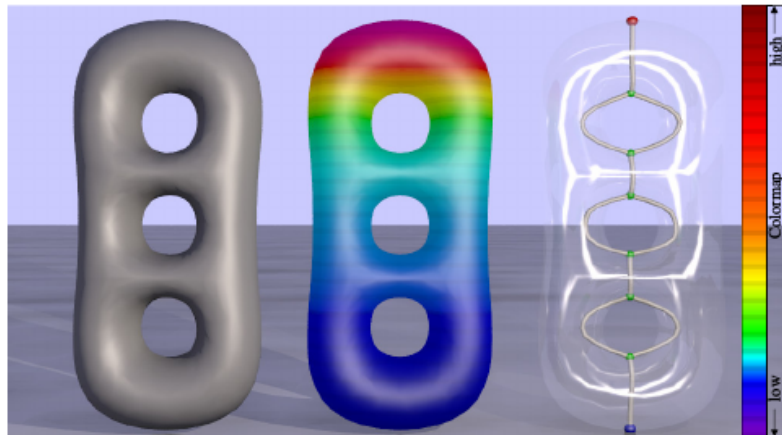


FIGURE 1.8 – Un modèle 3D et son graphe de Reeb associé [Pascucci et al., 2007].

L'avantage des descripteurs par prise de vues est qu'ils sont applicables à une grande variété de représentations de modèles 3D : maillages, voxels, nuages de points, B-Rep solides, etc.

La méthode proposée dans [Ohbuchi et al., 2008] permet d'extraire des points d'intérêt à partir des images 2D générées depuis un modèle 3D par des prises de vues sur la surface d'une sphère contenant le modèle 3D. Les points d'intérêt sont trouvés en utilisant l'algorithme SIFT [Lowe, 2004]. Les points caractéristiques sont ensuite répartis en différentes classes, représentées sous forme d'un histogramme. La distance entre deux formes est ensuite calculée en utilisant la divergence de *Kullback-Leibler* (Figure 1.11).

Ohbuchi et al. [Ohbuchi et al., 2010] proposent un descripteur de forme 3D basé sur l'utilisation des vues prises d'un modèle 3D. Cette méthode combine des caractéristiques locales et globales afin de calculer la similarité des objets. Ces caractéristiques sont extraites à partir d'un ensemble de prises de vues d'un modèle 3D.

1.2.2/ DISSIMILARITÉ DES OBJETS : CALCUL DE DISTANCE

Dans le processus de recherche par forme clef, le calcul de distance permet de mesurer l'éloignement d'un objet donné par rapport à la requête. Une petite distance signifie que les objets sont similaires. Les objets de la base de données sont triés par distance croissante à la requête.

Definition 1. *Étant donné un ensemble d'objets $O = \{o_1, o_2, \dots, o_n\}$, et S_i un descripteur de forme associé à la forme o_i , nous notons $d(S_i, S_j)$ la distance entre o_i et o_j .*

La distance d peut vérifier les propriétés suivantes :

- L'identité : $\forall o_i \in O, d(S_i, S_i) = 0$
- La symétrie : $\forall o_i, o_j \in O, d(S_i, S_j) = d(S_j, S_i)$
- La positivité : $\forall o_i, o_j \in O, o_i \neq o_j, d(S_i, S_j) > 0$
- l'inégalité triangulaire : $\forall o_i, o_j, o_k \in O, d(S_i, S_k) \leq d(S_i, S_j) + d(S_j, S_k)$

Une fonction de distance d peut être par exemple une fonction comme la distance euclidienne entre deux vecteurs signatures. Supposons que les deux descripteurs de formes S_1 et S_2 associés aux deux formes données o_1 un cube et o_2 une sphère sont deux vecteurs caractéristiques représentant leurs surfaces et leurs volumes. Nous notons A_1

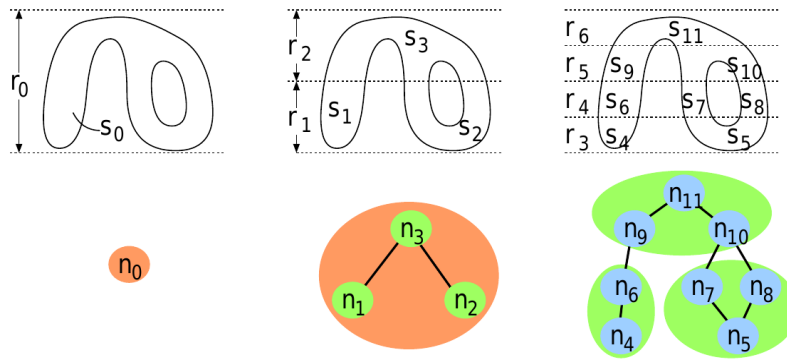


FIGURE 1.9 – Graphe de Reeb multirésolution à 1, 3, et 11 composantes [Hilaga et al., 2001].

et V_1 l'aire et le volume du cube, et A_2 et V_2 ceux de la sphère, donc $S_1 = (A_1, V_1)$ et $S_2 = (A_2, V_2)$. La distance entre les deux objets o_1 et o_2 peut être simplement la distance euclidienne entre les deux descripteurs (vecteurs) S_1 et S_2 . La distance entre les deux objets o_1 et o_2 est donnée par :

$$d(o_1, o_2) = d(S_1, S_2) = \sqrt{(A_1 - A_2)^2 - (V_1 - V_2)^2} \quad (1.2)$$

Notez qu'une étape de normalisation est requise avant de mesurer la distance entre deux objets. De cette façon, aucun attribut ne prend plus d'importance qu'un autre dans la mesure de distance.

Pour une requête Q , le processus de recherche calcule les distances $d(S, o_i)$, $i = 1, \dots, n$, et trie les objets par distance croissante.

En effet, un grand nombre de distances utilisées pour mesurer la dissimilarité entre les formes a été proposé. Chaque distance est complètement liée au descripteur de forme considéré. Dans la pratique, les descripteurs de forme peuvent être de nature plus complexe ; nous verrons plus loin dans les chapitres à venir différents descripteurs plus élaborés, ainsi que leurs distances induites. Nous présentons dans ce qui suit quelques fonctions distances.

Definition 2. Pour deux formes A et B données, nous notons $S_A = \{a_1, a_2, \dots, a_n\}$ et $S_B = \{b_1, b_2, \dots, b_n\}$ respectivement les vecteurs caractéristiques calculés en n sommets de chaque forme.

- **La distance euclidienne** : c'est la distance la plus simple et la plus utilisée. La distance entre deux formes A et B est donnée par :

$$d(S_A, S_B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

- **La distance de Manhattan** : cette distance est donnée par :

$$d(S_A, S_B) = \sum_{i=1}^n |a_i - b_i|$$

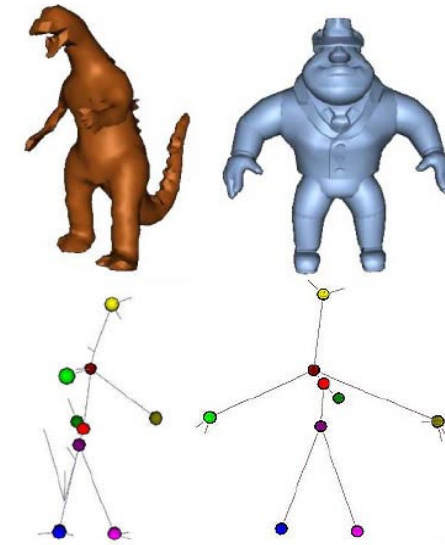


FIGURE 1.10 – Exemples de modèles 3D et leur squelettes associés [Sundar et al., 2003].

- **La distance de Minkowski** : c'est une généralisation de la distance euclidienne et de la distance de Manhattan. Elle est donnée par :

$$d(S_A, S_B) = \sqrt[p]{\sum_{i=1}^n |a_i - b_i|^p}$$

- **La distance de Hausdorff** :

$$d(S_A, S_B) = \max(\max_{a_i \in S_A} \min_{b_j \in S_B} \|a_i - b_j\|, \min_{a_i \in S_A} \max_{b_j \in S_B} \|a_i - b_j\|)$$

où $\|\cdot\|$ est la distance euclidienne.

1.3/ DISTRIBUTION DE FORME ET DISTANCES STATISTIQUES

1.3.1/ DESCRIPTEURS DE DISTRIBUTION DE FORME

Dans cette thèse, nous nous intéressons plus particulièrement aux descripteurs de forme appelés *distribution de forme*. Ces descripteurs permettent de décrire une forme via la distribution des valeurs d'une fonction tenant compte des caractéristiques géométriques décrivant la forme.

L'idée de base est de caractériser les formes par un dé généralisé, c-à-d une fonction de densité de probabilité, en pratique un histogramme. La surface de l'objet est échantillonnée soit de manière uniforme (le nombre d'échantillons dans une région est proportionnelle à sa superficie), soit en certains points saillants ou caractéristiques ; il faut ensuite calculer un descripteur en chaque point de l'échantillonnage.

Par exemple, le fameux descripteur D2 présenté par Osada et al. [Osada et al., 2001] échantillonne n points sur la surface de l'objet, et associe à chaque point une valeur

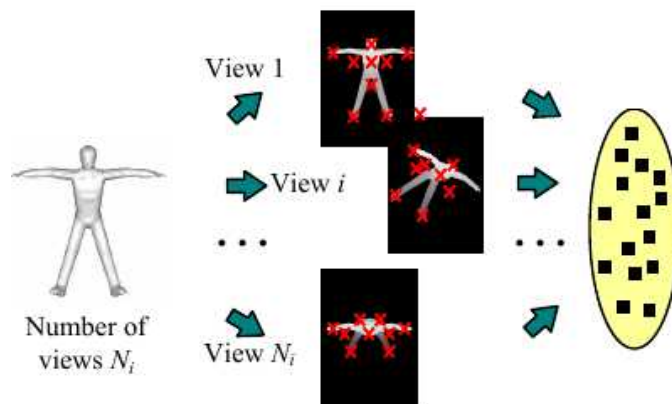


FIGURE 1.11 – Descripteur de points caractéristiques en utilisant une méthode basée sur la prise de vues pour un modèle 3D [Ohbuchi et al., 2008].

numérique en utilisant une mesure donnée. La mesure peut être la distance par rapport au centre de gravité de l'objet, l'angle formé entre trois points, etc. Ces valeurs sont ensuite transformées en une distribution ou histogramme. La distance entre deux objets est la distance entre leurs histogrammes associés (Figure 1.12).

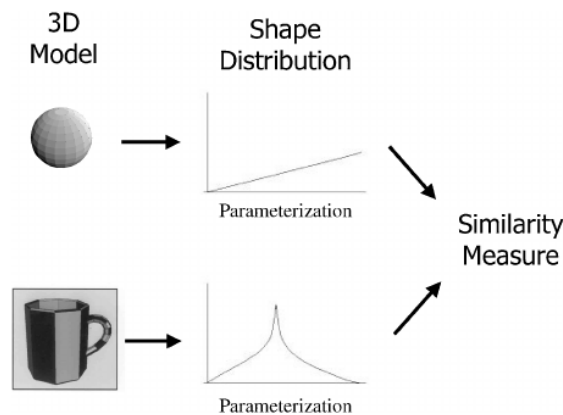


FIGURE 1.12 – Distribution de formes D2 [Osada et al., 2001].

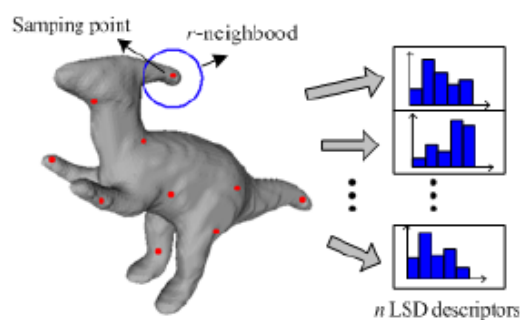


FIGURE 1.13 – Descripteur de diamètre local [Li et al., 2012].

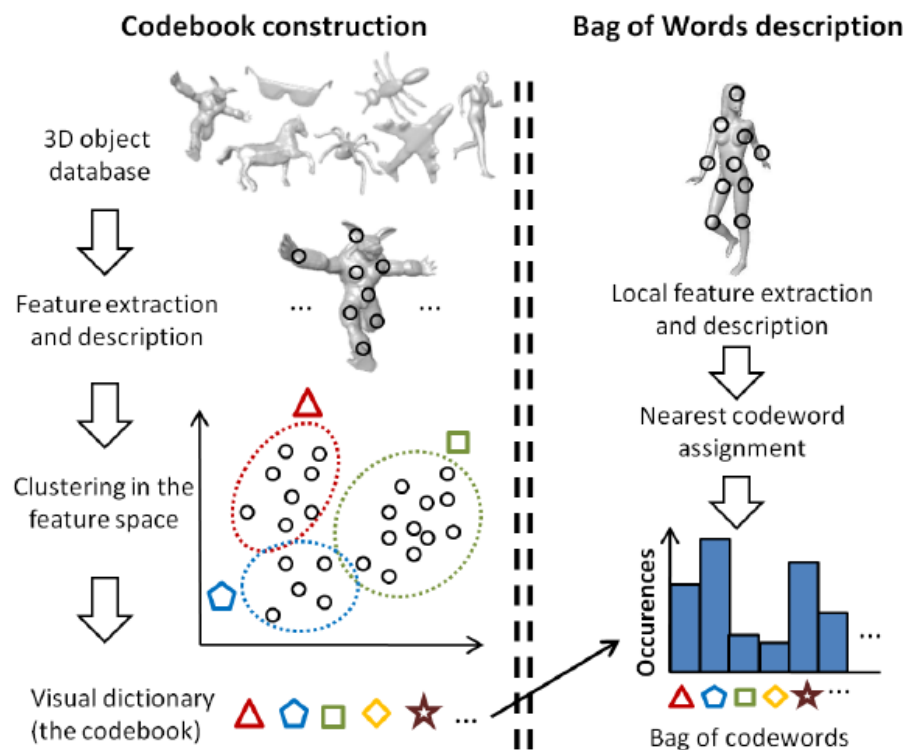


FIGURE 1.14 – Sac de mots visuels [Lavoué, 2011].

D'autres descripteurs considèrent le graphe extrait à partir du maillage triangulaire de la forme comme une chaîne de Markov. Pour cela, une certaine probabilité est associée à chaque arête, par exemple pour un arc $u \rightarrow v$, la probabilité peut dépendre de la longueur euclidienne, ou une certaine puissance p de la longueur euclidienne $\|uv\|$, c-à-d après la normalisation, la probabilité de l'arc de transition $u \rightarrow v$ est le rapport $\|uv\|^p / \sum_w \|uw\|^p$.

Le descripteur de distribution local LSD (*Local shape distribution*) [Li et al., 2012] extrait n points à partir de la surface de l'objet, et caractérise chaque point p par la distribution des distances entre le point p et l'ensemble des points dans son voisinage. Le voisinage du point p est représenté par une sphère de centre p (Figure 1.13). Les descripteurs de sac de mots visuels (*Bag of words*) [Lavoué, 2011, Bronstein et al., 2011, Lavoué, 2012] sont aussi des descripteurs de distribution utilisés pour décrire localement les objets. Il s'agit de construire un dictionnaire de mots visuels à partir des caractéristiques extraites à partir d'une large collection d'objets. Ces caractéristiques sont réparties en différentes classes, dont chacune est représentée par un mot visuel (triangle, carré, etc). Les différents mots visuels calculés pour un objet donné sont transformés en histogramme (Figure 1.14), selon le dictionnaire de mots construit au préalable.

Les descripteurs de distribution de formes sont largement utilisés pour leur efficacité et leur simplicité [Fang et al., 2011, Coifman et al., 2005, Mahmoudi et al., 2009, Osada et al., 2001].

1.3.2/ LA GÉOMÉTRIE DE DIFFUSION ET LA DÉCOMPOSITION SPECTRALE

Ces dernières années ont vu l'écllosion de la géométrie de la diffusion [Coifman et al., 2005]. C'est une branche de la théorie algébrique qui s'intéresse à l'étude des décompositions spectrales des graphe. Elle utilise une discrétisation de l'opérateur Laplacien [Sario et al., 1977] (*Laplace-Beltrami operator*). L'opérateur Laplacien permet de faire le lien entre une représentation discrète telle qu'un graphe, et une représentation continue telle qu'un signal, ou la propagation de chaleur le long de la surface.

Dans le domaine de l'appariement de formes, la théorie spectrale a été largement utilisée pour définir des descripteurs de distributions invariants aux transformations géométriques, et aux changements de postures. Elle utilise des propriétés locales définies dans le voisinage des sommets pour décrire la géométrie globale de la forme. La géométrie de diffusion fournit une grande variété de descripteurs basés sur des propriétés intrinsèques des formes ; ceux-ci ont une interprétation intuitive en termes de fréquences spatiales. Elle permet de réutiliser les concepts et les outils de l'analyse harmonique. Ce type de méthodes est utilisé dans d'autres domaines scientifiques. En fait, il remonte aux travaux de Joseph Fourier, qui créa l'analyse harmonique et la géométrie de diffusion de la chaleur [Coifman et al., 2005] pour poser et résoudre l'équation de la chaleur. Récemment, en mécanique quantique, J. Galvez [Galvez, 2012] a associé à chaque particule élémentaire un graphe et a trouvé une relation numérique, valable pour toutes les particules élémentaires, qui lie la masse, mesurée empiriquement, de chaque particule avec certains des moments spectraux de la matrice d'incidence (la matrice Laplacienne) du graphe associé à la particule (le n i-ème moment spectral est la somme des puissances n i-èmes des valeurs propres de la matrice, autrement dit la trace de la puissance n i-ème de la matrice).

Dans la géométrie 3D, la décomposition spectrale (géométrie de diffusion) des maillages 3D est largement utilisée pour la décomposition des maillages [Liu et al., 2007], la compression [Liu et al., 2004], et le paramétrage des surfaces [Tian et al., 2005]. Plus particulièrement, dans le domaine de la reconnaissance et de la classification de formes, la géométrie de diffusion est devenue un outil très important et souvent utilisé. Elle est liée au processus de diffusion (distribution) de la chaleur u sur la surface du modèle.

$$\left(\Delta_x + \frac{\delta}{\delta t}\right)u = 0 \quad (1.3)$$

où Δ_x représente l'opérateur Laplacien.

La solution de l'équation de la chaleur est connue sous le nom du noyau de chaleur. Ce concept a fait naître divers descripteurs de formes. La méthode proposée dans [Sun et al., 2009] utilise le noyau de chaleur pour la description de formes 3D (le descripteur est appelé *Heat Kernel Signature*). Ce descripteur est largement utilisé dans l'analyse de formes, pour des applications tels que la segmentation, la classification, et l'appariement de formes. Le descripteur HKS attribue à chaque point sur la surface du modèle un vecteur de caractéristiques, représentant la propriété de diffusion de chaleur le long de la forme. La signature d'un point est définie par :

$$p(x) = c(x)(K_{t_1}(x, x), \dots, K_{t_n}(x, x)) \quad (1.4)$$

où $c(x)$ est un paramètre choisi de telle sorte que $\|p(x)\|_2 = 1$, et $K_t(u, v)$ représente la solution de l'équation de la chaleur. La signature en un point est exprimée en terme

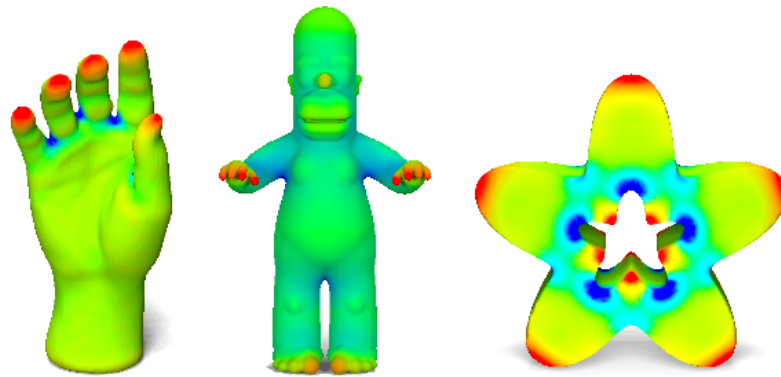


FIGURE 1.15 – Le descripteur de diffusion de chaleur calculé pour différents modèle [Sun et al., 2009]. Les valeurs du descripteur calculées pour les points sont représentées par une carte de couleurs graduée du bleu (petite valeur) au rouge (grande valeur).

de la dispersion d'une unité de chaleur appliquée en ce point, diffusant vers les autres points du modèle au fil du temps. Cette diffusion de chaleur permet aussi de donner une indication sur l'importance des régions sur la surface du modèle, puisque la chaleur se propage plus rapidement sur des régions denses, c-à-d des régions importantes.

Récemment, une nouvelle méthode basée sur la théorie spectrale pour l'analyse de la forme des os du carpe est présentée dans [Chaudhari et al., 2014]. La surface corticale de l'os est représentée en utilisant un système de coordonnées basé sur la décomposition spectrale en utilisant l'équation différentielle de Helmholtz.

L'une des applications possibles de la décomposition spectrale est le partitionnement spectral des graphes. Les algorithmes de partitionnement spectral (*spectral clustering*) ont eu beaucoup de succès, du fait qu'ils permettent d'améliorer la performance des algorithmes classiques, tel que nuées dynamiques à titre d'exemple. Un avantage de ce type de d'algorithmes, c'est qu'ils sont faciles à programmer, en utilisant des algorithmes standard de l'algèbre linéaire.

Boaz *et al.* [Nadler et al., 2005] présentent une méthode de partitionnement de données basée sur l'interprétation probabiliste des valeurs et des vecteurs propres du graphe Laplacien. Le principe de la méthode est d'utiliser la distance de diffusion [Coifman et al., 2005] pour le partitionnement des points en différents groupes (classes), de telle sorte que les distances de diffusions entre les points au sein du même groupe (classe) soient les plus petites possible, et ceux des points appartenant aux groupes différents, soient les plus grandes possible. Les auteurs démontrent aussi que l'utilisation du processus de diffusion permet de trouver une approximation discrète de l'opérateur de *Focher-Planck* [Risken, 1984], décrivant l'évolution de la densité des probabilités, en appliquant des forces aléatoires sur une particule en déplacement au fil du temps.

Une autre méthode d'apprentissage basée sur le partitionnement spectral du graphe est utilisée dans la segmentation de maillage 3D [Sharma et al., 2010]. La méthode utilise un algorithme d'apprentissage basé sur un modèle probabiliste utilisé pour superviser la segmentation d'un objet 3D. Les résultats de la segmentation seront par la suite utilisés pour apparier point par point des objets d'apprentissage segmentés avec d'autres non segmentés. La méthode utilise une version modifiée de la matrice Laplacienne du graphe représentant l'objet pour décomposer le maillage en utilisant l'algorithme des nuées dy-

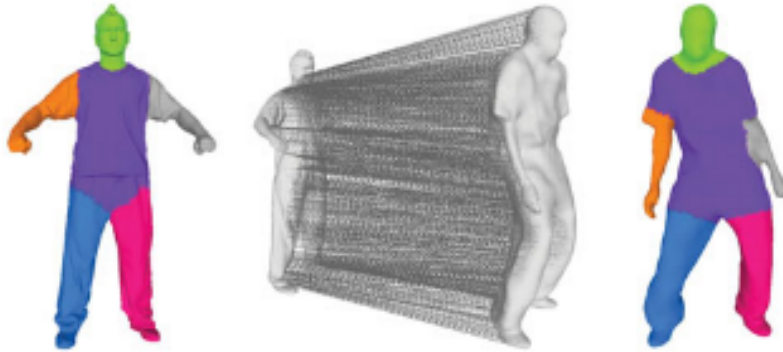


FIGURE 1.16 – Détection des symétries pour un objet articulé [Sharma et al., 2010].

namiques. La Figure 1.16 montre le résultat de segmentation et d'appariement d'objets en utilisant cette méthode.

Rong Liu et al. [Liu et al., 2007] utilisent aussi l'opérateur Laplacien pour la segmentation de maillages 3D (figure 1.17). La méthode est basée sur un processus récursif où à chaque étape une partie de l'objet est projetée sur un plan tangent afin d'extraire les contours de celle-ci. Une combinaison de l'opérateur Laplacien et d'un opérateur géométrique est mise en place afin de calculer les projections.

La décomposition spectrale a été aussi utilisée pour la détection de symétries dans un objet 3D [Ovsjanikov et al., 2008]. L'approche développée dans [Ovsjanikov et al., 2008] permet de déterminer efficacement les symétries d'une forme indépendamment des changements de postures, et des transformations isométriques (figure 1.18). La méthode consiste à détecter la symétrie sur la surface d'un objet, en transformant sa symétrie intrinsèque en symétrie euclidienne, en utilisant les valeurs et vecteurs propres de l'opérateur Laplacien.

La décomposition spectrale a été largement utilisée comme un outil de représentation et de description de formes. Rustamov [Rustamov, 2007] présente une représentation de la surface, résistante aux déformations, en utilisant la décomposition en valeurs et vecteurs propres de la matrice Laplacienne du graphe représentant l'objet. Cette représentation est aussi robuste vis à vis des changements locaux de la topologie des objets. Une forme est décrite en terme des distributions de signatures (G2-distributions) de tous les sommets, c-à-d la distribution des distances entre toutes les paires de sommets sur la surface du modèle, en considérant le descripteur GPS (*Global Point Signature*) [Rustamov, 2007]. Le descripteur GPS en un sommet v est un vecteur de valeurs donné par :

$$GPS(v) = \left(\frac{1}{\sqrt{\lambda_1}} \phi_1^{(v)}, \frac{1}{\sqrt{\lambda_2}} \phi_2^{(v)}, \dots, \frac{1}{\sqrt{\lambda_k}} \phi_k^{(v)} \right) \quad (1.5)$$

où λ_i et ϕ_i sont les valeurs et les vecteurs propres de la matrice Laplacienne du graphe représentant le modèle.

Une méthode de reconnaissance d'objets 3D représentés sous formes de nuages de points a été introduite par Mona et al. [Mahmoudi et al., 2009]. La méthode est basée sur la comparaison des distributions intrinsèques des nuages de points, où chaque nuage de points est caractérisé par la distance de diffusion entre toutes les paires de points du nuage. La signature d'un point est donnée par la moyenne des distances de diffusion

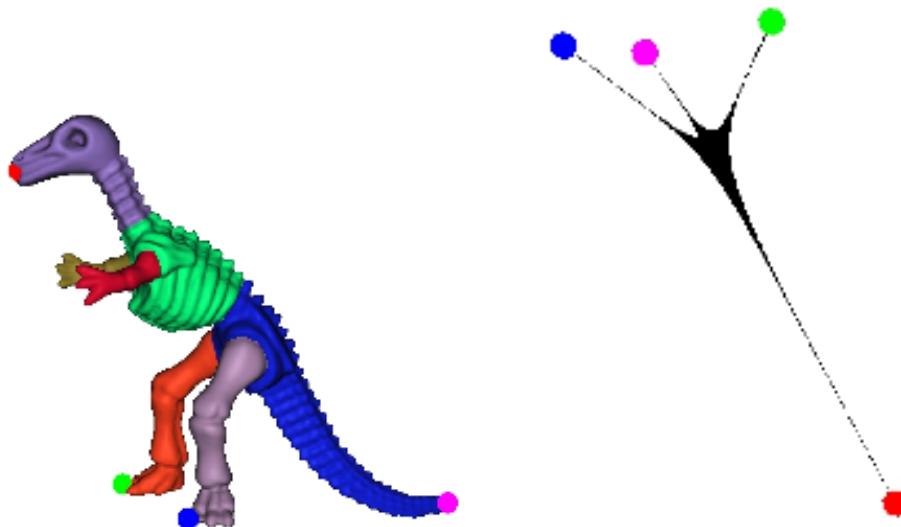


FIGURE 1.17 – Segmentation d'un modèle 3D en utilisant la méthode proposée dans [Liu et al., 2007]. A gauche : le modèle segmenté. A droite : la représentation spectrale du modèle.

entre celui ci, et tous les autres points du nuage. La distance de diffusion entre deux points u et v est donnée par :

$$d(u, v) = \sum_{i=0}^K \lambda_i^2 \phi_i(u) \phi_i(v) \quad (1.6)$$

où λ_i et ϕ_i sont les valeurs et les vecteurs propres de la matrice Laplacienne du graphe, dont le poids des arêtes est calculé en utilisant le noyau Gaussien $w(u, v) = \frac{\|u-v\|^2}{\sigma}$, où σ est la profondeur du voisinage.

Varun et al. [Jain et al., 2007] proposent une méthode d'amélioration de la reconnaissance d'objets articulés. Cette méthode utilise les vecteurs propres de la matrice d'affinité associés aux graphes représentant des objets 3D pour redéfinir les descripteurs SHD [Kazhdan et al., 2003] et LFD [Chen et al., 2003]. Les tests montrent que les résultats d'appariements sont nettement améliorés.

Mathieu et al. proposent une méthode de caractérisation de formes 3D non-rigides WKS (*Wave Kernel Signature*) [Aubry et al., 2011]. Le descripteur WKS (Figure 1.19) représente la moyenne des probabilités calculées en chaque sommet, en mesurant l'évolution d'une particule quantique sur la surface du modèle. Cette particule est représentée via une fonction $\psi(x, t)$ mesurant l'amplitude des ondes sur la surface. Cette évolution représente la solution de l'équation de Schrödinger :

$$\frac{\partial \psi}{\partial t}(x, t) = i \Delta \psi(x, t) \quad (1.7)$$

L'évolution temporelle des particules de la mécanique quantique peut paraître similaire à l'équation de la chaleur : la constante i remplace le coefficient de conductivité thermique qui apparaît dans l'équation de la chaleur. Cependant, cette évolution est un ensemble d'oscillations (sans amortissement) et non pas une diffusion. Contrairement à l'évolution

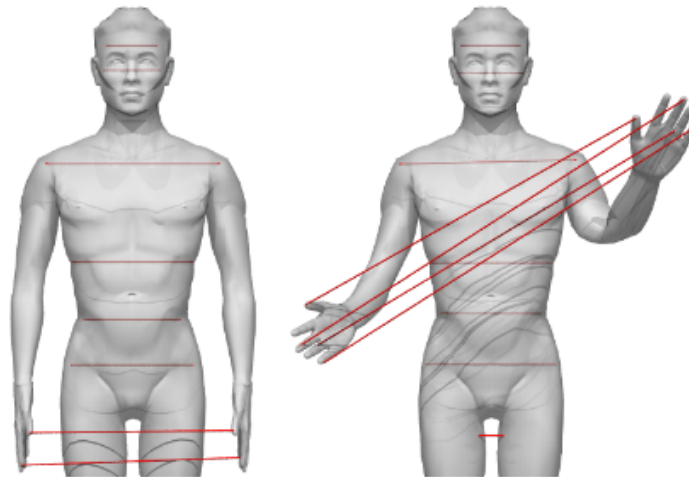


FIGURE 1.18 – Détection des symétries pour un objet avec changement de posture [Ovsjanikov et al., 2008].

des particules de la mécanique quantique, la diffusion de la chaleur est un processus physique irréversible : l'entropie augmente ; l'information est perdue car la chaleur est uniformément répartie à la fin du processus, quelle que soit sa distribution initiale.

L'évolution d'une particule (énergie E) tout au long de la surface est donnée par :

$$\psi_E(x, t) = \sum_{k=0}^{\infty} iE_k t \phi_k(x) f_E(E_k) \quad (1.8)$$

où $|\psi_E(x, t)|^2$ représente la probabilité d'être en un sommet x , et f_E^2 est la probabilité de distribution. La signature d'un sommet est donnée en terme de probabilités moyennes :

$$WKS(E, x) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T |\psi_E(x, t)|^2 \quad (1.9)$$

Plusieurs algorithmes basés sur le noyau de chaleur ont été aussi proposés et utilisés pour la représentation et l'appariement de formes.

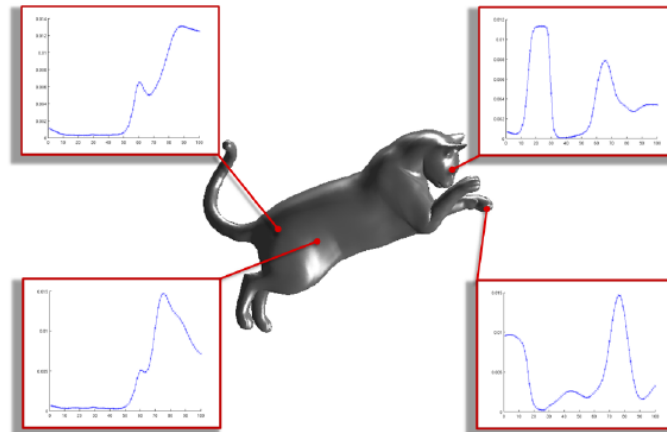


FIGURE 1.19 – Histogrammes des sommets calculés en utilisant le descripteur WKS [Aubry et al., 2011].

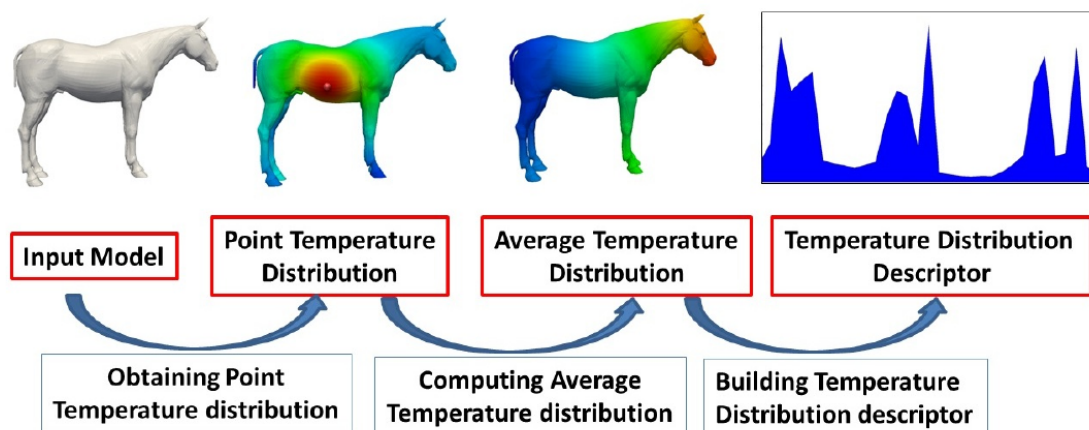


FIGURE 1.20 – Le descripteur de diffusion de chaleur calculé pour différents modèles [Sun et al., 2009]. Les valeurs du descripteur calculées pour les points sont représentées par une carte de couleurs graduée du bleu (petite valeur) au rouge (grande valeur).

La méthode proposée dans [Ovsjanikov et al., 2010] (Figure 1.21) définit la correspondance entre deux points en terme de la capacité de préserver la chaleur entre ces derniers. Cette méthode est aussi utilisée pour déterminer les symétries des objets.

Fang *et al.* développent un descripteur de formes basé sur la distribution de température TD (*Temperature Distribution*). Le descripteur mesure la propagation sur la surface du modèle d'une unité de chaleur appliquée en un point donné. La distribution de températures est calculée pour tous les points ; ensuite la signature de chacun de ces points est donnée par la moyenne des distributions de températures. Ces distributions sont ensuite exprimées sous forme d'histogrammes (figure 1.20). La distribution de température pour un objet est calculée en utilisant la décomposition en valeurs et vecteurs propres de la matrice Laplacienne du graphe sous-jacent au maillage triangulaire représentant l'objet.

Le noyau de chaleur a été aussi combiné avec une méthode de partitionnement PBC (*Persistence-based clustering* [Chazal et al., 2013]) pour la segmentation des maillages 3D [Skraba et al., 2010]. La méthode de partitionnement PBC identifie les parties comme étant des bassins d'attractions ; néanmoins, ce type d'algorithme est souvent instable. La combinaison de l'algorithme de partitionnement PBC et du noyau de chaleur [Sun et al., 2009] permet d'obtenir des résultats plus stables, et résistants aux déformations isométriques (Figure 1.22).

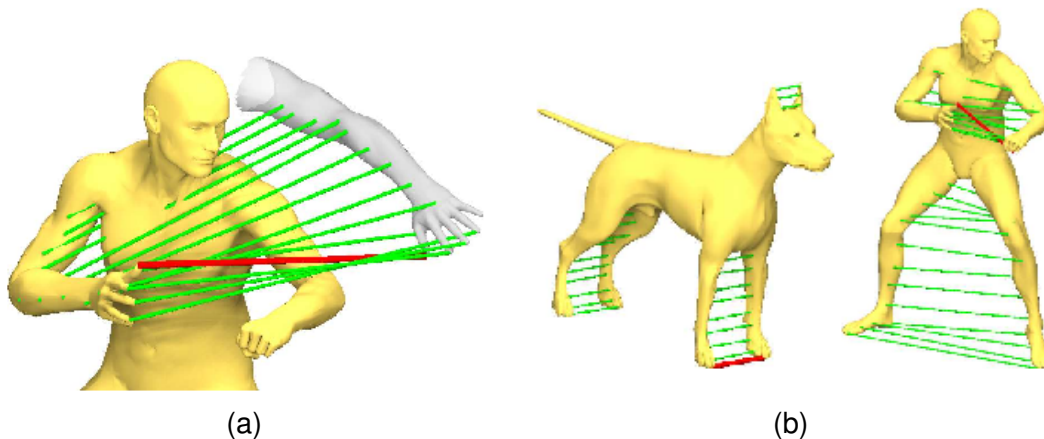


FIGURE 1.21 – Calcul de similarité et symétrie en utilisant la méthode proposée dans [Ovsjanikov et al., 2010]. (a) : correspondance entre un modèle entier et une partie d'un modèle. (b) : détection de symétrie dans des modèles.

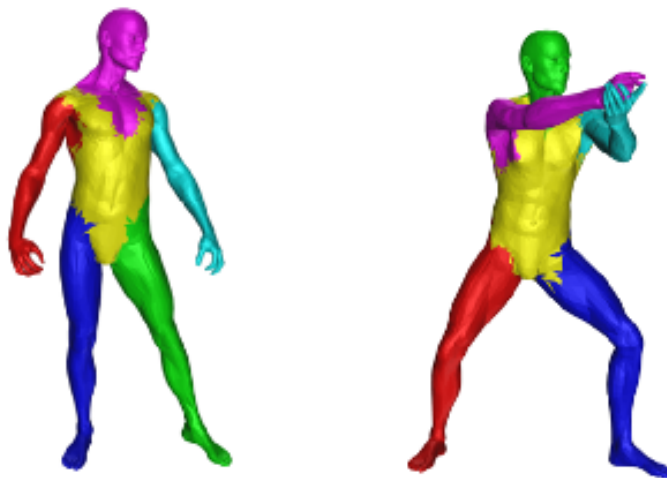


FIGURE 1.22 – Un exemple de segmentation en utilisant la méthode de combinaison de HKS et PBC [Skraba et al., 2010].

1.3.3/ DISTANCES ENTRE DISTRIBUTION : DISTANCES STATISTIQUES

Pour chaque objet, les valeurs associées à ses sommets, calculées en utilisant un descripteur de distribution, sont résumées par un histogramme de b tiroirs. La distance d entre deux objets est la distance entre leurs histogrammes. Plus rigoureusement, il s'agit

d'une pré-distance, car elle ne vérifie pas toujours l'inégalité triangulaire. Nous supposons que d est positive, et que $d(o, o)$ est égale à zéro pour n'importe quelle forme o , et que $d(o, o)$ appartient à l'intervalle $[0, 1]$.

Dans la littérature, il existe plusieurs distances entre distributions : la divergence de Kullback–Leibler, la distance de Hellinger, la distance de Bhattacharyya, la loi du χ^2 , etc.

Definition 3. Nous notons H l'histogramme (distribution) calculé à partir des n sommets d'une forme donnée. $b \approx \sqrt{n}$ est le nombre de tiroirs de l'histogramme H . L'histogramme H est défini par :

$$H(i) = \frac{m_i}{n}, i = 0, 1, \dots, b - 1$$

où m_i est le nombre de points dont la valeur du descripteur est dans l'intervalle $[i/b, (i + 1)/b]$.

Les distances statistiques entre deux formes A et B définies respectivement par leur distributions H_1 et H_2 sont définies comme suit.

— **Divergence de Kullback–Leibler :**

$$D_{KL}(H_1, H_2) = \sum_i H_1(i) \log \frac{H_1(i)}{H_2(i)}$$

— **Distance de Bhattacharyya :**

$$D_{BH}(H_1, H_2) = -\ln\left(\sum_i \sqrt{H_1(i)H_2(i)}\right)$$

— **La norme L_p :**

$$L_p(H_1, H_2) = \left(\sum_i |H_1 - H_2|^p\right)^{\frac{1}{p}}$$

— **Le test du χ^2 :**

$$\chi^2(H_1, H_2) = \frac{1}{2} \sum_{j=0}^{b-1} \frac{(H_1(j) - H_2(j))^2}{H_1(j) + H_2(j)}$$

Dans nos expériences tout au long de ce manuscrit, nous utiliserons le test du χ^2 pour calculer la distance entre les distributions.

1.4/ CONCLUSION

Dans ce premier chapitre, nous avons introduit les notions de base de l'appariement et de la recherche par forme clef. Nous avons aussi donné un état de l'art des méthodes utilisées pour la représentation et l'appariement de formes, selon leurs différentes catégories. Les descripteurs de formes utilisés dans le domaine de la recherche par forme clef peuvent être repartis en trois classes principales : les descripteurs géométriques, les descripteurs structurels et topologiques, et les descripteurs par prises de vues. Les descripteurs géométriques sont répartis en deux classes : les descripteurs locaux, et les descripteurs globaux. À ces descripteurs peuvent se rajouter les descripteurs basés sur la géométrie de diffusion et la décomposition spectrale. Ce type de descripteur prend en

compte des informations géométriques locales pour décrire la totalité de l'objet. Nous avons introduit les descripteurs de distribution de forme. Ces descripteurs calculent la signature des sommets d'un objet, et les transforment ensuite en un histogramme. La distance entre deux objets est la distance entre leur histogramme. Nous utiliserons ce type de descripteurs pour la définition des différentes contributions de notre travail.

QUELQUES CONCEPTS UTILES

2.1/ INTRODUCTION

L'objectif de notre thèse est de fournir un ensemble de contributions dans le domaine de l'appariement de formes et la recherche par forme clef. Les méthodes que nous présenterons tout au long de ce manuscrit sont générales, c-à-d qu'elles ne sont pas dédiées à une application particulière ; les méthodes proposées peuvent être adaptées et réutilisées quel que soit le type de représentation d'objets utilisé, et quelle que soit l'application.

Dans ce chapitre, nous présentons les différents outils et concepts nécessaires pour la mise en oeuvre des différentes approches de notre travail. Dans la Section 2.2, nous décrivons les différentes représentations d'objets, plus particulièrement les représentations utilisées par nos contributions. Les bases de données sont définies dans la Section 2.3. Nous introduisons les mesures d'évaluation du processus de recherche dans la Section 2.4.

2.2/ REPRÉSENTATION DES OBJETS

2.2.1/ LES OBJETS 2D

Il existe différentes représentations des images numériques, les plus courantes sont les images de niveaux de gris, les images en couleur, et les images binaires. Les types des traitements et les algorithmes utilisés dans le domaine de l'analyse et de reconnaissance des formes dépendent des images utilisées.

Habituellement, des traitements sont appliqués aux différentes types d'images afin d'extraire les informations décrivant les objets qu'une image peut contenir : détection de contours, segmentation, filtrage, etc.

Dans notre travail, nous n'utilisons des images 2D que dans le chapitre suivant : les images binaires sont utilisées. Les contributions principales de notre travail sont appliquées à des objets 3D.

Les images binaires sont la représentation d'images 2D la plus simple. Elles sont représentées sous forme d'une matrice dont les éléments sont codés dans la base 2 (0 ou 1). Un objet contenu dans une image binaire est donc l'ensemble des pixels blancs (ou noirs) de l'image binaire.

Afin d'évaluer notre méthode décrite dans le chapitre suivant, nous utilisons des images binaires de la base de données MPEG-7 [Lai et al., 2009]. La Figure 2.1 montre quelques exemples d'objets 2D utilisés.



FIGURE 2.1 – Un exemple d'images binaires représentant des modèles 2D.

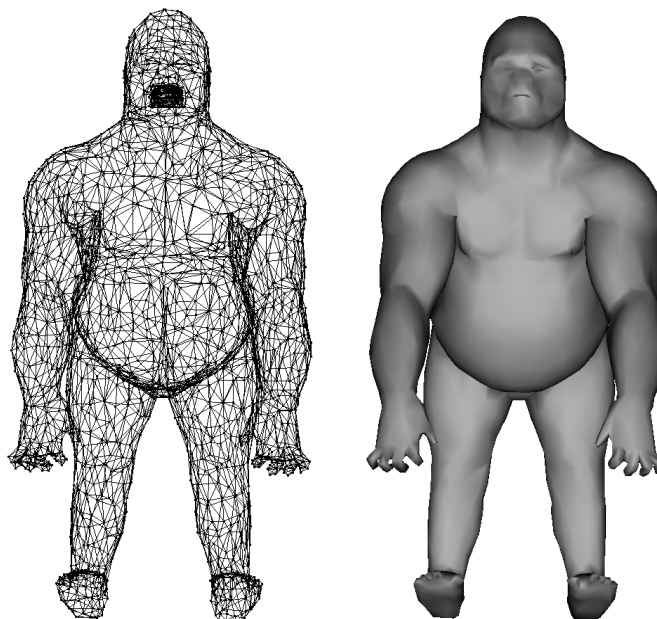


FIGURE 2.2 – Un exemple d'un objet polygonale. gauche : facettes triangulaires. droite : objet de synthèse obtenu en utilisant le lissage de Gouraud.

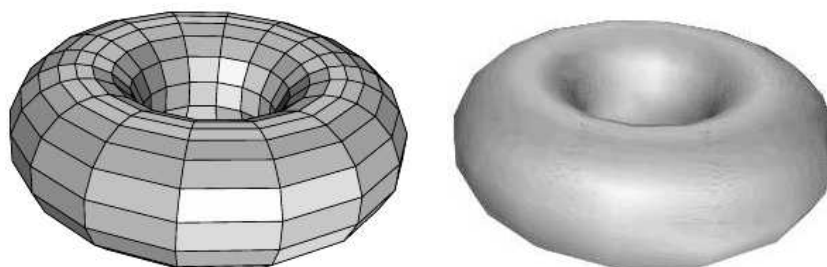


FIGURE 2.3 – Un tore modélisé par la surface de Bézier <http://morpheo.inrialpes.fr/people/Boyer/Teaching/RICM/c5.pdf>

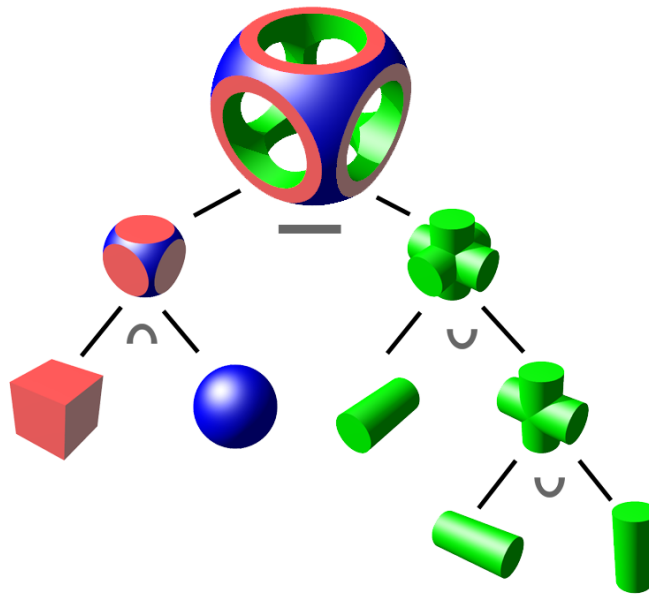


FIGURE 2.4 – Exemple d'objets CSG construits en utilisant différentes opérations https://en.wikipedia.org/wiki/Constructive_solid_geometry

2.2.2/ LES OBJETS 3D

Actuellement, avec l'avancement des technologies de numération et de modélisation, différentes variantes des modèles 3D sont massivement produites. La modélisation 3D consiste à construire des modèles 3D en utilisant un ensemble de données géométriques permettant de décrire le modèle à représenter. Il existe différents types de modélisations ; les principaux types sont :

- **Modélisation polygonale** : C'est le type de modélisation le plus utilisé. Les modèles sont des polyèdres. La Figure 2.2 montre un exemple d'un modèle polygonal. Dans le cas le plus simple et le plus fréquent, tous les polygones du polyèdre sont des triangles.
- **Modélisation par surfaces courbes** : un modèle est construit à partir des surfaces générées avec des points de contrôles. Il existe plusieurs types de surfaces, citons à titre indicatif les B-Splines, les surfaces de Bézier, et les Catmull-Rom cubiques. La Figure 2.3 montre un exemple d'un modèle construit en utilisant les courbes de Bézier.
- **Modélisation par surfaces implicites** : les objets sont définis par des surfaces décrites par une équation de la forme $f(x, y, z) = 0$, $f : \mathbb{R}^3 \rightarrow \mathbb{R}$. Le signe de $f(x, y, z)$ permet de savoir si le point x, y, z est à l'intérieur ou à l'extérieur de l'objet. Les quadriques, les tores sont des exemples de surfaces ayant une équation implicite simple.
- **Modélisation par géométrie** : la géométrie de construction des solides, en anglais appelée *Constructive Solid Geometry* (CSG), consiste à construire des objets solides en appliquant des opérations booléennes, ou des transformations géométriques sur des formes de bases (Figure 2.4) ; ces formes de bases sont typiquement des surfaces implicites.
- **Modélisation fonctionnelle et paramétrique** : ce type de modélisation est le plus

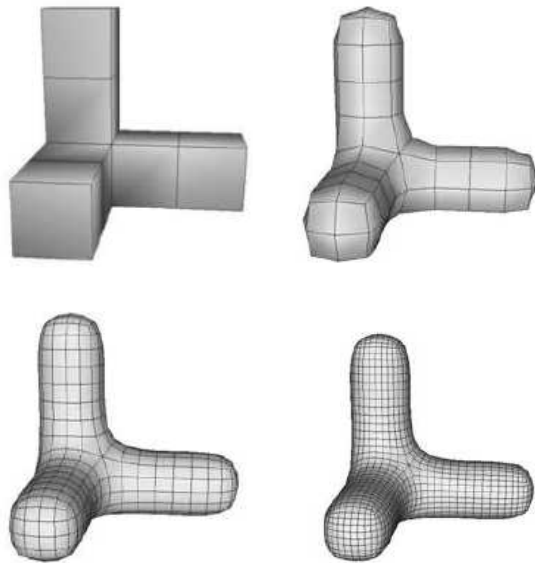


FIGURE 2.5 – Différents niveaux de subdivisions pour un maillage rectangulaire [Boumzaid et al., 2009]. haut droite : maillage original.

utilisé pour des modèles de CFAO (*CAD/CAO*). Une fonction, au sens informatique du terme, construit l'objet ; les arguments de la fonction sont les paramètres définissant l'objet (dimensions, nombre de marches pour un escalier ou de dents pour des roues dentées, etc).

- **Modélisation volumique** : elle permet de représenter l'intérieur d'un objet. Nous décrivons en détails ce type de modélisation dans la Section 2.2.4.

Il existe bien d'autres méthodes de modélisation ; elles sont généralement dérivées des méthodes décrites ci-dessus. Les surfaces de subdivision par exemple représentent une variante des surfaces à pôles. A partir d'une surface polygonale, un procédé itératif subdivise chaque face en sous faces selon un schéma de subdivision (Figure 2.5). La surface résultante est la limite du processus de subdivision.

La modélisation B-rep est une autre méthode de modélisation géométrique permettant de représenter des objets solides par leur frontière, en utilisant le plus souvent des surfaces B-splines, ou de Bézier. Elle est une généralisation de la modélisation polygonale : les polygones sont remplacés par des morceaux de surfaces gauches.

La modélisation polygonale est extrêmement répandue en raison de sa simplicité, et la facilité de stockage externe dans des fichiers.

2.2.3/ MODÈLES POLYGONAUX : MAILLAGES

Il s'agit de représenter un objet (une surface) par un polyèdre. On appelle ce processus la discrétisation de la surface. Les sommets sont reliés par des segments que l'on appelle arêtes, ce qui permet de trouver une représentation approximative de la surface. Cette discrétisation de la surface est appelée maillage. Pour construire un maillage à partir d'un nuage de points, il faut disposer de différentes informations telles que les arêtes adjacentes à un sommet, les voisins du sommet, et les arêtes partageant le même sommet. Le calcul de toutes ces informations nécessite un temps de calcul et de stockage mé-

moire important. En pratique, pour éviter de refaire les calculs, et pour consommer moins d'espace mémoire, toutes ces informations sont calculées au préalable et stockées dans des fichiers. Il suffit donc, lors de la reconstruction d'un maillage, de lire les différentes informations sauvegardés dans ces fichiers. Il existe différents types de formats de fichiers tels que les formats off, obj, VRML etc . . . Nous utilisons le format obj. Un fichier obj basique d'un maillage triangulaire est sous la forme suivante :

```
#les coordonnées des sommets v
v x1 y1 z1 # coordonnées du sommet 1
v x2 y2 z2 # coordonnées du sommet 2
.
.
.
#les indices des sommets d'un triangle (facettes f) : des nombres entiers
f fi fj fk # indices des sommets de la face 1
f fi fm fn # indices des sommets de la face 2
.
.
.
#Les coordonnées des vecteurs normales n
n x'1 y'1 z'1 # normale de la face 1
n x'2 y'2 z'2 # normale de la face 2
.
.
.
```

2.2.4/ REPRÉSENTATION VOLUMIQUE DISCRÈTE DES OBJETS 3D

Dans le chapitre suivant, nous utiliserons des représentations volumiques discrètes, calculées à partir des modèles polygonaux. Ce type de représentation discrétise l'espace 3D en un tableau 3D de voxels : un voxel est un petit cube élémentaire, plein ou vide : c'est un pixel 3D. Cette représentation est largement utilisée dans des techniques d'imagerie médicale, tel que l'IRM et les scanner 3D, pour représenter les tissus des organes. La Figure 2.6 représente un modèle volumique (b) obtenu à partir d'un modèle polygonal (a). Seuls les voxels pleins sont affichés.

Pour obtenir un modèle volumique à partir d'un modèle polygonal, nous utilisons la méthode de l'arbre octal [Meagher, 1980]. Il s'agit d'un partitionnement récursif d'un cube englobant le polyèdre, dans lequel les feuilles représentent les voxels. Nous utilisons le logiciel *binvox* [<http://www.cs.princeton.edu/min/binvox/>,] pour obtenir des modèles volumiques à partir des maillages 3D.

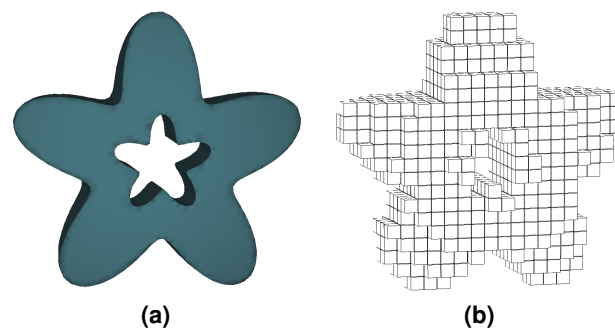


FIGURE 2.6 – Représentation volumique d’un objet 3D. (a) : modèle polygonal. (b) : modèle volumique.

2.3/ BASES DE DONNÉES

Le processus de recherche par forme clef consiste à trouver dans une base de données les formes similaires à cette forme. Potentiellement, la base de données utilisée dans le processus de recherche par forme clef est le Web. Plus modestement, afin d’évaluer les différents algorithmes présentés dans ce manuscrit, nous utilisons des bases de données telles que la base de données MPEG-7 [Lai et al., 2009] pour des objets 2D (uniquement chapitre 2), et TOSCA [YA. M. Bronstein, 2007] et SHREC’11 [Li et al., 2012] pour des objets 3D.

- **MPEG-7** : la base de données MPEG-7 [Lai et al., 2009] contient des images binaires représentant des formes 2D. Elle contient 70 catégories, 20 images binaires pour chacune des catégories. Les différents éléments de cette base de données sont présentés dans la figure 2.7.
- **TOSCA** : cette base de données [YA. M. Bronstein, 2007] contient 148 modèles 3D (par ex. *Cats*, *Centaur*s, *dogs*, *Wolves*, *Horses*, *Lions*, *Gorillas*, *Shark*, personnages féminins et masculins). Les modèles sont répartis en 10 catégories avec une variété de poses dans chaque catégorie. La figure 2.8 montre les différentes formes contenues dans cette base de données.
- **SHREC’11** : La base de données SHREC’11 [Li et al., 2012] contient environ 600 objets 3D, classés dans différents groupes, contenant presque le même nombre de modèles. Les différents objets de cette base sont illustrés dans la figure 2.9.

Dans les deux bases de données TOSCA et SHREC, les modèles 3D sont représentés sous forme de maillages triangulaires stockés dans des fichiers ASCII au format obj. Le nom de chaque fichier donne implicitement sa classe (par exemple *Cat*, *Dog*, etc.), ce qui permet de mesurer les performances des résultats de recherche. Tous les modèles sont normalisés par notre logiciel : la somme des aires de tous les triangles pour chaque forme est égale à un.

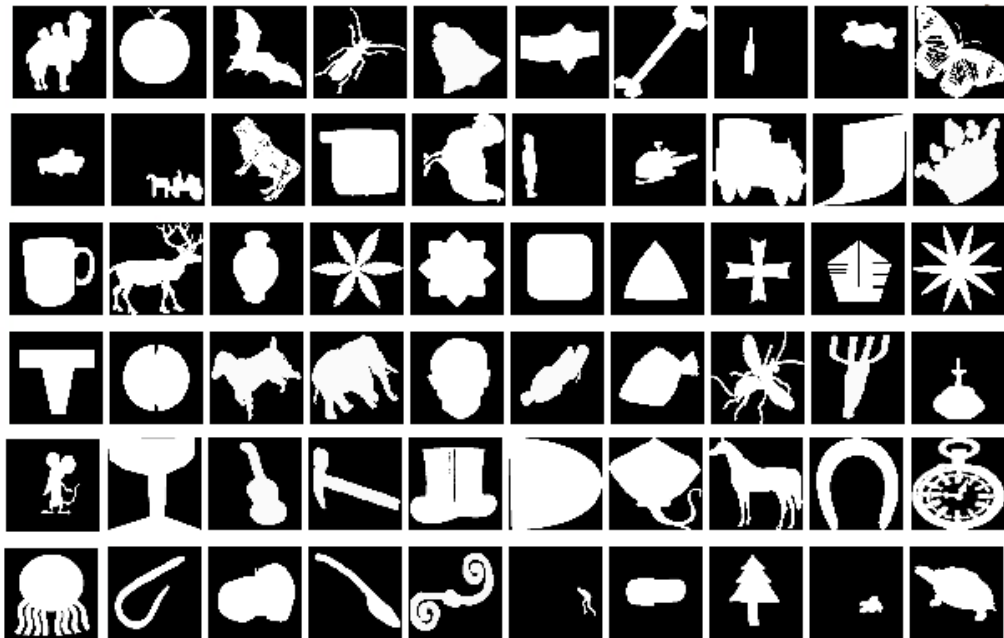


FIGURE 2.7 – Les éléments de la base de données MPEG-7.

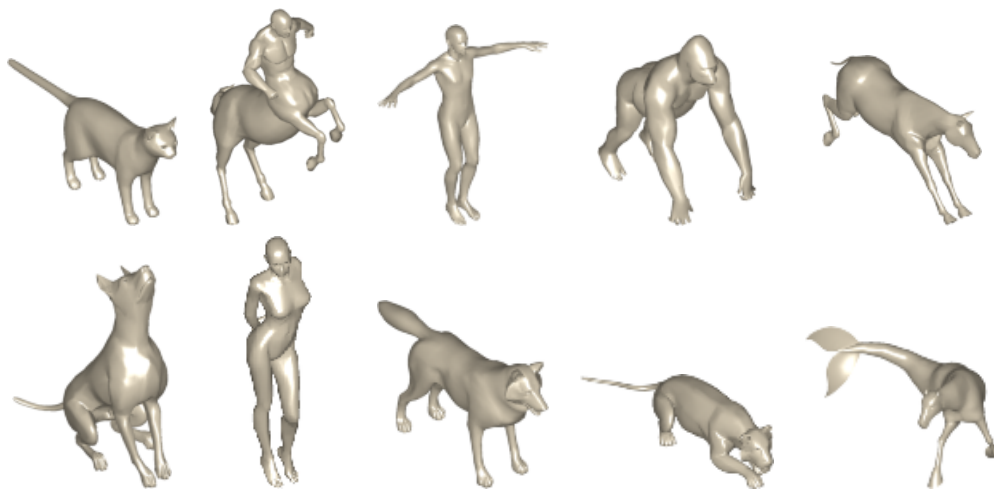


FIGURE 2.8 – Les différents modèles de la base de données TOSCA.

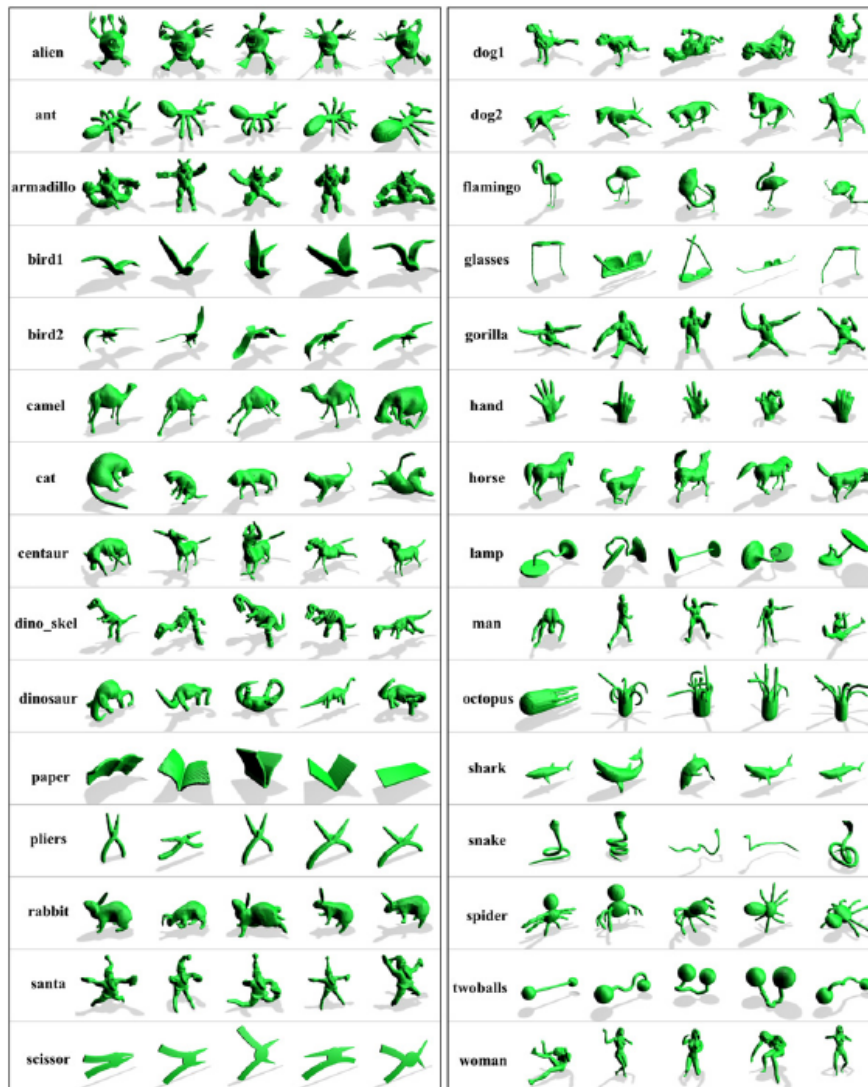


FIGURE 2.9 – Les différents modèles de la base de données SHREC'11.

2.4/ MESURER LA PERFORMANCE DU PROCESSUS DE RECHERCHE PAR FORME CLEF

Cette section définit les notions classiques de rappel et précision [Manning et al., 2008] utilisées pour mesurer la performance des algorithmes de recherche par forme clef. La précision et le rappel sont les deux mesures les plus utilisées dans l'évaluation des méthodes de recherche par forme clef.

Dans nos expériences, pour chaque requête Q , le processus de recherche fournit les m formes les plus proches : certaines sont correctes et sont appelées *Vrai Positif* (VP), les autres sont fausses et sont appelées *Faux Positif* (FP). Les éléments similaires à la requête qui ne sont pas retrouvés sont appelés *Faux Négatif* (FN). Par définition, $m = |TP| + |FP|$.

La précision de la recherche est définie comme le rapport du nombre de *Vrai Positif* (TP)

et du nombre total des formes extraites m :

$$P = \frac{|TP|}{|TP| + |FP|} = \frac{|TP|}{m} \quad (2.1)$$

Le rappel est le rapport du nombre de *Vrai Positif* (TP) et du nombre total de formes pertinentes (TP + FN) :

$$R = \frac{|TP|}{|TP| + |FN|} \quad (2.2)$$

La précision et le rappel peuvent être représentés sous forme de courbe précision-rappel. Chaque point de la courbe représente les coordonnées des valeurs de rappel et de précision calculées pour chaque requête.

2.5/ CONCLUSION

Dans ce chapitre, nous avons présenté les différents concepts qui nous serviront dans les chapitres à venir. Nous avons donc introduit les différents représentations d'objets, les bases de données utilisées, ainsi que les mesures d'évaluation du processus de recherche par forme clef.

Dans le prochain chapitre, nous allons présenter notre première contribution : nous présentons un nouveau descripteur de formes utilisé pour apparier des objet 2D et 3D.

DÉCOMPOSITION D'OBJETS EN SQUELETTE D'ELLIPSOÏDES

3.1/ INTRODUCTION

Ce chapitre décrit un nouveau descripteur de formes 2D/3D. Ce descripteur permet de décrire les points intérieurs de l'objet sous forme de squelettes d'ellipsoïdes.

En effet, dans le but d'apparier des formes, on considère les points intérieurs de l'objet (pixels, ou voxels) : le nuage de points est partitionné en ellipsoïdes. Pour cela, on utilise une variante de l'algorithme *k-means*, appelé aussi algorithme des nuées dynamiques, ou algorithme des centres mobiles. On cherche la meilleure décomposition de l'objet en k -parties (régions) avec une variante de l'algorithme des nuées dynamiques de Diday [Diday, 1971]. L'idée est de trouver la meilleure partition des points intérieurs en différentes régions, sans connaissance préalable du nombre optimal de parties (classes) k .

Pour k donné, on cherche la meilleure partition en k classes : c'est celle qui maximise l'homogénéité des k classes. Le nombre optimal k de classes est le plus petit nombre k pour lequel la variance inter classe est localement maximale.

La similarité de deux formes A et B , représentées par leurs k ellipsoïdes, est équivalente à la résolution d'un problème de couplage optimal résolu par la méthode Hongroise introduite dans [Kuhn, 1955]. La solution optimale est celle qui donne la distance minimale entre A et B . Cette distance est la somme des distances euclidiennes entre les différents vecteurs signatures associés aux régions des formes A et B .

Les expériences effectuées à la fois sur des modèles 2D et 3D montrent les bonnes performances de la méthode.

3.2/ DÉCOMPOSITION D'OBJETS PAR L'ALGORITHME DES NUÉES DYNAMIQUES

3.2.1/ ALGORITHME DES NUÉES DYNAMIQUES

L'algorithme des nuées dynamiques [Diday, 1971], appelé aussi *k-means*, *k-moyenne* ou algorithme des centres mobiles est une méthode de classification automatique de don-

nées. Il s'agit de partitionner un ensemble de données, ou points dans notre cas, en k parties disjointes. Chaque donnée est affectée à la classe la plus proche (la classe la plus pertinente).

L'affectation est basée sur le calcul de distance (le plus souvent la distance euclidienne est utilisée) entre le représentant de la classe (son centre de gravité) et la donnée à classifier. Autrement dit, les données (ou points) sont regroupées en classes (ou régions) les plus homogènes possibles.

Cet algorithme est largement utilisé dans différents domaines de l'analyse de données tels que : la biologie, la sociologie, la médecine, etc. En informatique, cet algorithme est utilisé dans la fouille de données (en anglais *data mining*), l'analyse et la segmentation d'images, la segmentation de maillages 3D, etc.

Dans ce chapitre, nous utilisons l'algorithme des nuées dynamiques pour décomposer un objet 2D ou 3D en k classes différentes. Chaque classe (partie) c_i est représentée par son centre de gravité \bar{c}_i . L'union des classes donne le nuage global $\bigcup c_i = S$, et chaque point P de coordonnées (x, y, z) ne doit appartenir qu'à une seule et unique classe : $\bigcap c_i = \emptyset$. Pour un nombre k de classes donné, le partitionnement s'effectue en utilisant l'algorithme 1.

Algorithme 1 Partitionnement de points en k classes utilisant l'algorithme k-means(P, k, C)

Entrée : N points $P_i(x_i, y_i, z_i)$, k centres, $C = \{c_1, c_2, \dots, c_k\}$.

Calculer chaque \bar{c}_l pour $l = 1, \dots, k$

Répéter jusqu'à convergence :

pour $i = 1, \dots, N$:

Notons $c_n = \operatorname{argmin}_l^k(\operatorname{distance}(P_i, \bar{c}_l))$

Affecter(P_i, c_n)

Mettre à jour les centres \bar{c}_l pour $l = 1, \dots, k$.

Sortie : Un ensemble de k -classes c_1, \dots, c_k .

A partir des k centres initiaux, nous associons chaque point au centre le plus proche. Le centre de gravité $\bar{c}_l = (\bar{x}_l, \bar{y}_l, \bar{z}_l)$ de la classe c_l est donné par :

$$\bar{c}_l = \frac{1}{n_l} \sum_{P_j \in c_l} P_j = \frac{1}{n_l} \left(\sum_{P_j \in c_l} x_j, \sum_{P_j \in c_l} y_j, \sum_{P_j \in c_l} z_j \right) \quad (3.1)$$

où n_l est le nombre des points de la classe c_l . La distance entre un point et la classe c_l est la distance euclidienne entre le point et le centre de gravité \bar{c}_l de la classe c_l . Chaque point est associé à la classe la plus proche, et les centres de gravité sont recalculés à chaque itération. Le processus se répète jusqu'à convergence, c-à-d quand les centres de gravité deviennent stables.

La Figure 3.1 montre un exemple d'un objet 3D décomposé en différentes classes en utilisant l'algorithme 1.

En fait, après un certain nombre d'itérations, les centres se stabilisent, et on obtient des points répartis en k classes. Néanmoins, en pratique, il n'est pas évident d'anticiper le bon nombre k qui permet de trouver la meilleure répartition des points. Un choix aléatoire de k risque d'aboutir à des mauvais résultats. Pour résoudre ce problème, on prend en considération un critère permettant de trouver le bon nombre de classes k . Pour cela, on recourt à des notions de statistiques.

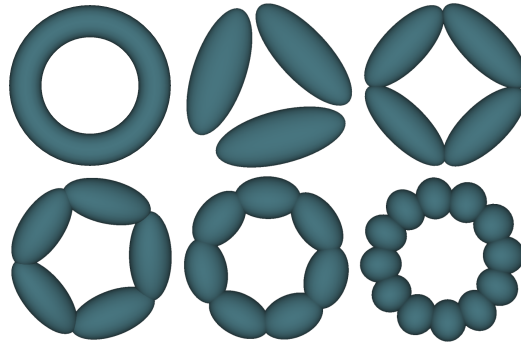


FIGURE 3.1 – Un objet 3D décomposé en 3, 4, 5, 7, 12, classes.

3.2.2/ PARTITIONNEMENT OPTIMAL

En statistiques, la variance d'un ensemble d'observations mesure la dispersion des données autour de leur moyenne, c'est un moment d'ordre 2. La variance de $P = \{P_1, P_2, \dots, P_n\}$ est définie comme suit :

$$\text{var}(P) = \frac{1}{n} \sum_{i=1}^n (P_i - \bar{P}) \cdot (P_i - \bar{P}) \quad (3.2)$$

C'est la moyenne des carrés des distances, où :

n est le nombre de points P_i ,

P_i est un point de coordonnées (x_i, y_i, z_i) ,

$\bar{P}(\bar{x}, \bar{y}, \bar{z})$ est la moyenne des observations (centre de gravité) telle que :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i, \quad \bar{z} = \frac{1}{n} \sum_{i=1}^n z_i$$

Donc :

$$\text{var}(P) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2 \quad (3.3)$$

la variance ne dépend pas du repère utilisé, car la distance n'en dépend pas.

Nous pouvons exprimer l'équation 3.2 comme suit :

$$\text{var}(P) = \text{var}(X) + \text{var}(Y) + \text{var}(Z) \quad (3.4)$$

avec :

$$\text{var}(X) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2, \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

TABLE 3.1 – Valeurs des variances *intra* *inter* classes selon le nombre de classes k .

k	1	2	3	4	5	6	7	8	9
<i>Intra</i>	0.84	0.44	0.25	0.17	0.06	0.02	0.015	0.013	0.011
<i>Inter</i>	0	0.17	0.25	0.32	0.38	0.39	0.42	0.40	0.36

$$\text{var}(Y) = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

$$\text{var}(Z) = \frac{1}{n} \sum_{i=1}^n (z_i - \bar{z})^2, \quad \bar{z} = \frac{1}{n} \sum_{i=1}^n z_i$$

La valeur de la variance donne une indication sur l'homogénéité de la classe. Plus cette valeur est petite, plus la classe est homogène. Quand la variance est grande, cela signifie que les points de la classe sont éloignés les uns des autres.

La variance intra-classe est la moyenne des variances de chacune des classes.

$$\text{var_intra}(c_1, c_2, \dots, c_k) = \frac{1}{k} \sum_{j=1}^k \text{var}(c_j) \quad (3.5)$$

De la même façon, on peut calculer la dispersion des centres de classes par rapport à leur centre (centre de gravité de l'objet). Cette valeur est dite variance inter-classes.

$$\text{var_inter} = \frac{1}{k} \sum_{j=1}^k (\bar{c}_j - \bar{s}) \cdot (\bar{c}_j - \bar{s}) \quad (3.6)$$

\bar{c}_j : ce sont les centres de gravité des classes c_j .

\bar{s} : représente le centre de gravité de l'objet.

Le Tableau 3.1 et la Figure 3.4 montrent la variation des valeurs des variances *intra* *inter* classes en fonction de k . Quand k égale N (N est le nombre de points), la variance intra-classe est nulle, et la variance inter-classe est la variance de P .

La Figure 3.2 montre un exemple de classification d'un ensemble de nombres entiers en différentes classes.

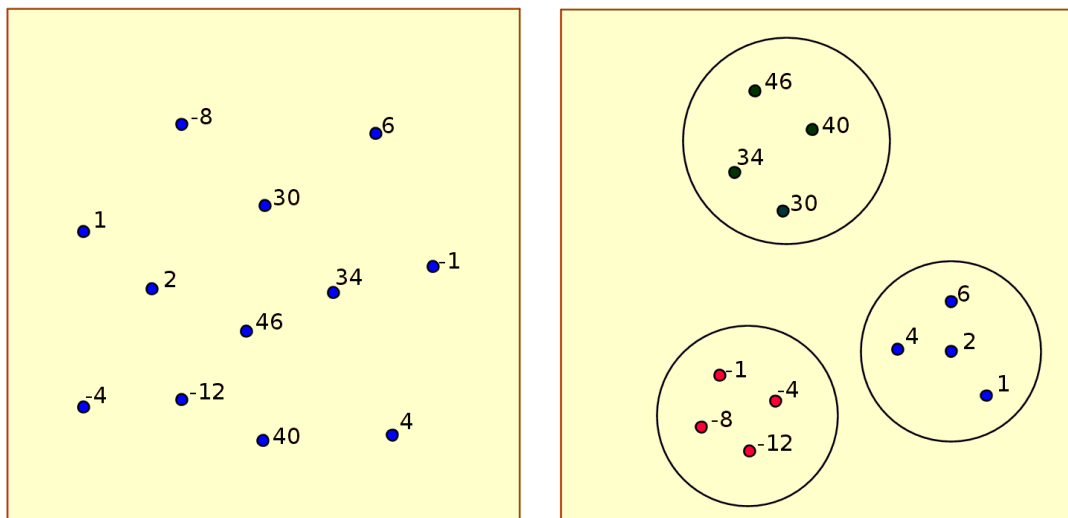


FIGURE 3.2 – Un exemple de classification 1D. Gauche : un ensemble de nombres entiers. Droite : partitionnement des nombres entiers en classes.

3.2.2.1/ UN EXEMPLE 1D

La méthode proposée permet de déterminer le bon nombre de classes. En effet, dans cet exemple, nous pouvons distinguer trois groupes : un groupe contenant des nombres entiers négatifs, un deuxième groupe des nombres positifs, et un dernier groupe contenant des nombres positifs de grandes valeurs. Dans l'exemple de la Figure 3.2, nous pouvons constater que les données appartenant à la même classe sont très proches les unes des autres, tandis que celles appartenant à des classes différentes sont éloignées. Cela signifie que la variance inter-classe doit être la plus grande possible. La différence entre les nombres d'une même classe est petite, tandis que la différence entre des classes différentes est grande. Le meilleur partitionnement doit avoir une variance inter-classe maximale.

3.2.2.2/ LE NOMBRE k OPTIMAL

L'exemple précédent illustre le caractère unimodulaire de la fonction $var_inter(k)$. [Ahamed Shafeeq, 2012] a proposé d'utiliser comme valeur du k la plus petite valeur telle que :

$$var_inter(k - 1) < var_inter(k)$$

et

$$var_inter(k) > var_inter(k + 1)$$

La Figure 3.3 montre les différentes classes trouvées en utilisant l'algorithme 1, selon les différentes valeurs de k . Le tableau 3.1 résume les différentes valeurs des variances intra/inter classes correspondant aux classes illustrées dans la Figure 3.3. A partir de cet exemple, nous pouvons encore une fois voir que la valeur de la variance intra-classe diminue quand la valeur de k augmente. Dans cet exemple, la valeur de la variance

inter-classes est maximale pour $k = 7$; donc, le meilleur nombre de classes k est 7. La Figure 3.4 montre la variation des valeurs de variances. La Figure 3.5 montre un exemple d'objets 2D décomposés k (optimal) classes. Dans cette figure, k a été choisi en utilisant le critère de [Ahmed Shafeeq, 2012].

En revanche, cette décomposition n'est pas nécessairement la meilleure pour tous les jeux de données. Dans le cas d'objets plus complexes, c-à-d contenant plus de détails : certaines classes peuvent être mal décomposées.

Afin d'introduire notre méthode utilisée pour la décomposition finale d'objets en classes, nous décrivons dans la section suivante la notion des ellipsoïdes associés aux classes.

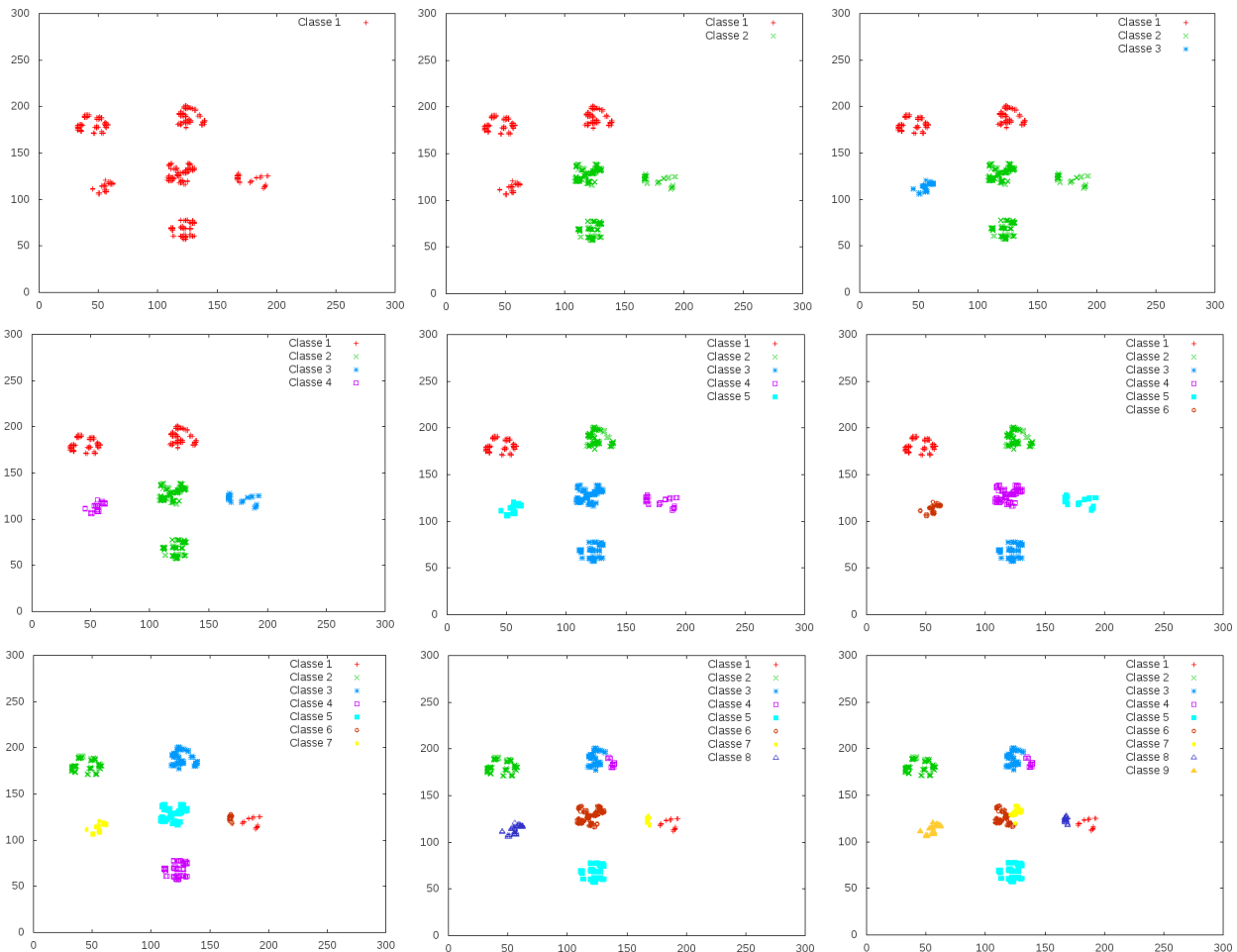


FIGURE 3.3 – Un exemple 2D de partitionnement en utilisant l'algorithme des nuées dynamiques. Nous enregistrons les différentes valeurs des variance intra/inter classes selon le nombre de classes k .

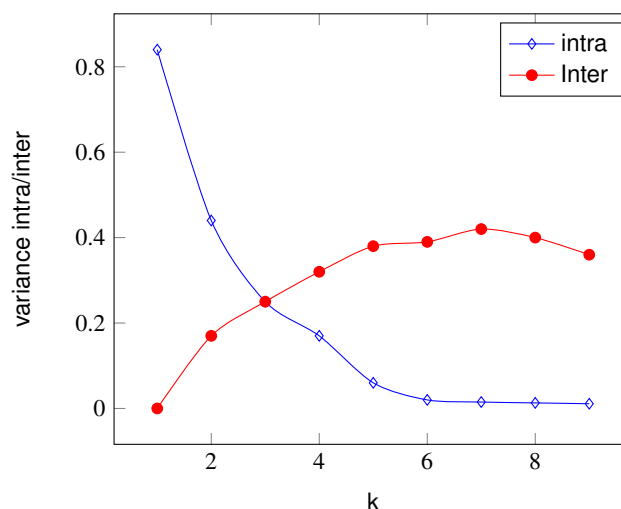


FIGURE 3.4 – Variation des valeurs intra/inter classes selon le nombre de classes k . $var_intra(k)$ est une fonction descendante, $var_inter(k)$ est une fonction unimodale.



FIGURE 3.5 – Objets 2D décomposés en k classes optimales.

3.3/ MATRICE DE COVARIANCE ET SQUELETTE D'ELLIPSOÏDES

3.3.1/ MATRICE DE COVARIANCE

Chaque classe d'un objet 2D/3D peut être représentée par un ellipsoïde. Le centre de l'ellipsoïde est le centre de gravité de la classe, et ses axes sont les vecteurs propres de la matrice de covariance. Ses diamètres sont les racines carrées des valeurs propres correspondant aux vecteurs propres. D'un point de vue physique, les vecteurs propres sont les axes principaux d'inertie.

Pour calculer un ellipsoïde correspondant à une classe, nous avons besoin de calculer la matrice de covariance M_c . La matrice de covariance mesure la distribution des points autour des axes d'inertie du nuage de points. Cette matrice est indépendante du repère utilisé, et est donc invariante par isométrie. Elle résiste aux bruits et aux aléas de l'échantillonnage. La matrice de covariance M_c est donnée par :

$$M_c = \begin{pmatrix} \text{var}(X) & \text{cov}(XY) & \text{cov}(XZ) \\ \text{cov}(YX) & \text{var}(Y) & \text{cov}(YZ) \\ \text{cov}(ZX) & \text{cov}(ZY) & \text{var}(Z) \end{pmatrix} \quad (3.7)$$

avec :

$$\text{cov}(XY) = \text{cov}(YX) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}) \times (y_i - \bar{y})$$

$$\text{cov}(XZ) = \text{cov}(ZX) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}) \times (z_i - \bar{z})$$

$$\text{cov}(YZ) = \text{cov}(ZY) = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}) \times (z_i - \bar{z})$$

La matrice de covariance M_c est symétrique et définie positive, en général. Donc ses valeurs propres sont réelles et positives, et ses vecteurs propres sont perpendiculaires. Dans les cas particuliers, tous les points sont coplanaires ($\sigma_3 = 0$), ou tous les points sont alignés ($\sigma_2 = \sigma_3 = 0$), ou tous les points sont égaux ($\sigma_1 = \sigma_2 = \sigma_3 = 0$). Même dans les cas particuliers, M_c est diagonalisable.

L'orthogonalisation de la matrice permet de déterminer ses valeurs et vecteurs propres :

$$M_c \vec{v}_\sigma = \text{diag}(\vec{\sigma}) \vec{v}_\sigma = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \vec{v}_\sigma \quad (3.8)$$

où σ_1, σ_2 , et σ_3 sont les valeurs propres de la matrice de covariances M_c , et \vec{v}_{σ_i} est le vecteur propre correspondant à la valeur propre σ_i .

La figure 3.6 montre un exemple de décomposition d'un objet en $k = 5$ classes représentées par des ellipsoïdes.

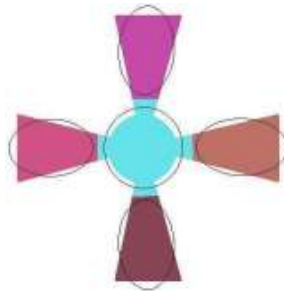


FIGURE 3.6 – Squelette d'ellipsoïde d'un objet 2D.

3.3.2/ PARTITIONNEMENT FINAL

Comme mentionné précédemment, dans certains cas, trouver le bon nombre de classes en utilisant la méthode décrite ci-dessus (Section 3.2.2.2) conduit à un partitionnement

erroné (ex. Figure 3.7) : les parties 1 et 2 ne devraient pas être de la même classe. Nous proposons de décomposer ce genre de classe en sous classes.

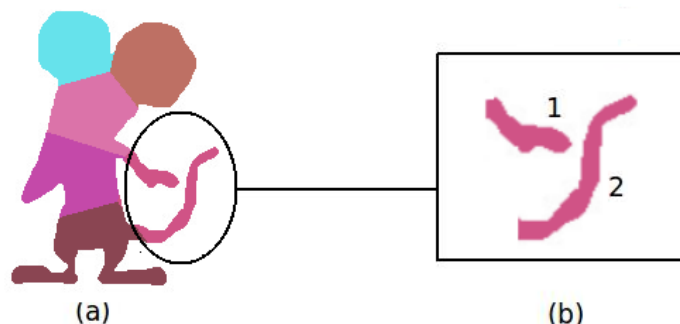


FIGURE 3.7 – Un exemple d'un partitionnement contenant des classes inconsistantes.

En effet, ce problème est dû à l'utilisation de la distance euclidienne lors du partitionnement : chaque point est associé à la classe dont le centre de gravité est le plus proche. Aucune autre information n'est prise en compte.

Pour résoudre ce problème, nous cherchons à identifier les classes inconsistantes, et nous essayons de les décomposer à leur tour en sous-classes, en utilisant la même technique décrite dans la Section 3.2.2.2. Pour déterminer si une classe est inconsistante ou pas, nous calculons une erreur. Elle mesure à quel point un ellipsoïde couvre sa classe associée. Cette erreur que nous notons ϵ représente le rapport entre l'aire des pixels d'une classe et l'aire de l'ellipsoïde associé à cette classe. Une classe est décomposée en sous classes si $\epsilon > 5\%$.

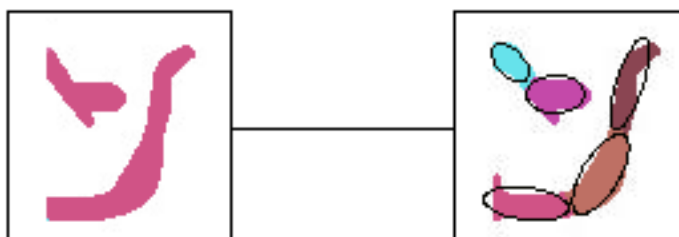


FIGURE 3.8 – Décomposition d'une classe en sous classes.

Supposons que nous continuons à incrémenter le nombre de classes au lieu d'utiliser cette technique ; dans ce cas, nous allons décomposer même les classes qui sont correctes. En utilisant notre technique, nous évitons ce cas de figure. De plus, cette technique permet une convergence rapide de notre algorithme de partitionnement.

3.3.3/ LES k -PARTITIONS INITIALES

En plus du nombre de classes k , et la métrique (distance) utilisée dans notre algorithme, l'initialisation des classes de départ joue un rôle important. Le choix des centres initiaux peut influencer les résultats de partitionnement. Ce n'est pas évident de choisir les

centres de départ. Plusieurs méthodes ont été présentées pour résoudre ce genre de problème. Nous avons choisi d'utiliser la méthode proposée dans [Guellil et al., 2009].

Definition 4. *Étant donné une classe C , et c le centre de gravité de cette classe, un point f dans C est le point le plus éloigné (nous notons $f = \text{dist}(C)$) si la distance qui sépare le point c et f est la plus grande (en utilisant la distance euclidienne). Vu qu'il y a k classes, $F = \{f_1, f_2, \dots, f_k\}$ est l'ensemble des points les plus éloignés de chaque classe. Nous notons $\text{distant}(F)$ le point dans F avec la plus grande distance au centre de gravité de sa classe.*

Initialement, l'algorithme débute avec deux centres pour calculer les deux classes de départ. Ces deux points sont les points de l'objet les plus éloignés l'un de l'autre (Figure 3.9.b).

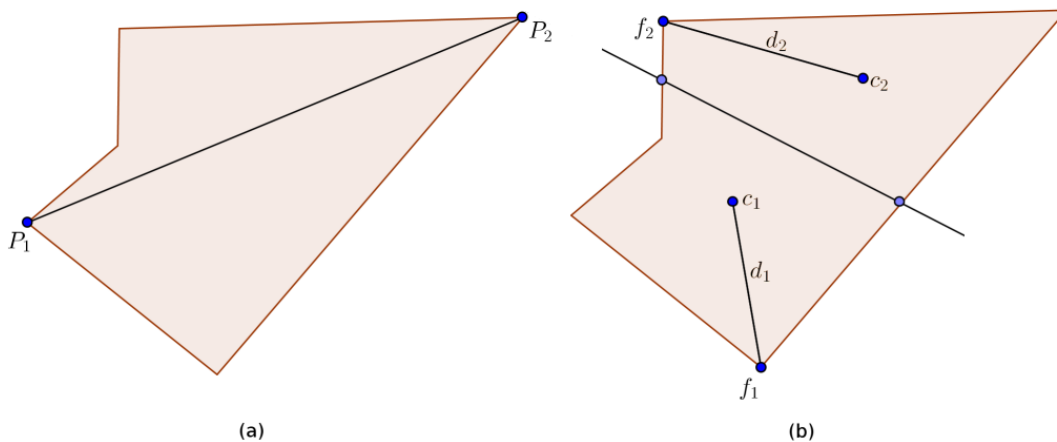


FIGURE 3.9 – Les centres utilisés pour initialiser l'algorithme des nuées dynamiques. P_1 et P_2 sont les deux points initiaux (les points les plus éloignés). (b) Les deux points les plus éloignés dans leur classe c_1 and c_2 . Si $d_1 < d_2$, P_1 est le nouveau centre, sinon c'est P_2 .

Quand k est incrémenté, et pour trouver les nouvelles classes, nous calculons pour chaque classe existante C_i , son point le plus éloigné $f_i = \text{dist}(C_i)$ (Figure 3.9.b). Donc, $\text{distant}(F)$ est le nouveau centre. La justification est que si un point est éloigné du centre de gravité de sa classe, il y a de fortes chances pour qu'il n'appartienne pas à cette classe. La Figure 3.10 montre des exemples de décomposition d'objets en utilisant l'algorithme 2. Cette méthode permet de trouver le nombre approprié k de classes.

La Figure 3.11 et la Figure 3.12, illustrent le meilleur partitionnement en ellipsoïdes pour des objets 2D et 3D.

Algorithm 2 Nouveau k -means

Données : $P_i = (x_i, y_i, z_i)$, $C = \{c_1, c_2\}$

k -means ($P, 2, C$)

Tant que ($VarInter_{new} > VarInter_{old}$) :

$c_k \leftarrow \text{distant}(\bigcup_{i=1}^k \text{dist}(C_i))$

$C \leftarrow C \cup \{c_k\}$

$k \leftarrow k + 1$

k -means(k, P, C)

Retourner k -means(k, P, C)

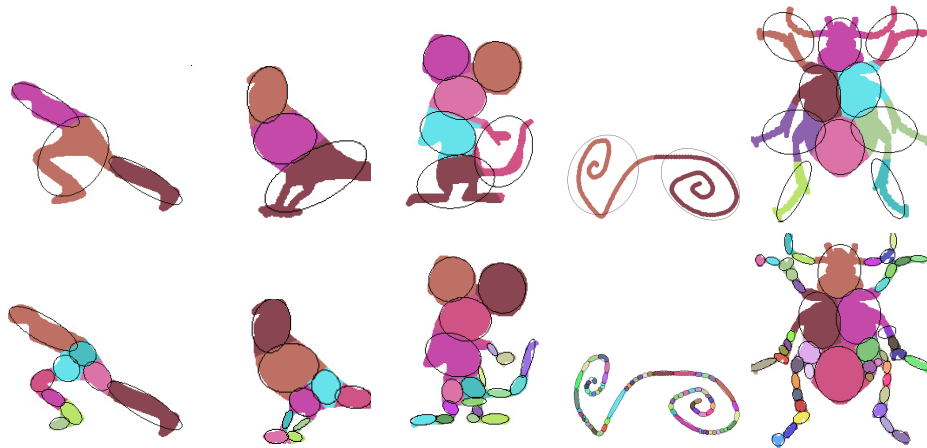


FIGURE 3.10 – Décomposition adéquate d'objets en squelettes d'ellipsoïdes.

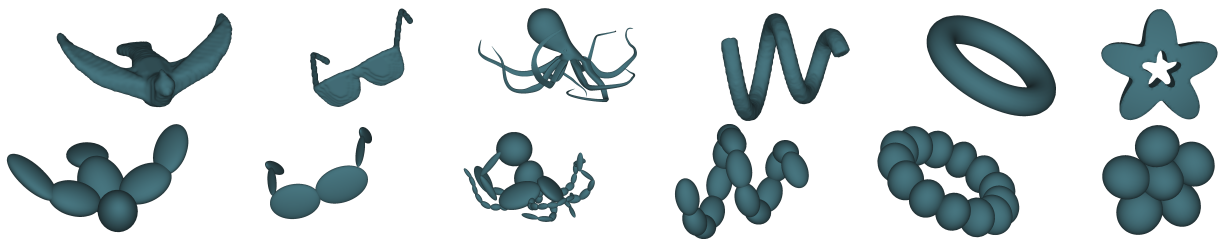


FIGURE 3.11 – Exemples d'objets 3D décomposés en squelette d'ellipsoïdes. Ligne 1 : modèles 3D. Ligne 2 : squelette d'ellipsoïdes 3D.

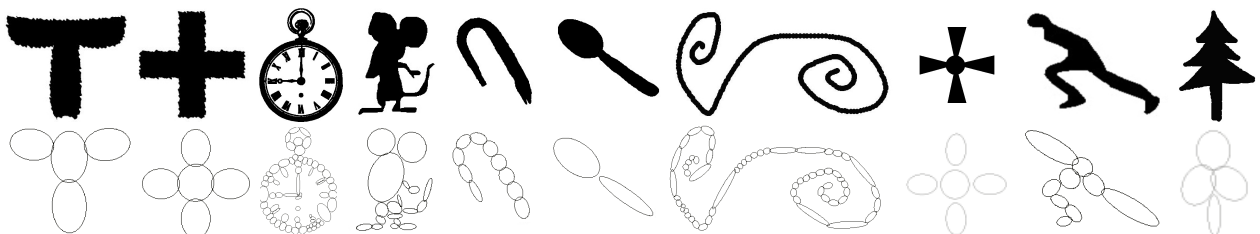


FIGURE 3.12 – Exemples d'objets 2D décomposés en squelette d'ellipsoïdes. Ligne 1 : modèles 2D. Ligne 2 : squelette d'ellipsoïdes 2D.

3.4/ CLASSIFICATION D'UNE BASE DE DONNÉES

La reconnaissance d'un modèle donné Q implique une recherche dans une base de données des modèles O_1, O_2, \dots, O_m les plus proches de Q , c-à-d, les objets similaires. Pour gérer une base de données assez grande, et pour accélérer le processus de recherche en utilisant notre méthode, nous avons développé un nouvel outil pour indexer une base de données. Le principe consiste à répartir les objets de la base de données en différents groupes, selon le nombre de classes k : deux objets avec le même nombre de classes k sont mis dans la même classe. Dans une phase de pré-traitement, tous les objets de la base de données sont transformés en squelette d'ellipsoïdes. Les objets décrits par les squelettes sont répartis en différents groupes. Nous utilisons l'algorithme décrit dans la Section 3.3.3 pour obtenir la meilleure décomposition de la base de donnée en groupes. Nous obtenons n groupes d'objets, dont chaque groupe contient des objets avec approximativement le même nombre k de classes (ellipsoïdes). Pour chercher les objets les plus proches d'un objet Q , il suffit juste de regarder dans la classe contenant des objets avec à peu près le même nombre de classes k .

3.5/ APPARIEMENT DE FORMES

3.5.1/ VECTEUR CARACTÉRISTIQUE D'UNE CLASSE

Afin de calculer les distances entre les objets caractérisés par des squelettes d'ellipsoïdes, nous décrivons chaque classe d'un objet par un vecteur caractéristique. Les éléments d'un vecteur caractéristique associé à une classe (ellipsoïde) C_i sont les trois valeurs propres de la matrice de covariance M_{C_i} de la classe C_i , et la distance euclidienne d entre le centre de gravité de la classe et le centre de gravité de l'objet entier.

Dans le travail présenté par Banégas dans [Banégas et al., 2001], d'autres variantes du vecteur caractéristique d'une classe ont été utilisées. La variante que nous utilisons dans ce travail est la meilleure selon Banégas. En 3D, un vecteur caractéristique est donné par :

$$(\lambda_1, \lambda_2, \lambda_3, d) = (\sqrt{\sigma_1}, \sqrt{\sigma_2}, \sqrt{\sigma_3}, d) \quad (3.9)$$

où vp_i sont les longueurs des axes de l'ellipsoïde E_i (triées dans un ordre décroissant). Les σ_i sont les valeurs propres de M_{C_i} , et d est la distance euclidienne entre le centre de gravité de l'ellipsoïde E_i et le centre de gravité de l'objet.

D'autres vecteurs peuvent être utilisés ; on peut considérer l'orientation des ellipsoïdes, ou seulement considérer les coordonnées du centre de l'ellipsoïde. Cependant, dans le cas où un ellipsoïde est proche d'un disque (les valeurs propres sont toutes égales ou presque), les vecteurs propres deviennent instables. C'est pour cette raison qu'il vaut mieux utiliser les valeurs propres plutôt que les vecteurs propres de l'ellipsoïde.

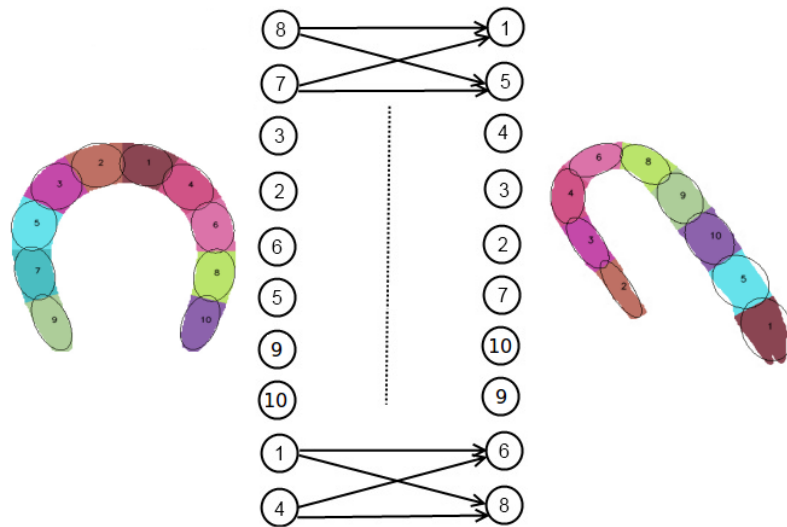


FIGURE 3.13 – Un exemple d'un couplage optimal dans un graphe biparti pour deux formes 2D.

3.5.2/ CALCUL DE DISTANCE : COUPLAGE OPTIMAL DANS UN GRAPHE BIPARTI

Le calcul de similarité de deux formes A et B décrites chacune par leurs squelettes d'ellipsoïdes E_1, E_2, \dots, E_k et E'_1, E'_2, \dots, E'_k est équivalent à un couplage optimal dans un graphe biparti avec $2k$ sommets. Le couplage optimal est résolu par la méthode Hongroise présentée dans [Kuhn, 1955]. Elle est de complexité polynomiale. Les sommets de ce graphe sont les ellipsoïdes des formes A et B (Figure 3.13). Les premiers k sommets représentent les ellipsoïdes de A , et les k derniers, les ellipsoïdes de B . Chaque sommet de A est lié aux k sommets de B , ce qui donne un graphe biparti complet avec k^2 arêtes.

Le poids d'une arête $A_i B_j$ est la distance euclidienne entre les deux vecteurs caractéristiques des ellipsoïdes E_i et E'_j .

Le couplage optimal est le couplage parfait de coût minimal. Un couplage parfait est une bijection entre les sommets de A et ceux de B . Cette bijection, ou couplage parfait, est définie par k arêtes disjointes du graphe biparti complet. Le coût d'un couplage est la somme des poids de ses arêtes. Le couplage optimal est le couplage parfait de coût minimal.

3.6/ RÉSULTATS

Dans une phase de pré-traitement, tous les objets de la base de donnée sont transformés en squelette d'ellipsoïdes en utilisant notre algorithme de partitionnement. Ensuite, nous partitionnons la base de donnée en groupes, de telle sorte que chaque groupe contient des objets avec chacun le même nombre d'ellipsoïdes k ou presque. Les résultats de classification des bases de données 2D et 3D sont illustrés dans le Tableau 3.2. Les résultats illustrés dans le Tableau 3.2 montre une bonne classification de la base de données en utilisant notre méthode.

TABLE 3.2 – Partitionnement de la base de données. Le tableau montre la répartition des objets de la base de données en différents groupes, en utilisant notre algorithme.

Les groupe de la base de données	objets 2D	objets 3D
Group1 ($2 \leq k \leq 4$)	Spoons (95%) Children (93%) Bones (87%)	Airplanes (97%) Dolphins (90%) Pliers (95%)
Group2 ($5 \leq k \leq 10$)	Devices (96%) Snicks (89%)	Human (90%) Teddy (91%)
Group3 ($10 \leq k \leq 20$)	Butterfly(60%) Octopus (57%) Springs (83%)	Ants (75%) Crabs (78%) Dinosaurs (80%)

TABLE 3.3 – Matrice de distances entre des objets similaires. Les valeurs sont petites, contrairement aux distances entre les objets dissimilaires (Tableaux 3.4, 3.5, 3.6.)

	Rat1	Rat2	Rat3	Rat4	Rat5
Rat	0.014	0.026	0.117	0.059	0.077
	Key1	Key2	Key3	Key4	Key5
Key	0.056	0.085	0.12	0.091	0.034
	Bottle1	Bottle2	Bottle3	Bottle4	Bottle5
Bottle	0.028	0.075	0.042	0.033	0.056

TABLE 3.4 – Matrice de distances pour des objets différents (*Fish, Children*)

	Fish1	Fish2	Fish3	Fish4
Childr1	0.406	0.448	0.425	0.497
Childr2	0.439	0.456	0.412	0.446
Childr3	0.482	0.437	0.492	0.465
Childr4	0.497	0.454	0.463	0.431

TABLE 3.5 – Matrice de distances pour des objets différents (*Bone, Bottle*)

	Bone1	Bone1	Bone1	Bone4
Bottle1	0.341	0.368	0.335	0.319
Bottle2	0.322	0.394	0.367	0.331
Bottle3	0.383	0.328	0.372	0.324
Bottle4	0.337	0.341	0.382	0.353

TABLE 3.6 – Matrice de distance pour des objets différents (*Key, Butterfly*)

	Keys1	Keys1	Keys1	Keys4
Butter1	0.637	0.591	0.603	0.654
Butter2	0.622	0.637	0.599	0.601
Butter3	0.642	0.621	0.598	0.615
Butter4	0.626	0.597	0.681	0.663

TABLE 3.7 – La précision du processus de recherche pour des objets 2D et 3D.

2D Objets	précision	objets 3D	précision
Whole	81%	Whole	85%
Bone	87%	Airplanes	96%
Devices	90%	Ants	78.5%
Beetle	78%	Hand	75%
Spring	82 %	Pliers	95%
Spoon	93%	Cups	80%

Pour trouver les objets les plus proches d'une requête, nous comparons cette dernière avec les objets qui se trouvent dans des groupes proches. Les objets sont ordonnés par distance croissante à l'objet de la requête. La figure 3.14 montre les résultats de recherche pour des exemples de requêtes 3D. Le premier élément de chaque ligne représente la requête, et les éléments restants représentent les objets les plus proches de la requête, triés par ordre croissant. La précision du processus de recherche par forme clef est illustrée dans les Tableaux 3.3, 3.4, 3.5, et 3.6. Ces tableaux montrent les résultats du calcul de similarité pour des objets 2D de la base de données MPEG-7. *Les objets sont similaires si la distance qui les sépare est petite (Table 3.3)*

La précision de notre algorithme appliqué aux deux bases de données est illustré dans le tableau 3.7.

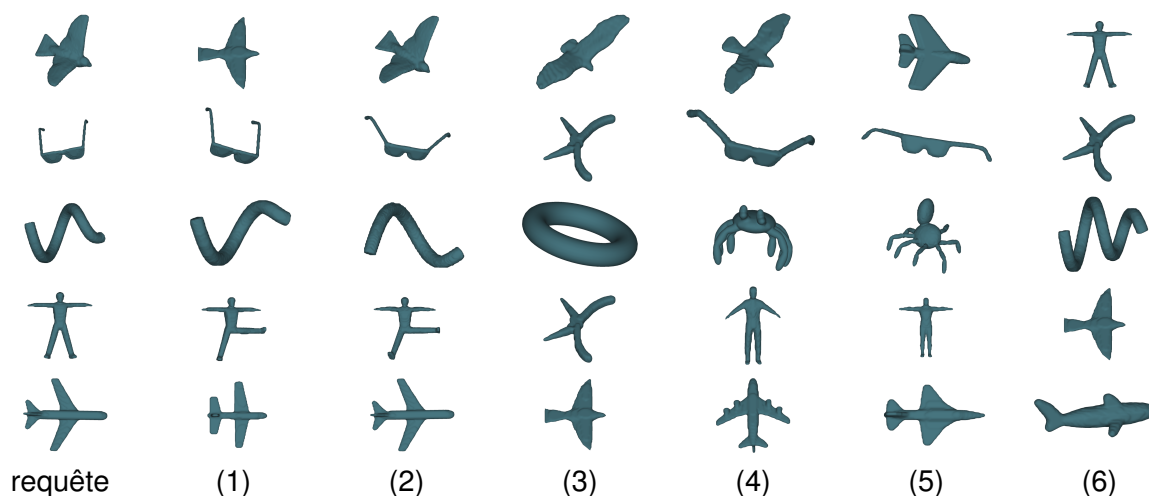


FIGURE 3.14 – Résultats de recherche pour des exemples de requêtes.

3.7/ CONCLUSION

Dans ce chapitre, nous avons décrit notre nouvelle méthode basée sur la décomposition d'objets en squelette d'ellipsoïdes, utilisée pour appairer des objets 2D et 3D. La méthode utilise une nouvelle variante de l'algorithme des nuées dynamiques.

La méthode proposée permet de trouver la décomposition optimale d'un objet en classes ou régions, sans fixer le nombre de classe k au préalable. La méthode est complètement

automatique.

L'utilisation des squelettes d'ellipsoïdes montre une grande résistance aux transformations géométriques tels que les rotations, et le changement d'échelle, et aussi aux aléas de l'échantillonnage.

Cependant, la méthode montre une faiblesse pour des objets articulés. Dans des futurs travaux, des améliorations peuvent être mises en place afin de résoudre ce problème. Une possibilité serait d'utiliser la distance géodésique, la géométrie de diffusion [Coifman et al., 2005], ou encore le partitionnement des graphes [West et al., 2001], lors du calcul des classes d'un objet.

DESCRIPTEURS ET DISTRIBUTION DE FORMES

4.1/ INTRODUCTION

Ce chapitre introduit les différents descripteurs de formes utilisés pour décrire les contributions des chapitres à venir ; nous utilisons différents descripteurs de formes. Ces derniers sont calculés pour chaque sommet d'un graphe associé au maillage représentant un objet 3D. Les valeurs des descripteurs calculées en chaque sommet sont représentées par une distribution, en pratique un histogramme. Nous calculons la distance entre les histogrammes en utilisant des mesures statistiques.

Les notations concernant la représentation des objets 3D sont décrites dans la Section 4.2. Nous décrivons en détails chaque descripteur de formes dans la Section 5.2. Le principe de calcul de dissimilarité entre les distributions est donné dans la Section 4.4.

Étant donné que certains descripteurs utilisés par nos contributions, ainsi que d'autres méthodes existantes utilisées dans les comparaisons, utilisent la géométrie de diffusion et la théorie spectrale des graphes, un aperçu sur la théorie spectrale des graphes et ses concepts figure en annexe.

4.2/ NOTATIONS

Nous définissons maintenant le Graphe \mathcal{G} associé à un maillage 3D.

En effet, un maillage triangulaire \mathcal{M} peut être décrit par un graphe non orienté $\mathcal{G} = (V, E, w)$, où V est l'ensemble des sommets $V = \{v_1, v_2, \dots, v_n\}$ du maillage, avec $n = |V|$, $E = \{e_{ij}\} \subset V \times V$ est l'ensemble des arêtes de \mathcal{M} . Une arête e_{ij} relie deux sommets v_i et v_j si ces deux sommets sont adjacents. A chaque arête $(v_i, v_j) \in E$ est associé un poids, ou coût $w(v_i, v_j) = w(v_j, v_i)$. Le coût entre deux sommets non reliés est égal à zéro ; en fait, il est inutilisé. Nous notons $deg(v_i)$ le degré du sommet v_i : il représente le nombre de voisins du sommet v_i , et $N(v_i) = \{v_j | (v_i, v_j) \in E\}$ le voisinage d'un sommet v_i ; un sommet n'appartient pas à son voisinage.

La matrice d'adjacence A du graphe \mathcal{G} , non orienté, est une matrice symétrique de taille

$n \times n$; elle est définie par :

$$A(i, j) = \begin{cases} 1 & \text{si } (v_i, v_j) \in E \\ 0 & \text{autrement} \end{cases}$$

La matrice des degrés D est une matrice diagonale, dont les éléments de la diagonale sont les degrés $deg(v_i)$ de chaque sommet v_i dans le graphe.

$$D(i, j) = \begin{cases} deg(v_i) & \text{si } i = j \\ 0 & \text{autrement} \end{cases}$$

Nous notons S_i la somme des poids des arêtes incidentes au sommet v_i . S_i est donnée par :

$$S_i = \sum_{v_j \in N(v_i)} w(v_i, v_j) \quad (4.1)$$

\mathcal{D} est une matrice diagonale définie en fonction de S_i :

$$\mathcal{D}(i, j) = \begin{cases} S_i & \text{si } i = j \\ 0 & \text{autrement} \end{cases}$$

La matrice des poids W est définie en fonction des poids (coûts) d'arêtes $w(v_i, v_j)$ reliant les sommets du graphe. La matrice W est définie comme suit :

$$W(i, j) = \begin{cases} w(v_i, v_j) & \text{si } (v_i, v_j) \in E \\ 0 & \text{autrement} \end{cases}$$

Le coût $w(v_i, v_j)$ associé à chaque arête peut être défini en considérant différentes mesures :

- $w(v_i, v_j)$ est la distance euclidienne entre les sommets v_i et v_j .
- Le coût $w(v_i, v_j)$ est défini par le noyau Gaussien :

$$w(v_i, v_j) = \exp(-\|v_i - v_j\|^2 / \sigma^2) \quad (4.2)$$

Le paramètre σ détermine la profondeur du voisinage ; le plus souvent σ est égal à 1 ;

- La moyenne des cotangentes des angles opposés de part et d'autre de l'arête reliant les deux sommets v_i et v_j (Figure 4.1) [Meyer et al., 2003] :

$$w(v_i, v_j) = \frac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij}) \quad (4.3)$$

- La somme des coûts de toutes les arêtes incidentes en un sommet v_i est égale à 1. Ce schéma de coûts est appelé en Anglais "*umbrella weighting*".

Nous pouvons imaginer d'autres coûts tels que :

- L'angle dièdre interne β_{ij} entre les deux faces consécutives autour de l'arête reliant v_i et v_j et la longueur l_{ij} de cette arête :

$$w(v_i, v_j) = \frac{1}{4}l_{ij}(\pi - \beta_{ij}) \quad (4.4)$$

Ce coût va être utilisé pour définir le descripteur de courbure moyenne.

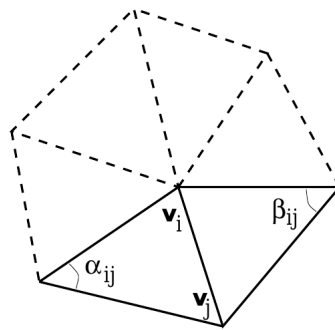


FIGURE 4.1 – Coût défini en termes des angles opposés à l'arête reliant deux sommets voisins [Meyer et al., 2003].

4.3/ DESCRIPTEURS DE FORMES

Dans cette section, nous présentons les différents descripteurs de formes utilisés.

4.3.1/ DESCRIPTEUR DE LA GÉOMÉTRIE DE DIFFUSION (GD)

Ce descripteur est basé sur la chaîne de Markov dans le graphe \mathcal{G} . Il est très résistant aux flexions des objets articulés. Étant donnée la définition d'une marche aléatoire dans un graphe (voir Annexe), nous pouvons déterminer la similarité entre les sommets du graphe, en tenant compte de l'évolution de leur distribution de probabilités. La dissimilarité entre deux sommets v_i et v_j peut donc être calculée en utilisant la décomposition de la matrice de transition en valeurs et vecteurs propres donnée par l'équation (Annexe 4.40). Cette mesure est appelée distance de diffusion ; elle mesure la longueur moyenne des marches aléatoires entre ces sommets. La Figure 4.2 illustre un exemple des chemins reliant deux sommets sur la surface d'un modèle en considérant différents types de distances : distance Euclidienne, distance géodésique, et distance de diffusion.

La distance de diffusion entre deux sommets v et v_j est donnée par la formule suivante :

$$d_{\Delta_t}(v, v_j)^2 = \sum_{k=1}^K \lambda_k^{2\Delta_t} (\psi_k(v) - \psi_k(v_j))^2 = \sum_w (p(t, w|v) - p(t, w|v_j))^2 \quad (4.5)$$

où $p(t, w|v_i)$ et $p(t, w|v_j)$ sont respectivement la probabilité d'atteindre un sommet $w \in V$, en partant des sommets v_i et v_j , en Δ_t transitions ou moins. λ_k et ψ_k sont définis dans l'annexe.

En s'appuyant sur cette mesure de distance, nous pouvons définir les coordonnées Ψ , ou coordonnées intrinsèques (*feature coordinates*), d'un sommet v_i en utilisant les vecteurs propres ; cette transformation est appelée une carte de diffusion (en anglais *diffusion map*) :

$$\Psi_{\Delta_t}(v_i) = (\lambda_1^{\Delta_t} \psi_1(v_i), \lambda_2^{\Delta_t} \psi_2(v_i), \dots, \lambda_k^{\Delta_t} \psi_k(v_i)) \quad (4.6)$$

Ces coordonnées sont intrinsèques, contrairement aux coordonnées cartésiennes qui dépendent du repère cartésien utilisé.

Le descripteur de géométrie de diffusion en un sommet v est donné par la somme des distances de diffusion : un histogramme est construit à partir des valeurs calculées pour tous les sommets.

$$GD(v) = \sum_{i=1}^{n-1} d_{\Delta_i}(v, v_i)^2 \quad (4.7)$$

La Figure 4.3 montre un exemple de carte de couleurs des valeurs associées aux sommets de deux modèles, calculées en utilisant la distance de diffusion entre toutes les paires de sommets sur les surfaces des modèles.

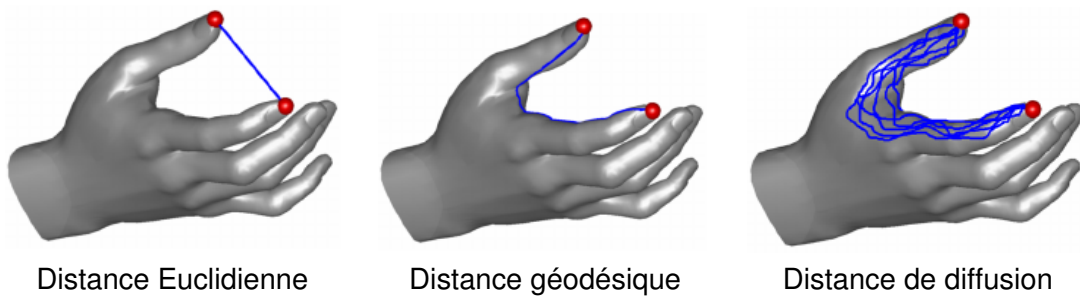


FIGURE 4.2 – Différentes mesures de distance entre deux sommets sur la surface d'un modèle [Dubrovina et al., 2010].

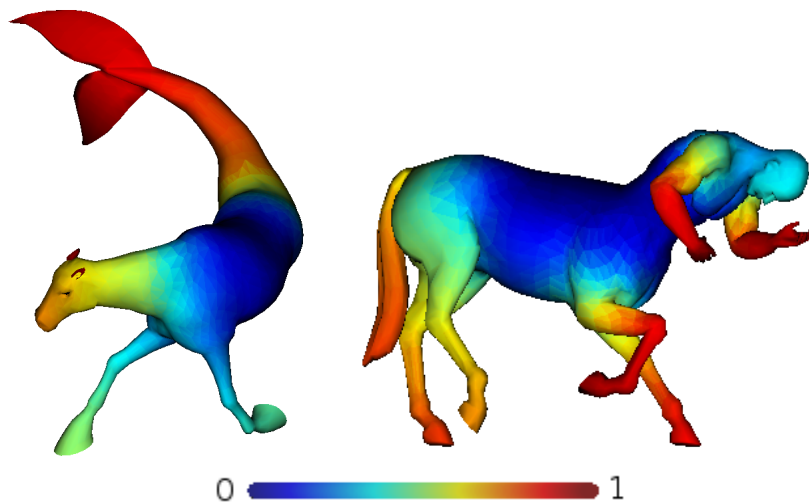


FIGURE 4.3 – Carte de couleurs des valeurs associées aux sommets calculée en utilisant le descripteur de la géométrie de diffusion (GD). En bleu les petites valeurs, et en rouge les grandes valeurs.

4.3.2/ LE DIAMÈTRE LOCAL (LFD)

Le descripteur LFD (*The local diameter*) en un point v de la surface d'une forme représente l'épaisseur locale, ou le diamètre, au point v . Nous notons \vec{N} le vecteur normal (pointé vers l'intérieur) au point v . Le cône de sommet v , d'axe \vec{N} et de semi-angle 60 degrés est échantillonné avec R rayons (par exemple $R = 30$). Pour chaque rayon entrant au sommet v , le premier point sortant w et la longueur de l'intervalle vw sont calculés.

Le descripteur LFD au point v est donc la moyenne pondérée de ces longueurs d'intervalles. Notons au passage que l'histogramme de ces longueurs pourrait être un autre descripteur possible. Ce descripteur montre une grande robustesse aux changements de poses.

4.3.3/ LE DESCRIPTEUR DE L'ÉNERGIE DISCRÈTE (DE)

Dans la littérature, plusieurs descripteurs de forme locaux ont été proposés. P.Heider *et al.* [Heider et al., 2011] donnent un état de l'art des descripteurs de forme locaux. Ils montrent que les descripteurs basés sur les courbures sont les plus efficaces.

Le descripteur de l'énergie discrète (*Discrete Energy*), ou l'énergie de déformation discrète, est basé sur des informations géométriques locales d'une forme. Il calcule la courbure en chaque sommet. Ce descripteur $DE(v)$ d'un sommet v est donné par :

$$DE(v) = 4H(v)^2 - 2K(v) \quad (4.8)$$

où $H(v)$ et $K(v)$ sont respectivement les descripteurs DMC et DGC définis ci-dessous.

Dans la modélisation géométrique, il est fréquent de minimiser l'ED du maillage triangulaire, afin d'obtenir une surface linéaire par morceaux élégante (*fair*) [Farin, 2002].

— Courbure Gaussienne (DGC)

Le descripteur DGC (*Discrete Gaussian Curvature*) [Meyer et al., 2003] en un sommet v est donné par :

$$DGC(v) = \frac{2\pi - \sum_{i=1}^{deg(v)} \theta_i}{\frac{1}{3}a_v} \quad (4.9)$$

où θ_i est l'angle entre deux arêtes consécutives incidentes en v , a_v est l'aire des triangles incidents en le sommet v (voisins directs), et $deg(v_i)$ le degré du sommet v .

— Courbure moyenne (DMC)

La courbure moyenne DMC (*Discrete Mean Curvature*) en un sommet v est donnée par :

$$DMC(v) = \frac{1}{4} \sum_{i=1}^{deg(v)} l_i(\pi - \beta_i) \quad (4.10)$$

où $deg(v)$ est le degré du sommet v , β_i est l'angle dièdre interne entre deux faces consécutives incidentes au sommet v , et l_i est la longueur de l'arête commune à ces deux faces.

$\sum_v DMC(v)$ a une certaine similitude avec la mesure de Dehn $\sum_e l(e)(\theta(e) + \mathbb{Z}\pi)$, où e est une arête, $l(e)$ sa longueur Euclidienne et $\theta(e)$ son angle dièdre interne exprimé en radians. En réponse au troisième problème de Hilbert, deux polyèdres sont congruents si ils ont le même volume et la même mesure de Dehn, c-à-d qu'il est possible de couper l'un des deux en un nombre fini d'éléments polyédriques et de rassembler les morceaux pour obtenir l'autre polyèdre.

4.3.4/ DESCRIPTEUR DE LA NORMALE EN UN SOMMET (VND)

Ce descripteur (*The Vertex Normal-based Descriptor*) est simple, rapide, et basé sur une idée simple. Il calcule le vecteur normal aux sommets. Le vecteur normal \vec{N} (normalisé) en un sommet v est la moyenne des vecteurs normaux (normalisés) dans le voisinage du sommet :

$$\vec{N}(v) = \frac{1}{l} \sum \alpha_f \vec{N}_f \quad (4.11)$$

où l est le nombre de faces qui entourent le sommet v , et α_f est un coefficient associé à une face : c'est le rapport entre l'aire de f et la somme des aires des faces incidentes en v .

Le vecteur normal \vec{N}_f d'une face f avec trois points p_1, p_2 et p_3 , est donné par :

$$\vec{N}_f = (p_2 - p_1) \times (p_3 - p_1) \quad (4.12)$$

$p_i = (x_i, y_i, z_i)$, $i = 1, 2, 3$, et \times représente le produit vectoriel. Ensuite N_f est normalisé.

L'orientation de \vec{N}_f n'a pas d'importance, mais on peut supposer que p_1, p_2, p_3 tournent dans le sens horaire, de sorte que \vec{N}_f est vers l'extérieur. Le descripteur VND en un sommet $v \in F$ est donné par :

$$VND(v) = \frac{\|\vec{N}(v)\|}{\sum_{v' \in F} \|\vec{N}(v')\|} \quad (4.13)$$

où F est l'ensemble des points échantillonnés et $\|\cdot\|$ est la longueur euclidienne.

4.3.5/ LA DISTRIBUTION LOCALE DE LA FORME (LSD)

Le descripteur LSD (*Local Shape Distribution*) [Li et al., 2012] extrait n sommets aléatoires, et caractérise chaque sommet v en termes des distances euclidiennes à tous les autres sommets appartenant à son voisinage. Le voisinage du sommet v est une zone sphérique centrée au point v . Les descripteurs LSD associés à chaque région sont représentés par des histogrammes.

Pour calculer la dissimilarité entre deux formes A et B , un graphe biparti complet g est construit comme suit : le premier ensemble de sommets du graphe g est donné par les régions de A , le deuxième ensemble est donné par les régions de B . Le coût d'une arête (a, b) entre deux régions de g est la distance du χ^2 entre l'histogramme de a et celui de b . Le couplage parfait de coût minimal dans g est calculé avec, par exemple, la méthode

Hongroise [Kuhn, 1955] : un couplage optimal est une bijection entre les deux ensembles de sommets, et le coût d'une mise en correspondance est la somme des coûts des arêtes utilisées dans la correspondance. Par définition, la distance entre A et B est le coût de l'appariement optimal.

Un avantage de cette méthode est qu'elle ne calcule pas seulement une distance entre deux formes A et B , mais elle calcule aussi la correspondance entre les régions de A et les régions de B .

4.3.6/ LE DESCRIPTEUR DE DISTRIBUTION DE TEMPÉRATURE (TD)

Le descripteur de distribution de température (*Temperature Distribution*) simule le processus de diffusion de la chaleur sur la surface d'un modèle : une unité de chaleur, ou calorie, est appliquée en un sommet, et elle se diffuse aux autres sommets au cours du temps. Ce descripteur [Fang et al., 2011] en un sommet est la moyenne des températures mesurées sur tous les sommets de la surface du modèle. La température moyenne pour un sommet v , au temps t de la dissipation de la chaleur, est donnée par :

$$TD(v) = \frac{1}{N-1} \sum_{w, w \neq v} \sum_i^K e^{-\lambda_i t} \phi_i(v) \phi_i(w) \quad (4.14)$$

où N est le nombre de sommets, t est une constante, et λ_i est le i ème valeur propre de l'opérateur Laplacien du graphe associé au maillage (voir Annexe), et ϕ_i le i ème vecteur propre correspondant. En pratique, seuls quelques vecteurs propres sont utilisés ($K = 20$), ceux correspondant aux valeurs propres de plus grand module.

La répartition des valeurs de la température moyenne est représentée par un histogramme. La distance entre deux formes est la norme L_2 calculée entre leurs histogrammes.

Le descripteur de TD est invariant par transformations isométriques comme les changements de pose, et résistant au bruit d'échantillonnage.

4.3.7/ DESCRIPTEUR DE DISTRIBUTION (D2)

Le descripteur D2 est global [Osada et al., 2001]. Ce descripteur échantillonne n points sur la surface de l'objet, et associe à chaque point une valeur numérique en utilisant une mesure donnée. La mesure peut être la distance au centre de gravité de l'objet, l'angle formé entre trois points, etc. Ces valeurs sont ensuite transformées en une distribution ou histogramme. La distance entre deux objets est la distance entre leurs histogrammes associés. Ce descripteur est très simple et donne de bons résultats pour des objets rigides.

4.3.8/ LE DESCRIPTEUR DE L'ENVELOPPE CONVEXE (CH)

L'enveloppe convexe CH (*Convex Hull*) est une idée simple. Elle est calculée en utilisant différentes méthodes telles que *Quickhull* [Barber et al., 1996], *Gift wrapping*

[Dixon, 1993], etc. Dans nos expériences, nous avons utilisé l'algorithme *Gift wrapping* pour calculer l'enveloppe convexe d'une forme.

Ce descripteur [Kalyanaraman et al., 2005] est un histogramme des distances euclidiennes entre les sommets de l'enveloppe convexe d'un modèle 3D.

La Figure 4.4 montre la carte de couleurs des valeurs associées aux sommets, calculées en utilisant les différents descripteurs ; les petites valeurs sont représentées en bleu, et les grandes valeurs en rouge.

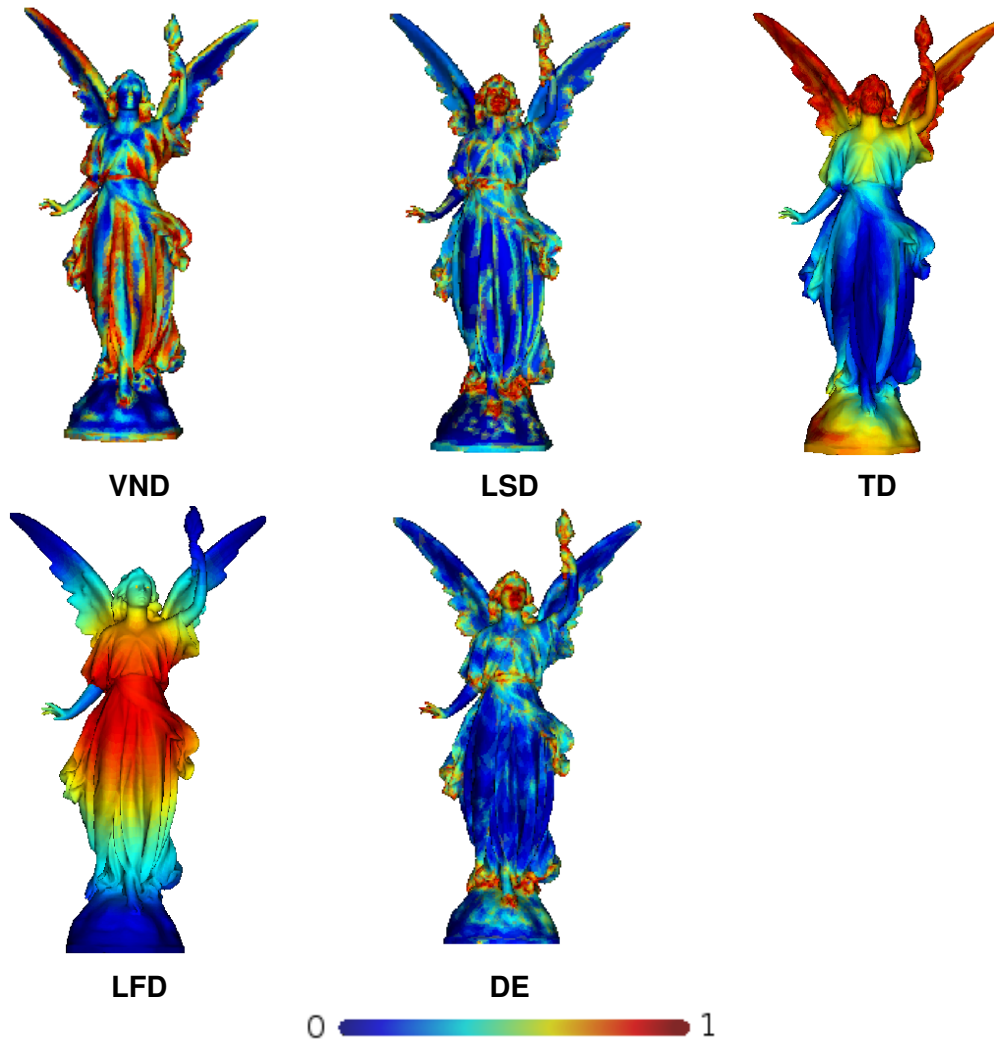


FIGURE 4.4 – Carte de couleurs des valeurs associées aux sommets calculées en utilisant les différents descripteurs

4.4/ MESURE DE DISSIMILARITÉ : DISTANCE

Nous calculons un descripteur donné (par exemple le descripteur D2) en chaque sommet de la surface d'un modèle. Après une phase de normalisation, la valeur du descripteur en chaque sommet échantillonné est comprise entre 0 et 1. Pour chaque modèle, un histogramme H de $b \approx \sqrt{n}$ tiroirs est construit à partir des n valeurs du descripteur calculé

pour les n sommets extraits de la surface de l'objet. L'histogramme est défini par :

$$H(i) = \frac{m_i}{n}, \quad i = 0, 1, \dots, b - 1. \quad (4.15)$$

où m_i est le nombre de points dont la valeur du descripteur est dans l'intervalle $[i/b, (i + 1)/b]$.

La distance ou la dissimilarité d de deux formes A et B est calculée en utilisant la loi du χ^2 entre leur histogrammes H_1 et H_2 . Elle est donnée par :

$$d(A, B) = \chi^2(H_1, H_2) = \frac{1}{2} \sum_{j=0}^{b-1} \frac{(H_1(j) - H_2(j))^2}{H_1(j) + H_2(j)} \quad (4.16)$$

$\forall i, 0 \leq H_1(i) \leq 1$ et $\sum_j H_1(j) = 1$, et $\forall i, 0 \leq H_2(i) \leq 1$ et $\sum_j H_2(j) = 1$, donc χ^2 appartient à l'intervalle $[0, 1]$.

4.5/ CONCLUSION

Dans ce chapitre, nous avons décrit les différents descripteurs de formes utilisés par les contributions des chapitres qui suivent. Nous utilisons une variété de descripteurs sélectionnés parmi des catégories différentes. Les descripteurs choisis sont simples et efficaces.

Un descripteur donné est calculé en chaque sommet du graphe sous-jacent à un maillage triangulaire. Les valeurs calculées en ces sommets sont ensuite transformées en un histogramme (distribution discrète). La distance entre deux formes A et B représentées respectivement par leur histogramme H_1 et H_2 est de la loi du χ^2 calculée entre ces histogrammes.

Nous donnons dans l'annexe un aperçu général sur la décomposition spectrale des graphes, utilisée dans la définition de certains descripteurs.

Dans les chapitres à venir, nous présenterons nos contributions au domaine de l'appariement de formes et de la recherche par forme clef, en utilisant les différents descripteurs de formes décrits dans ce chapitre.



ANNEXE

4.A/ LA DÉCOMPOSITION SPECTRALE DES MAILLAGES 3D

La décomposition spectrale, ou théorie spectrale des graphes, est une branche de la théorie algébrique des graphes. Elle s'intéresse à l'étude des valeurs propres et des vecteurs propres des matrices associées aux graphes : les matrices Laplaciennes.

4.A.1/ COORDONNÉES DIFFÉRENTIELLES ET MATRICE LAPLACIENNE

Il est parfois difficile d'effectuer des opérations sur les maillages, en utilisant les coordonnées cartésiennes pour décrire la géométrie de la surface. Dans certains cas, il est indispensable de garder une trace sur la structure des sommets du maillage, lorsque certaines transformations (par exemple des déformations) sont effectuées sur le maillage. La matrice Laplacienne du graphe est utilisée pour conserver ces informations sur la structure du maillage.

En effet, les coordonnées différentielles [Do Carmo et al., 1976] (*δ-coordinates*) représentent une alternative aux coordonnées cartésiennes pour représenter la géométrie du maillage, et stocker la structure de ce dernier.

Les coordonnées différentielles d'un sommet v_i représentent la différence entre ses coordonnées cartésiennes et les coordonnées du centre de gravité de ses voisins.

$$\delta_{v_i} = (\delta_{v_i}^{(x)}, \delta_{v_i}^{(y)}, \delta_{v_i}^{(z)}) = v_i - \frac{1}{deg(v_i)} \sum_{j \in N(v_i)} v_j \quad (4.17)$$

Le passage des coordonnées cartésiennes aux coordonnées différentielles (*δ-coordinates*) peut être représenté sous forme d'une matrice L . Il existe deux variantes de cette matrice : $L = I - D^{-1}A$ ou $L = I - \mathcal{D}^{-1}W$. Notez que dans le premier cas ($L = I - D^{-1}A$), la matrice L est une matrice associée à un graphe non pondéré, c-à-d, le coût de toutes les arêtes dans le graphe est égal à 1. Il est plus commode de considérer la version symétrique de la matrice L , qui est la matrice $L = D - A$ (ou $L = \mathcal{D} - W$), appelée *la Matrice Laplacienne* du graphe ; elle est définie dans le premier cas comme suit :

$$L(i, j) = \begin{cases} deg(v_i) & \text{si } i = j \\ -1 & (v_i, v_j) \in E \\ 0 & \text{autrement} \end{cases}$$

et selon le deuxième cas ainsi :

$$L(i, j) = \begin{cases} S_i & \text{si } i = j \\ -w(v_i, v_j) & (v_i, v_j) \in E \\ 0 & \text{autrement} \end{cases}$$

Pour passer des coordonnées cartésiennes aux coordonnées différentielles, et réciproquement, nous utilisons le système d'équations suivant (à gauche le premier cas, à droite le second cas) :

$$LX = D\delta^{(x)}, \quad LX = \mathcal{D}\delta^{(x)} \quad (4.18)$$

$$LY = D\delta^{(y)}, \quad LY = \mathcal{D}\delta^{(y)} \quad (4.19)$$

$$LZ = D\delta^{(z)}, \quad LZ = \mathcal{D}\delta^{(z)} \quad (4.20)$$

où X, Y, Z sont des vecteurs de taille n contenant respectivement les coordonnées x, y , et z de tous les sommets.

4.A.2/ MATRICE LAPLACIENNE D'UNE MARCHÉ ALÉATOIRE

Nous pouvons aussi associer à un graphe \mathcal{G} une matrice de transition P correspondant à une marche aléatoire dans le graphe \mathcal{G} . Dans le cas d'un graphe non pondéré :

$$P = D^{-1}A \quad (4.21)$$

tels que :

$$P(i, j) = \begin{cases} 0 & \text{si } i = j \\ \frac{1}{\text{deg}(v_i)} & (v_i, v_j) \in E \\ 0 & \text{autrement} \end{cases}$$

Dans le cas d'un graphe pondéré, c-à-d en considérant la matrice \mathcal{D} , la matrice P est donnée par :

$$P = \mathcal{D}^{-1}W \quad (4.22)$$

où :

$$P(i, j) = \begin{cases} 0 & \text{si } i = j \\ \frac{w(v_i, v_j)}{S_i} & (v_i, v_j) \in E \\ 0 & \text{autrement} \end{cases}$$

$P(i, j)$ représente la probabilité de transition d'un sommet v_i vers un sommet v_j .

La matrice Laplacienne L_r correspondant à la matrice des probabilités de transition P est donnée par :

$$L_r = I - P \quad (4.23)$$

La matrice L_r n'est pas symétrique, dû au fait que $P_{i,j} \neq P_{j,i}$, mais il est possible de la remplacer par une matrice symétrique appelée *matrice Laplacienne de la marche aléatoire* L :

$$L = D^{\frac{1}{2}} P D^{-\frac{1}{2}} \quad (4.24)$$

4.A.3/ LES VALEURS ET LES VECTEURS PROPRES D'UNE MATRICE LAPLACIENNE

Une matrice symétrique L peut être exprimée en terme de ses valeurs propres λ_i , et de leurs vecteurs propres ϕ_i correspondants, tel que :

$$L\phi = \lambda\phi \quad (4.25)$$

Pour calculer les valeurs propres λ_i , nous considérons le système d'équations $|L_i - \lambda I| = 0$. Ensuite les vecteurs propres ϕ_i sont calculés en résolvant le système d'équation $L_i\phi_i = \lambda_i\phi_i$.

Étant donnée φ une matrice dont les colonnes contiennent les vecteurs propres ϕ_i , et ψ est une matrice diagonale dont les éléments diagonaux sont les valeurs propres λ_i , l'équation 4.25 peut être écrite comme suit :

$$L\varphi = \varphi\psi \quad (4.26)$$

En multipliant à droite par φ^T , et en utilisant les valeurs et les vecteurs propres de la matrice L , on obtient la décomposition spectrale de la matrice L :

$$L = L\varphi\varphi^T = \varphi\psi\varphi^T = \sum_{k=1}^K \lambda_k \phi_i^{(k)} (\phi_i^{(k)})^T \quad (4.27)$$

où K représente le nombre de valeurs propres, et $i = \{1, 2, \dots, n\}$ les indices des sommets. $\phi_i^{(k)}$ représente la valeur du vecteur propre $\phi^{(k)}$ à l'indice i .

Les valeurs propres sont réelles et triées par ordre croissant tel que : $0 = |\lambda_0| \leq |\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_{K-1}|$, $k \ll n$. Les vecteurs propres $\Phi_n = \{\phi^{(1)}, \phi^{(2)}, \dots, \phi^{(K)}\}$ forment une base orthogonale dans \mathbb{R}^n . Chaque vecteur propre est un tableau de n éléments, dont les éléments représentent les valeurs du vecteur propre en tous les sommets du graphe : $\phi_k = (\phi_{v_1}^k, \phi_{v_2}^k, \dots, \phi_{v_n}^k)$.

En effet, les vecteurs et les valeurs propres d'un graphe sont utilisés pour trouver une représentation compacte, permettant de passer d'un espace de grande dimension (graphe $n \times n$) à un espace réduit, en l'occurrence un espace vectoriel. Ce plongement, ou réduction d'espace, donne une visualisation de données dans un espace non Euclidien (un vecteur de taille n). Il est effectué en utilisant le vecteur propre correspondant à la plus petite valeur propre non nulle. Pour des objets 3D, représentés sous forme de graphes, cette valeur est représentée sous formes de carte de couleurs, représentant les valeurs du vecteur propre en chaque sommet du graphe. Une couleur est associée à chaque sommet selon la valeur du vecteur propre en ce sommet. Les couleurs sont graduées du bleu (petite valeur) au rouge (grande valeur). La Figure 7.4 montre un exemple de carte de couleurs calculée pour un modèle donné, en utilisant les différents vecteurs propres ϕ_i de la matrice Laplacienne L du graphe correspondant au modèle. Les couleurs associés aux sommets du graphe \mathcal{G} correspondent aux valeurs des vecteurs propres en ces sommets.

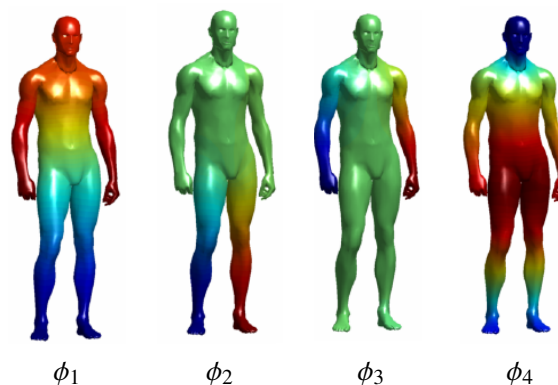


FIGURE 4.5 – Carte de couleurs représentant les valeurs et les vecteurs propres de la matrice Laplacienne [Dubrovina et al., 2010].

Cette représentation permet de comprendre la géométrie des surfaces, et la façon dont les sommets sont reliés entre eux. La première valeur propre non-nulle λ_1 (en anglais appelée *Fidler value*) donne une indication sur la connectivité algébrique du graphe ; plus cette valeur est grande, plus le graphe est connecté. Si tous les sommets du graphe sont connectés (graphe complet), alors la valeur propre est égale à 1. Quand au vecteur propre correspondant à cette valeur propre (en anglais appelé *Fidler vector*), il permet de donner un ordre aux sommets dans le graphe selon l'axe de l'objet, verticalement pour un être humain.

4.B/ MARCHE ALÉATOIRE ET GÉOMÉTRIE DE DIFFUSION

La géométrie de diffusion consiste à évaluer un processus de diffusion sur la surface d'un modèle ; cela peut être vu comme une sorte d'énergie qui se propage le long de la surface. A chaque instant, cette quantité se diffuse d'une façon aléatoire à partir d'un sommet, pour atteindre les sommets voisins à l'instant suivant.

Dans la littérature, la plupart des études qui s'intéressent à la géométrie de diffusion, montrent qu'elle est liée à l'étude d'une marche aléatoire dans un graphe, en utilisant ses matrices associées. Une marche aléatoire dans un graphe est un processus aléatoire qui commence en un sommet, et qui passe aux autres sommets au cours du temps. Le passage d'un sommet à un autre dépend du type de graphe ; pour un graphe non pondéré, les sommets suivants sont choisis d'une façon uniforme (la même probabilité de transition), tandis que pour un graphe pondéré, la probabilité de transition est proportionnelle au poids de l'arête.

Nous présentons dans cette section le processus de marche aléatoire en utilisant les valeurs et les vecteurs propres des matrices associées à un graphe, et son utilisation dans la géométrie de diffusion.

4.B.1/ CHAÎNE DE MARKOV

Les modèles aléatoires sont souvent utilisés pour l'évolution au cours du temps d'un système dynamique, à titre indicatif les phénomènes naturels. Les chaînes de Markov sont largement utilisées comme modèle aléatoire pour étudier ces systèmes dynamiques. Une chaîne de Markov à temps discret est un processus stochastique (aléatoire) représentant une évolution dynamique d'un système aléatoire dans le temps. La prédiction de l'état futur ne dépend que de son état présent. Donnons maintenant quelques définitions.

Le processus Markovien est une séquence (suite) de variables aléatoires $(X_1, X_2, \dots, X_n, n \in \mathbb{N})$, dont chaque variable aléatoire X_i possède une valeur dans l'espace des états, autrement dit, X_i ($i = 1, 2, \dots, n$) représente l'état du processus à l'instant i . Les variables aléatoires représentent donc l'état du processus au cours du temps.

Soit $E = \{x_1, x_2, \dots, x_n\}$ l'espace des états du processus.

Definition 5. Une suite de variables aléatoires X_i ($1 \leq i < n$), pour tout $n \in \mathbb{N}$ à valeur dans E est une chaîne de Markov à état discret si pour $x_i \in E$, on a :

$$\mathbb{P}(X_{i+1} = x_v | X_i = u, X_{i-1}, \dots, X_0) = \mathbb{P}(X_{i+1} = x_v | X_i = x_u) \quad (4.28)$$

où $u, v \in \{1, 2, \dots, n\}$.

$\mathbb{P}(X_{i+1} = x_v | X_i = x_u)$ représente la probabilité de transition de l'état x_u à l'état x_v en un pas. Pour des raisons de simplicité, on note cette probabilité par $P_{u,v}$.

La matrice stochastique $M = (P_{u,v})$ de taille $n \times n$ est appelée la matrice de transition de la chaîne de Markov. Elle vérifie :

$\forall u \in \{1, 2, \dots, n\} \sum_{v=1}^n P_{u,v} = 1$, c-à-d, la somme des éléments de chaque ligne est égale

à 1. Cette matrice peut être représentée par un graphe probabiliste, dont les sommets représentent les états de la chaîne de Markov, et les arcs les probabilités de transition. Donnons un exemple d'une chaîne de Markov représentée par un graphe probabiliste et sa matrice de transition associée, en considérant que toutes les arêtes ont la même probabilité d'être empruntées.

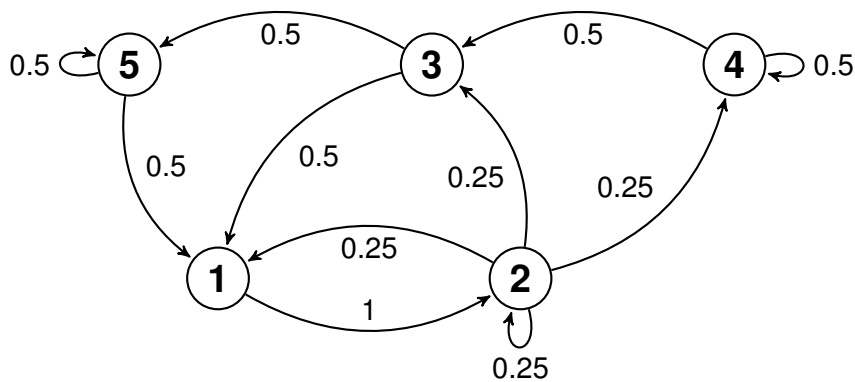


FIGURE 4.6 – Graphe G_1 associé à une chaîne de Markov avec les mêmes probabilités pour les arêtes sortantes d'un sommet donné.

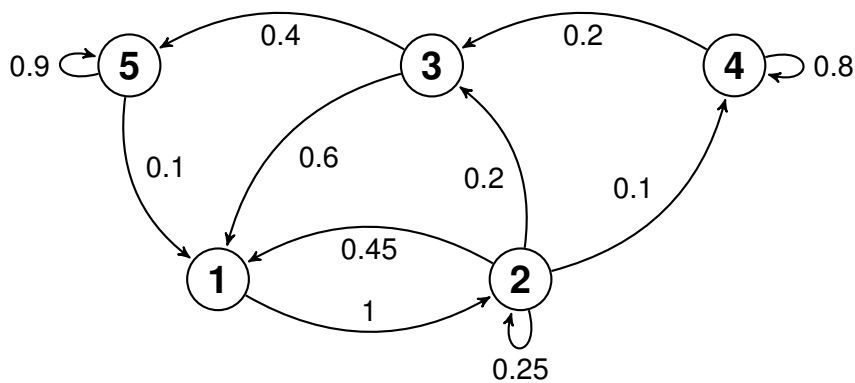


FIGURE 4.7 – Graphe G_2 associé à une chaîne de Markov, considérant des probabilités distinctes pour les arêtes sortantes d'un sommet donné.

$$M_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0.25 & 0.25 & 0.25 & 0.25 & 0 \\ 0.5 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0 & 0.5 \end{bmatrix} \tag{4.29}$$

$$M_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0.45 & 0.25 & 0.2 & 0.1 & 0 \\ 0.6 & 0 & 0 & 0 & 0.4 \\ 0 & 0 & 0.2 & 0 & 0.8 \\ 0.1 & 0 & 0 & 0 & 0.9 \end{bmatrix} \quad (4.30)$$

Dans un graphe probabiliste, une marche aléatoire représente une séquence aléatoire de transition entre les sommets $(1, 2, \dots, n, n \in \mathbb{N})$ du graphe. Elle peut être vue comme un point qui se déplace aléatoirement sur le graphe, en empruntant les arcs reliant les sommets. Nous partons d'un point (sommet) initial u , et nous choisissons aléatoirement un voisin direct v , et après nous choisissons de la même façon un autre sommet w à partir de v , et ainsi de suite. Pour emprunter un arc en allant d'un sommet vers un de ses sommets voisins, nous distinguons deux cas de figures : la probabilité de transition $P_{u,v}$ est la même pour tous les arcs entrants au sommet u : $P_{u,v} = 1/d(v), \forall u \in N(v)$; Les arcs sont empruntés aléatoirement, et en tenant compte des probabilités distinctes associées aux arêtes.

Les figures 4.6, 4.7 et leurs matrices associées 4.29, 4.30 représentent deux exemples de chaînes de Markov, en tenant compte des deux types de probabilités associées aux arcs.

En effet, l'évolution au cours du temps d'une Chaîne de Markov d'espace d'états E , et de matrice de transition M est donnée par :

$$\pi_{t+1} = \pi_t M = \pi_0 M^{t+1} \quad (4.31)$$

où π_t est le vecteur ligne des probabilités de chaque état à l'instant t .

La matrice M élevée à la puissance k représente les probabilités de transitions entre les différents état en k intervalles de temps (étapes). En général, la chaîne de Markov est ergodique, c-à-d qu'il existe une unique distribution stationnaire π^* tel que $\pi^* = \pi^* M$. quelque soit la probabilité de distribution de probabilités initiale π_0 , $\pi_0 M$ tant vers π^* quand k tend vers l'infini.

Notons que $\pi_0 = \{p_1, p_2, \dots, p_n\}$ est un vecteur donnant la probabilité p_i d'être en chaque état i de la chaîne de Markov à l'instant 0, et la somme des p_i égale 1. La vitesse de convergence vers π^* est déterminée par la seconde valeur propre en module, λ_2 .

4.B.2/ CHAÎNE DE MARKOV D'UN MAILLAGE 3D

Le graphe \mathcal{G} donne une chaîne de Markov ; les sommets de la chaîne de Markov sont les sommets du graphe. Chaque arête du graphe AB donne deux arcs $A \rightarrow B$ et $B \rightarrow A$ dans la chaîne de Markov. Nous notons $P(v_i, v_j)$ la probabilité de transition entre le sommet, ou état, v_i , et le sommet v_j . Quand une arête lie v_i et v_j dans le graphe, la probabilité de transition de v_i à v_j de la chaîne de Markov est donnée par l'équation (4.32), sinon elle est nulle.

$$P(X(t+1) = v_j \mid X(t) = v_i) = \frac{w(v_i, v_j)}{\sum_{v_k \in N(v_i)} w(v_i, v_k)} \quad (4.32)$$

$\sum_{v_j \in N(v_i)} P(v_i, v_j) = 1$, donc la matrice P est stochastique.

La probabilité associée à une arête peut être vue comme la conductance ou la résistance de l'arête.

En conséquence, la matrice des probabilités de transition, appelée aussi la matrice Laplacienne normalisée de la chaîne de Markov, P , peut être définie par l'équation (4.33) :

$$P = \mathcal{D}^{-1}W \quad (4.33)$$

où les éléments $P(i, j)$ de cette matrice représentent les probabilités de la transition d'une marche aléatoire d'un sommet v_i vers le sommet v_j en un pas.

Nous notons $p(t, v | u)$ la probabilité d'atteindre un sommet v , en partant d'un sommet u en t déplacements de la marche aléatoire.

$$p(t, v | u) = \pi_0 P^t \quad (4.34)$$

où π_0 est la distribution de probabilité initiale.

Remarquons qu'en général, la matrice P n'est pas symétrique : $P(v_i, v_j) \neq P(v_j, v_i)$. Nous remplaçons cette matrice P par la matrice P_s définie par :

$$P_s = \mathcal{D}^{\frac{1}{2}} P \mathcal{D}^{-\frac{1}{2}} = \mathcal{D}^{-\frac{1}{2}} W \mathcal{D}^{-\frac{1}{2}} \quad (4.35)$$

Comme W est symétrique, et que \mathcal{D} est diagonale, $P_s = \mathcal{D}^{\frac{1}{2}} P \mathcal{D}^{-\frac{1}{2}} = \mathcal{D}^{-\frac{1}{2}} W \mathcal{D}^{-\frac{1}{2}}$ est aussi symétrique ; P_s est une matrice diagonalisable de taille n par n , qui a les mêmes valeurs propres $\{\lambda_i\}_{i=1}^n$ que la matrice P . Les vecteurs propres correspondants $\{\phi_i\}_{i=1}^n$ forment une base orthogonale dans \mathbb{R}^n .

$$P_s \phi_i = \lambda_i \phi_i \quad (4.36)$$

Comme les ϕ_i sont des vecteurs propres orthogonaux pour le produit scalaire canonique de \mathbb{R} , la matrice P_s admet deux familles de vecteurs bi-orthogonaux, un vecteur ligne ψ_i , et un vecteur colonne φ_i . Ils se définissent en fonction des vecteurs propres ϕ_i de P [Nadler et al., 2005], par l'équation (4.37) :

$$\varphi_i = \phi_i \mathcal{D}^{\frac{1}{2}}, \quad \psi_i = \phi_i \mathcal{D}^{-\frac{1}{2}} \quad (4.37)$$

Les valeurs propres réelles (car P_s est symétrique réelle) λ_i sont triées en ordre décroissant $\lambda_0 = 1 \geq |\lambda_1| \geq \dots \geq |\lambda_n|$.

Le premier vecteur propre à gauche φ_0 correspondant à la valeur propre $\lambda_0 = 1$ est (à un scalaire multiplicatif près) la distribution stationnaire de probabilité de la chaîne de Markov du graphe G .

$$\lim_{t \rightarrow \infty} p(t, v | u) = \varphi_0(v) \quad (4.38)$$

En pratique, ces chaînes de Markov sont toujours irréductibles et non périodiques, et ont donc une seule distribution stationnaire de probabilité, dans ce cas φ_0 :

$$\varphi_0 = \varphi_0 P \quad (4.39)$$

Les probabilités de distribution peuvent être exprimées en utilisant les valeurs et les vecteurs propres de la matrice de transition P . La probabilité pour une marche aléatoire, qui commence en un sommet v_i au temps t , d'atteindre un sommet v_j du graphe au temps $t + \Delta t$, est :

$$P(X(t_0 + \Delta t) = v_j \mid X(t_0) = v_i) = \varphi_0(v_j) + \sum_{k=1}^K \psi_k(v_i) \lambda_k^{\Delta t} \varphi_k(v_j) \quad (4.40)$$

$\psi_k(\varphi_k)$ est la valeur du k i ème vecteur propre à droite (respectivement à gauche) au sommet, ou état v_i (respectivement v_j). K est le nombre de valeurs propres prises en compte. En pratique, $\Delta t = 40$, et seulement quelques valeurs sont prises ($K = 20 \ll n$) les plus grandes en valeur absolue. Le logiciel Scilab [Scilab Enterprises, 2012] peut être utilisé pour calculer les valeurs et vecteurs propres. Il est souhaitable d'utiliser une représentation creuse pour W : $W_{i,j}$ n'est pas mémorisé quand l'arête (v_i, v_j) n'existe pas dans le graphe. Le nombre d'arêtes du graphe est proportionnel au nombre de sommets. Un vecteur suffit pour stocker la matrice diagonale \mathcal{D} .

LA FUSION DES DESCRIPTEURS PAR LA RÈGLE DU PRODUIT

5.1/ INTRODUCTION

La majorité des méthodes d'appariement et de recherche de formes utilise un seul descripteur de forme pour représenter les objets 3D. Cependant, aucun descripteur de forme n'est assez puissant pour fournir des résultats appropriés pour toutes les formes. La technique la plus courante utilisée pour améliorer la performance du processus de recherche consiste à fusionner les descripteurs. Dans ce chapitre, nous proposons une nouvelle approche de fusion utilisée pour l'amélioration de la précision du processus de recherche par forme clef. Cette méthode est basée sur une fusion automatique de deux descripteurs : un descripteur de courbures et un descripteur basé sur la géométrie de diffusion. Nous fusionnons donc par la règle du produit des caractéristiques locales et globales extraites à partir d'une forme en utilisant les deux descripteurs. La règle du produit combine les valeurs attribuées par les deux descripteurs aux points situés sur la surface de la forme. Cette règle de fusion est la plus simple et donne de meilleurs résultats que les autres règles de fusion telles que le maximum, le minimum et la fusion linéaire. L'approche proposée améliore considérablement la précision du processus de recherche par forme clef, même avec des changements de postures. Ceci est illustré par les résultats obtenus en utilisant une large sélection d'objets 3D. Nous évaluons la performance de notre méthode, et nous la comparons avec différentes méthodes pré-existantes.

La Section 5.2 décrit brièvement les descripteurs utilisés (ils sont définis en détails dans le chapitre précédent) pour la fusion. Le concept de la fusion, les règles de fusion utilisés pour combiner différents descripteurs, et notre descripteur de fusion sont présentés dans la Section 5.3. Dans la Section 5.4, nous discutons les résultats de fusion de descripteurs, ainsi que les comparaisons avec divers méthodes.

5.2/ DESCRIPTEURS DE FORMES UTILISÉS ET DISSIMILARITÉ DE FORMES

Nous utilisons deux descripteurs de formes : le descripteur de courbures DE (*Discrete Energy*), noté S_1 , et le descripteur de la géométrie de diffusion GD, noté S_2 .

- **Le descripteur S_1** : ce descripteur est basé sur l'estimation locale de la courbure

en chaque sommet du maillage représentant un objet 3D. Le descripteur S_1 pour un sommet v_i est donné par :

$$S_1(v_i) = 4H(v_i)^2 - 2K(v_i) \quad (5.1)$$

où $H(v_i)$ et $K(v_i)$ sont respectivement la courbure moyenne et la courbure Gaussienne.

$$H(v_i) = \frac{1}{4} \sum_{j=1}^{d_i} l_j(\pi - \beta_j) \quad (5.2)$$

où β_j est l'angle diédral formé par deux facettes incidentes au sommet v_i , et l_j et la longueur de leur arête commune.

$$K(v_i) = \frac{2\pi - \sum_{j=1}^{d_i} \alpha_j}{\frac{1}{3}A_{v_i}} \quad (5.3)$$

α_j est l'angle entre deux arêtes consécutives incidentes au sommet v_i , A_{v_i} est la surface des triangles incidents en le sommet v_i , et d_i est le nombre de ses voisins.

- **Le descripteur S_2** : Ce descripteur est basé sur la distribution de distance de diffusion entre les sommets du maillage représentant l'objet. Cette distance est définie en utilisant la chaîne de Markov du graphe associé à l'objet.

Le descripteur S_2 d'un sommet v_i est donné par :

$$S_2(v_i) = \sum_{l=1}^n d_{\Delta_l}(v_i, v_l)^2 \quad (5.4)$$

tel que $d_{\Delta_l}(v_i, v_l)^2$ est la distance de diffusion ; elle est donnée par :

$$d_{\Delta_l}(v_i, v_j) = \sum_{k=1}^K \lambda_k^{\Delta_l} (R_k(v_i) - R_k(v_j)) \quad (5.5)$$

λ_i et R_k sont respectivement les valeurs et les vecteurs propres de la matrice Laplacienne du graphe associé à l'objet.

Les deux descripteurs de formes sont calculés pour tous les sommets du maillage d'un objet. Ces valeurs sont ensuite représentées par des distributions (histogrammes). La distance d entre deux objets A et B représentés chacun par leurs histogrammes H_1 et H_2 est donnée par :

$$d(H_1||H_2) = \chi^2(H_1, H_2) = \frac{1}{2} \sum_{j=0}^{b-1} \frac{(H_1(j) - H_2(j))^2}{H_1(j) + H_2(j)} \quad (5.6)$$

5.3/ LA FUSION DES DESCRIPTEURS

L'idée principale de notre méthode présentée dans ce chapitre consiste à utiliser simultanément deux descripteurs, l'un local S_1 , et l'autre global S_2 . Ces deux descripteurs sont complémentaires : ils décrivent à la fois des caractéristiques locales et globales de l'objet. Cela permet de prendre en compte des informations provenant de différentes sources, et donc de décrire plus efficacement un objet.

En effet, la fusion de classificateurs en utilisant des données obtenues à partir de différentes sources est largement utilisée car c'est une stratégie efficace pour améliorer la reconnaissance et la précision du processus de recherche par forme clef. On peut citer à titre indicatif la classification des images [Sari et al., 2015], la reconnaissance des expressions faciales [Zhang et al., 2015], et les systèmes d'authentification biométrique [Sudha et al., 2014].

Pour concevoir un schéma de fusion de classificateurs, il est très important de savoir quand, quoi et comment combiner. Deux options de fusion sont disponibles : la fusion précoce, et la fusion tardive [Snoek et al., 2005]. La fusion tardive calcule indépendamment deux distances d_1 et d_2 , entre les deux objets à comparer A et B , une pour chaque descripteur, puis combine ces distances. La distance d_1 est calculée en utilisant des histogrammes de A et B sur la base du descripteur S_1 . De même, la distance d_2 est calculée en utilisant des histogrammes de A et B sur la base du descripteur S_2 . Ensuite, les deux distances d_1 et d_2 sont combinées pour définir une distance d finale. La fusion précoce combine les deux descripteurs $S_1(v_i)$ et $S_2(v_i)$ calculés en v_i en un seul descripteur $S(v_i)$. Ainsi, chaque objet est caractérisé par un seul histogramme, sur la base du descripteur $S(v_i)$ qui est le résultat de la fusion de $S_1(v_i)$ et $S_2(v_i)$.

Snoek *et al.* [Snoek et al., 2005] ont étudié et comparé les deux types de fusions. Tenant compte du temps coûteux de la fusion tardive, et des résultats des tests, il a été conclu que la fusion précoce est bien meilleure. Dans notre travail, la fusion précoce est utilisée pour fusionner deux descripteurs S_1 et S_2 à chaque sommet d'un objet donné. Dans la littérature, des règles de fusion tels que le maximum, le minimum, le produit et la fusion linéaire sont les plus fréquemment utilisées [Kuncheva, 2002]. Les deux descripteurs S_1 et S_2 sont complémentaires. La règle du produit est théoriquement la meilleure dans notre situation, et cela est confirmé par nos expériences. Dans ce qui suit, nous expliquons la fusion précoce, ainsi que les différentes règles de fusion utilisées pour combiner les descripteurs.

5.3.1/ FUSION PRÉCOCE ET RÈGLES DE FUSION

Dans le but d'améliorer l'appariement de formes et le processus de recherche par forme clef, différentes règles de fusion ont été étudiées et comparées [Kuncheva, 2002].

Afin de définir la fusion des deux descripteurs considérés dans ce travail, nous devons tout d'abord calculer le degré de confiance en chaque descripteur. Cela est effectué en calculant dans une phase de pré-traitement la précision du processus de recherche pour chaque descripteur.

Nous notons μ_1 et μ_2 la précision du processus de recherche en utilisant respectivement S_1 et S_2 . En effet, μ_1 et μ_2 sont les moyennes des précisions du processus de recherche calculées après une campagne de tests : nous cherchons dans la base de données les objets les plus proches d'une requête, en utilisant le descripteur S_1 (respectivement S_2) ; μ_1 (respectivement μ_2) est la précision moyenne des résultats obtenus sur cent requêtes en utilisant le descripteur S_1 (respectivement S_2).

Par conséquent, $\mu_1 = \sum_{j=1}^T \frac{precision(S_1)_j}{T}$ et $\mu_2 = \sum_{j=1}^T \frac{precision(S_2)_j}{T}$, où T est le nombre de tests.

Nous avons estimé les valeurs de μ_1 , et μ_2 pour un échantillon $T = 100$ tests aléatoires et nous avons obtenu $\mu_1 = 0,74569$ et $\mu_2 = 0,7975$.

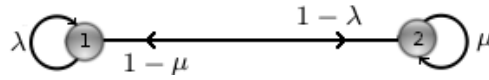


FIGURE 5.1 – Chaîne de Markov à deux états représentant le degré de confiance en chaque descripteur.

Le degré de confiance en chaque descripteur peut être considéré comme une chaîne de Markov à deux états (Figure 5.1). Soit Z_t le descripteur dans lequel nous avons confiance au temps discret t . Alors, (Z_t) est une chaîne de Markov avec les états $\{1, 2\}$.

La matrice de transition de cette chaîne est donnée par l'équation 5.7 :

$$M = \begin{bmatrix} \mu_1 & 1 - \mu_1 \\ 1 - \mu_2 & \mu_2 \end{bmatrix} \quad (5.7)$$

Cette chaîne de Markov a une seule distribution stationnaire α^* , que nous calculons en résolvant : $(I - M^T)\alpha^* = 0$, et $\alpha_1^* + \alpha_2^* = 1$, donc $\alpha_1^* = \frac{1 - \mu_2}{2 - (\mu_1 + \mu_2)}$ et $\alpha_2^* = \frac{1 - \mu_1}{2 - (\mu_1 + \mu_2)}$.

On peut généraliser ce résultat pour L descripteurs. Soit μ_i le degré de confiance en le i ième descripteur. Nous supposons que toutes les transitions de confiance d'un descripteur aux autres descripteurs sont équiprobables. Ainsi, la probabilité de confiance pour aller du i ième descripteur au temps $t = 0$ au j ième descripteur au temps $t = 1$ est donnée par :

$$\Pr(Z_1 = j \mid Z_0 = i) = \begin{cases} \mu_i & \text{si } i = j \\ \frac{1 - \mu_i}{L - 1}, & \text{si } i \neq j \end{cases}$$

Par conséquent, la matrice stochastique M représentant les probabilités de transitions des degrés de confiances est donnée par :

$$M = \begin{bmatrix} p_{11} & \dots & p_{1L} \\ p_{21} & \dots & p_{2L} \\ \vdots & \ddots & \vdots \\ p_{L1} & \dots & p_{LL} \end{bmatrix} \quad (5.8)$$

tel que : $M(i, j) = p_{ij} = \Pr(Z_1 = j \mid Z_0 = i)$, $0 < \mu_i < 1$, et $0 < \frac{1 - \mu_i}{L - 1} < 1$, pour $i = 1, \dots, L$.

Cette chaîne de Markov a une distribution stationnaire unique α^* qui peut être trouvée en résolvant : $(I - M^T)\alpha^* = 0$, sachant que $\sum_{j=1}^L \alpha_j^* = 1$.

Le vecteur de probabilité stationnaire est alors définie par $\alpha^* = (\alpha_1^*, \dots, \alpha_L^*)$, où la probabilité stationnaire α_i^* du i ième descripteur est donnée par :

$$\alpha_i^* = \frac{1 - \left(\sum_{j=1, j \neq i}^L \mu_j / (L - 1) \right)}{L - \sum_{j=1}^L \mu_j} \quad (5.9)$$

Definition 6. Soit α_k (nous retirons l'étoile de α_k^* pour plus de commodité) le degré de confiance dans le k ième descripteur dans la combinaison, et S_k la valeur du k ième descripteur. Remarquons que la somme des degrés de confiances vaut 1.

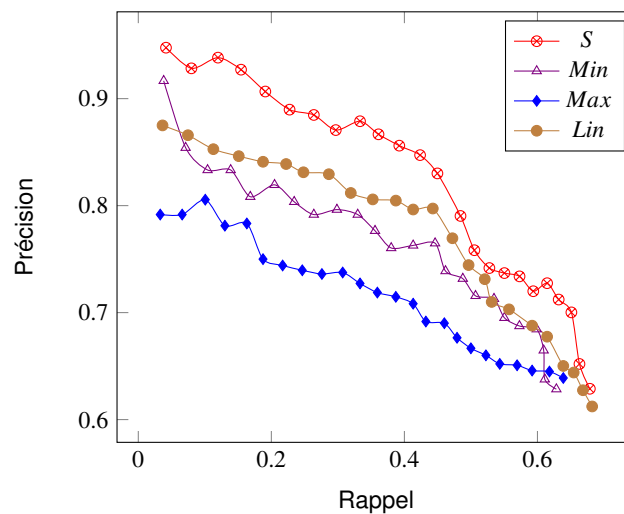


FIGURE 5.2 – Courbes de précision-rappel : résultats du processus de recherche pour la base de données TOSCA, en utilisant les différentes règles de fusion.

Nous présentons maintenant les règles des fusions utilisées dans ce chapitre afin de combiner les descripteurs : le maximum, le minimum, le produit, et la fusion linéaire.

- **Maximum** : La règle du maximum permet de sélectionner le descripteur qui a la valeur maximum. $S = \max\{\alpha_k \times S_k, k = 1, \dots, L\}$.
- **Minimum** : Le minimum prend le descripteur avec la valeur minimale. Donc, pour L descripteurs la règle du minimum est $S = \min\{\alpha_k \times S_k, k = 1, \dots, L\}$.
- **Fusion linéaire** : les descripteurs sont combinés en tenant compte des coefficients associés à ces derniers [Tax et al., 1997]. La fusion linéaire est donnée par la combinaison linéaire convexe : $S = \sum_{k=1}^L \alpha_k \times S_k$.
- **Produit** : Dans cette règle, les valeurs des descripteurs sont multipliées. Pour L descripteurs, la fusion par la règle de produit est donnée par :

$$S = \prod_{k=1}^L \alpha_k \times S_k = \left(\prod_{k=1}^L \alpha_k \right) \times \left(\prod_{k=1}^L S_k \right) \quad (5.10)$$

La règle du produit est l'une des règles les plus simples et les plus efficaces parmi toutes les règles utilisées dans les schémas de combinaison [Tax et al., 1997, Abdullah et al., 2009], en particulier lorsque les descripteurs sont indépendants (ou dans notre cas, complémentaires).

Pour deux descripteurs, nous avons : $S = (\alpha_1 \times S_1) \times ((1 - \alpha_1) \times S_2) = \alpha_1 \times (1 - \alpha_1) \times S_1 \times S_2$. Dans notre cas, les sommets des descripteurs sont estimés correctement, cette règle de produit fournit théoriquement la meilleure estimation pour la fusion. Ceci est également confirmé par les expériences que nous avons menées sur une variété d'objets 3D.

5.3.2/ LE DESCRIPTEUR PROPOSÉ : S

Dans cette section, nous décrivons notre nouveau descripteur S.

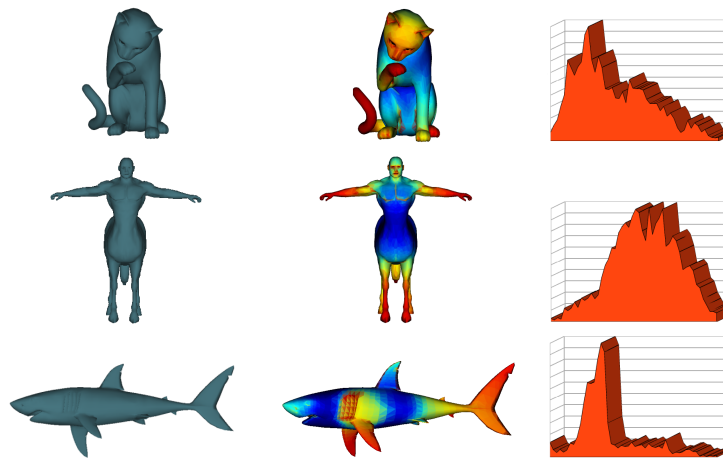


FIGURE 5.3 – Un exemple de descripteur S calculé pour différentes formes. A gauche : le modèle original. Au centre : la carte de couleurs correspondant aux valeurs du descripteur calculés en chaque sommet du modèle. A droite : les histogrammes correspondants.

Pour un sommet donné v_i , nous notons $S_k(v_i)$ la valeur du k ième ($k = 1, 2$) descripteur calculée pour ce sommet.

Le descripteur S applique la règle du produit définie dans la section 5.3.1 pour combiner les deux descripteurs S_1 et S_2 : $\alpha_1 \times (1 - \alpha_1) \times S_1 \times S_2$. Il faut noter que $S(v_i) = \prod_{k=1}^L \alpha_k \times S_k(v_i)$

est égal au produit des descripteurs $\prod_k^L S_k$, indépendamment de la constante multiplicative

$\prod_k^L \alpha_k$. Cette constante ne dépend pas des formes, de sorte que la même transformation est appliquée sur l'ensemble des histogrammes et n'a aucune incidence sur la distance relative entre histogrammes. De toute évidence, cette constante peut être ignorée, donc, la combinaison des L descripteurs est : $S(v_i) = \prod_k^L S_k(v_i)$. Ainsi le calcul de α^* est inutile pour la fusion avec la règle du produit. Cela signifie qu'aucune interaction avec l'opérateur humain n'est nécessaire, c-à-d aucune phase d'apprentissage. Enfin, le résultat de la combinaison (S) des deux descripteurs au niveau d'un sommet v_i est donnée par l'équation 5.11.

$$S(v_i) = S_1(v_i) \times S_2(v_i) \quad (5.11)$$

La Figure 5.3 montre un exemple de résultats de l'application du descripteur S à différentes formes. L'algorithme 3 résume les différentes étapes du processus de recherche par forme clef en utilisant S .

5.4/ RÉSULTATS EXPÉRIMENTAUX

Dans cette section, nous discutons les résultats du processus de recherche par forme clef en utilisant le descripteur S proposé.

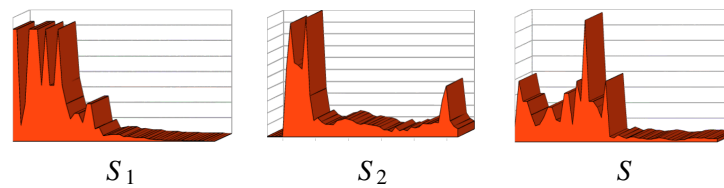


FIGURE 5.4 – Un exemple des histogrammes calculés en utilisant les différents descripteurs. A gauche : le descripteur S_1 . Au milieu ; le descripteur S_2 . A droite : La fusion des deux descripteurs.

5.4.1/ PARAMÈTRES

Ici, nous donnons les valeurs des différents paramètres utilisés par notre méthode pour obtenir des résultats optimaux. Nous avons empiriquement déterminé la meilleure valeur de chacun de ces paramètres, en traitant un échantillon des objets de la base de données TOSCA. Ces paramètres sont principalement liés au descripteur de géométrie de diffusion S_2 .

Les paramètres K et Δ_t dans l'équation 5.5 représentent respectivement le nombre des K valeurs propres avec la plus grande valeur absolue et les vecteurs propres correspondants, et le nombre des étapes de la marche aléatoire dans le graphe \mathcal{G} associé à l'objet. En effet, dans la pratique, ces paramètres sont définis empiriquement [Fang et al., 2011, Dey et al., 2010]. Une grande valeur de Δ_t permet de refléter la structure globale de la surface du modèle. En outre, il a été démontré que l'utilisation de quelques valeurs propres (les K plus grandes en valeurs absolues), avec une grande valeur de Δ_t , permet d'obtenir des résultats convenables [Coifman et al., 2005]. Nos expériences empiriques le confirment.

Nous avons testé plusieurs valeurs de Δ_t ($\Delta_t \in [0, 100]$). En tenant compte la complexité de calcul des K valeurs propres et leurs vecteurs propres, en sachant que la complexité est $O(Kn^2)$, le tuple $\Delta_t = 40$ et $K = 20$ (sachant que $n \approx 3400$) donne des résultats satisfaisants.

Comme mentionné ci-dessus, chaque descripteur de forme calculé pour les n sommets du maillage d'un objet est décrit par un histogramme H . L'histogramme H est divisé en b tiroirs (intervalles). Nous avons testé plusieurs valeurs pour b [Sturges, 1926, Hyndman, 1995] : $\lceil 2n^{\frac{1}{3}} \rceil$, $\lceil 1 + \log_2 n \rceil$, et simplement $\lfloor \sqrt{n} \rfloor$. Les résultats des tests montrent que $b \approx \lfloor \sqrt{n} \rfloor$ est la valeur commune qui donne de bons résultats pour la plupart des objets de la base de données utilisée.

5.4.2/ RÉSULTATS

Dans les premières expériences, nous avons étudié et comparé les résultats correspondants à l'utilisation des trois descripteurs S_1 , S_2 , et S . La Figure 5.5 montre quelques résultats de la classification des modèles 3D en considérant les différents descripteurs. Dans la figure, nous montrons les 9 premiers objet retournés par le processus de recherche pour une requête donnée.

Les résultats illustrés à travers les courbes de la Figure 5.5 montrent que le descripteur S améliore la performance des deux descripteurs S_1 et S_2 . Nous pouvons aussi constater

Algorithm 3 Le pseudo-algorithme d'appariement des forme en utilisant S .

```

1: Entrée : une requête  $Q$  et une base de données  $\mathcal{E}$ 
2: fonction_fusion :  $\mathcal{M}$  un maillage ;
3: for chaque sommet  $v_i \in \mathcal{M}$  do
4:   calculer le descripteur  $S_1(v_i)$  ;
5:   calculer le descripteur  $S_2(v_i)$  ;
6:   fusionner les deux descripteurs :  $S(v_i) = S_1(v_i) \times S_2(v_i)$  ;
7: end for
8: Retourner un vecteur  $S(\mathcal{M})$  ;
9: fonction_Fusion( $Q$ ) ;
10: calculer l'histogramme  $H(Q)$  ;
11: for chaque  $Q' \in$  dans la base de données  $\mathcal{E}$  do
12:   %%  $H(Q')$  est calculé dans une étape de pré-traitement %%
13:   calculer les histogrammes  $H(Q')$  ;
14:   calculer  $\chi^2(H(Q), H(Q'))$  ;
15: end for
16: Sortie : les objets  $\mathcal{E}$  les plus proches de  $Q$  ;

```

que celui ci permet de distinguer les hommes des femmes.

Les résultats sur la précision du processus de recherche pour les différents descripteurs appliqués pour chaque catégorie d'objet de la base de donnée TOSCA sont rapportés dans le tableau 5.1. En se basant sur les résultats de ce tableau ainsi que sur les courbes précision-rappel des différents descripteurs dans la figure 5.6, on peut voir que parfois S_1 est meilleur que S_2 , parfois le contraire, mais l'efficacité du descripteur S est toujours garantie et meilleure que celles des deux descripteurs.

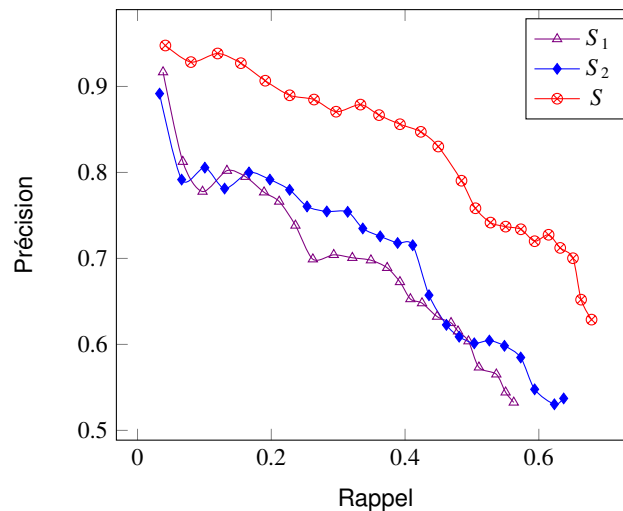


FIGURE 5.6 – Courbes de précision-rappel. (a) : résultats en utilisant les descripteurs S_1 , S_2 , et S .

En outre, à partir de la figure 5.2, on peut déduire que la règle du produit est meilleure que toutes les autres règles de fusion.

La deuxième partie de nos expériences concerne l'étude de la capacité de distinction

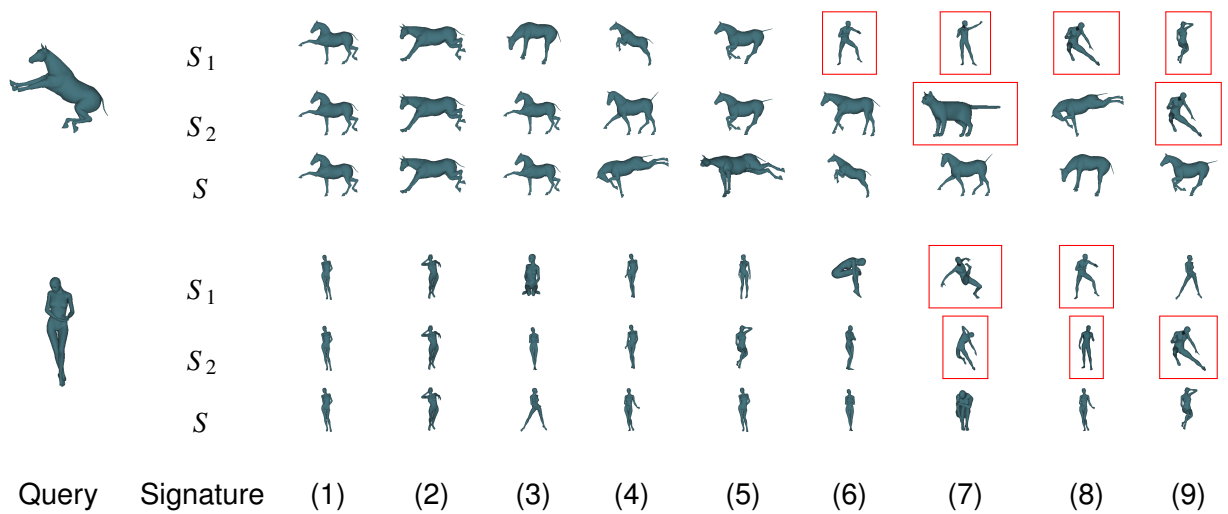


FIGURE 5.5 – Résultats de recherche en utilisant S_1 , S_2 et S . Colonne 1 : une requête donnée. Colonnes i : la liste des objets trouvés pour une requête. Les erreurs de la recherche sont entourées en rouge. S_1 et S_2 commettent des erreurs, S n’en commet pas.

d’objets par notre méthode.

La matrice de dissimilarité de tous les objets dans la base de données est présentée dans la figure 5.7. Elle montre la robustesse de notre méthode et sa capacité à distinguer les objets. Tous les objets de la même catégorie sont représentés par des éléments de la diagonale de la matrice ; les distances sur la diagonale de la matrice sont nettement plus petites que les autres.

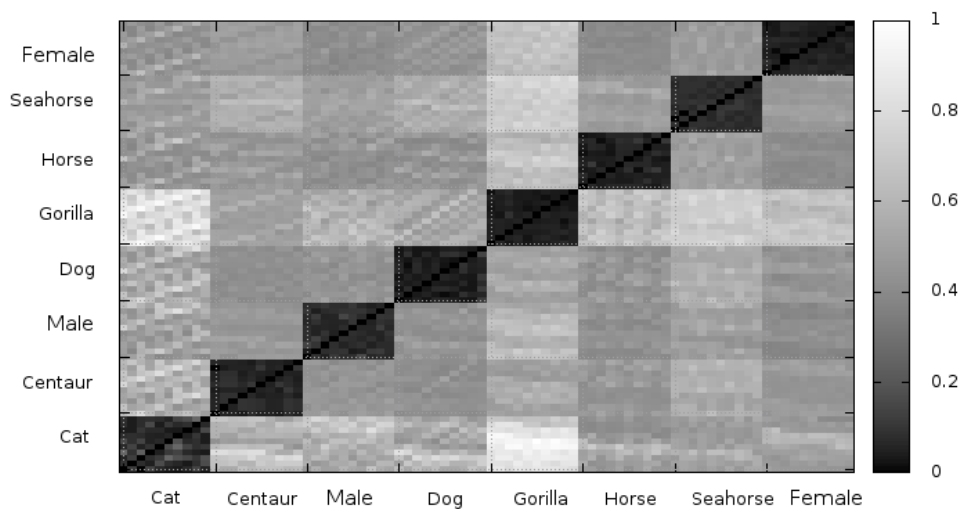


FIGURE 5.7 – Matrice de distance de la base de données. Les objets dissimilaires sont représentés en clair.

TABLE 5.1 – La précision moyenne du processus de recherche pour chaque catégorie d'objets de la base de données utilisée.

Catégorie	Précision (%)					
	First-tier (FT)			Second-tier (ST)		
	S_1	S_2	S	S_1	S_2	S
Cat	58.987	60.6061	88.541	32.875	26.282	35.589
Centaur	91.569	83.333	98.764	47.658	53.030	56.589
Male	79.485	81.039	87.826	40.856	51.4706	56.658
Horse	82.698	98.395	97.897	42.559	46.732	47.598
Seahorse	98	96.530	98.874	51.658	54.166	54.256
Female	81.897	77.651	86.578	52.56	48.090	49.236
Dog	47.789	65.444	79.548	38.658	44.444	46.897
Gorilla	52.987	75.041	84.758	47.236	50.555	52.145
Toutes	74.176	79.754	90.348	44.257	43.181	49.871

TABLE 5.2 – La précision moyenne de la base de données en utilisant différents descripteurs.

Catégorie	Précision (%)														
	First-tier (FT)								Second-tier (ST)						
	S	D2	LSD	TD	HMDG	LSF	IDM	S	D2	LSD	TD	HMDG	LSF	IDM	
Cat	88.54	29.16	64.59	84.96	88.65	66.45	80.55	35.58	17.62	25.54	25.64	33.65	27.30	23.55	
Centaur	98.76	31.94	75.87	78.78	93.60	79.5	76.45	56.58	26.51	41.32	43.93	51.44	44.65	52.24	
Male	87.82	39.18	62.87	70.90	89.30	60.33	65.33	56.65	30.05	42.45	46.55	55.30	40.35	51.33	
Horse	97.89	32.50	90.21	99.46	95.46	92.44	89.4	47.59	23.36	41.45	46.89	48.5	42.44	45.65	
Seahorse	98.87	56.80	97	96	90.65	91.32	90.15	54.25	40.70	50.24	54.16	49.2	52.60	44.13	
Female	86.57	42.16	63.56	71.96	80.90	69.80	72.78	49.23	33.50	42.44	43.75	52.3	46.30	51.75	
Dog	79.54	30	72.32	86.42	79.54	86.34	81.44	46.89	24.07	45.24	50	40.36	43.85	41.32	
Gorilla	84.75	29	62.87	81.18	83.15	60.87	80.35	84.75	25	52.32	54.44	49.88	56.55	47.67	
Toutes	90.34	36.34	73.66	84.29	87.65	75.63	79.55	49.87	27.60	42.63	45.67	47.57	44.25	44.07	

5.4.3/ RÉSULTATS DE COMPARAISON

Pour montrer l'efficacité de notre méthode, nous avons comparé les résultats fournis par le descripteur S avec ceux de certains descripteurs proposés dans la littérature, et notamment ceux qui sont liés à notre méthode. Nous avons choisi ces descripteurs pour leur lien avec notre méthode, et pour leur efficacité.

Les méthodes que nous utilisons dans nos comparaisons sont : *Hybrid shape Descriptor using Meta Distance Generation* (HMDG) [Li et al., 2014], *shape distribution* (D2) [Osada et al., 2001], *Local Shape Distribution Descriptor* (LSD) [Li et al., 2012], *Temperature Distribution* (TD) [Fang et al., 2011], *Local Statistical Features* (LSF) [Ohkita et al., 2012], et *Isometric Deformation Modelling* (IDM) [Smeets et al., 2009].

- **HMDG** : C'est une méthode permettant de fusionner des mesures de dissimilarité calculées en utilisant des caractéristiques locales et globales [Li et al., 2014]. La nouvelle distance calculée par la fusion est représentée comme une combinaison convexe des distances. Étant donné que cette méthode est coûteuse en temps de calcul, seul un nombre réduit de sommets de la surface d'un modèle est considéré ($n = 500$).

- **D2** : Ce descripteur permet de décrire les distances euclidiennes entre les points d'échantillonnage en tenant compte de leur distribution (histogramme). Le nombre de tiroirs b d'un histogramme est 40, et le nombre de points échantillonnés est $0.3 \times n$, où n est le nombre de sommets.
- **LSD** : Le descripteur LSD permet de décrire les n régions de rayon r centrées aux n sommets de la surface d'un modèle par une distribution des distances. Chaque région de LSD est associée à un histogramme décrivant les distances entre les points de la région et de son centre. Les n régions sont partitionnées en k descripteurs LSD. Le calcul de la distance entre deux formes représentées chacune par ses k descripteurs LSD est un problème d'appariement optimal dans un graphe biparti [Kuhn, 1955]. Les paramètres de cette méthode sont les suivants : $r = 0.3$, $n = 2000$, $k = 150$, et le nombre de tiroirs d'un histogramme b est 30.
- **TD** : Le descripteur TD simule la diffusion de la distribution de la température sur la surface d'un objet, en utilisant les valeurs et les vecteurs propres de la matrice de la Laplacienne du graphe du maillage triangulaire décrivant l'objet. Pour être cohérent avec la mise en oeuvre de ce descripteur dans [Fang et al., 2011], et pour obtenir des résultats optimaux, le temps de la propagation de la température t est fixé à 0.8 et le nombre de valeurs propres K vaut 150.
- **LSF** : C'est un descripteur local simple [Ohkita et al., 2012]. Il extrait $n = 500$ points caractéristiques à partir de la surface d'un objet, et associe à chacun un histogramme. Un histogramme est calculé à partir des k points appartenant à la sphère d'un rayon $r = 0.2$, dont le centre est un point caractéristique. Pour chaque point de la sphère, un 4-tuple est calculé. Un tuple contient la distance moyenne d entre le centre de la sphère (le point caractéristique) et les k points aléatoires dans la sphère, et trois angles calculés en terme des deux points parmi les k points et le centre de la sphère.
- **IDM** : Ce descripteur utilise les valeurs propres de la matrice des distances géodésiques associée à la surface d'un objet [Smeets et al., 2009]. La matrice des distances représente les longueurs des chemins géodésiques (les plus courts), sur la surface de l'objet entre les sommets de la surface de l'objet. Les 50 plus grandes valeurs propres et $n = 500$ points échantillonnés sont considérés pour le calcul de ce descripteur.

Les résultats de comparaison sont illustrés dans la figure 5.8 et le tableau 5.2. Les informations fournies par les courbes de précision-rappel de la Figure 5.8, et les valeurs de précision du processus de recherche dans le tableau 5.2 montrent clairement que notre méthode fonctionne mieux que les autres méthodes. Dans ces résultats expérimentaux, la fusion des descripteurs en utilisant la règle de produit surpasse la performance de chacun des descripteurs, ce qui confirme la théorie des schémas de combinaison.

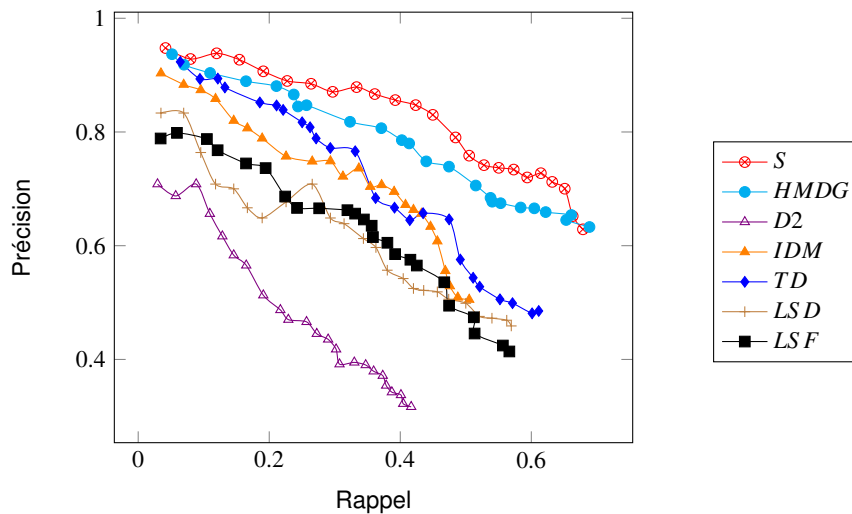


FIGURE 5.8 – Courbes précision-rappel pour la comparaison des descripteurs.

5.4.4/ PRÉ-TRAITEMENT ET TEMPS DE CALCUL

L'histogramme de chaque objet dans la base de données est calculé dans une phase de pré-traitement, et sauvegardé dans un fichier, pour être réutilisé plus tard.

En temps réel, seul le descripteur de forme de la requête Q est calculé. La requête Q peut ne pas faire partie de la base de données.

Le Tableau 5.3 illustre le temps de calcul pour les différents descripteurs de forme considérés, en fonction de leur nombre approprié de points d'échantillonnage (voir la section 5.4.3). Notez que la dernière ligne de ce tableau affiche le temps de calcul pour le processus de recherche pour une requête Q (elle est censée être extérieure à la base de données).

D'après les résultats présentés dans le Tableau 5.3, nous pouvons clairement voir que certains descripteurs, par exemple LSD, s'exécutent plus rapidement que notre descripteur proposé, mais ils sont moins précis. En outre, le descripteur HMDG qui est censé être meilleur que tous les autres est nettement plus lent que notre descripteur.

Notez que tous les tests ont été effectués sur un ordinateur portable (Intel (R) Core (TM) i5 CPU M 560 2.67GHz, 6 Go de RAM), fonctionnant sous Arch Linux GNU / Linux 3.16.

TABLE 5.3 – Temps de calcul nécessaire pour chaque descripteur, pour chaque catégorie de la base de données TOSCA.

Catégorie	# de sommets par catégorie	Temps de calcul en phase de pré-traitement (sec)						
		S	D2	LSD	TD	HMDG	LSF	IDM
Cat	30.6k	12.879	5.322	6.435	177.12	1191.141	7.535	1156.569
Centaur	20.4k	8.586	3.548	4.29	118.08	794.094	4.89	770.646
Male	119k	50.085	20,966	25.025	688.8	4632.215	28.525	4495.435
Horse	57.8k	24.327	10.053	12.155	334.56	2249.933	13.855	2183.497
Seahorse	20.4k	8.586	3.548	4.29	118.08	794.094	4.89	770.646
Female	85k	35.775	14,578	17.16	492	3303.725	20.375	3211.025
Dog	37.4k	15.741	6.505	7.865	216.48	1455.839	9.965	1412.851
Gorilla	71.4k	30.051	12.419	15.015	413.28	2779.329	17.115	2697.261
Temps réel		1.461	0.621	0.715	19.71	132.379	0.845	128.471

5.5/ CONCLUSION

Dans ce chapitre, nous avons proposé un nouveau descripteur basé sur la fusion d'un descripteur de courbures et un descripteur basé sur la géométrie de diffusion, en tenant compte des caractéristiques locales et globales. Le descripteur S est invariant par changement de poses, et permet de caractériser localement et globalement la forme grâce à la fusion entièrement automatique des deux descripteurs. En outre, il présente l'avantage d'être automatique, rapide et facile à mettre en oeuvre. Nous avons évalué notre approche sur une variété d'objets 3D. Les essais montrent que S est efficace et nettement plus performant que chacun des deux descripteurs, ainsi que des descripteurs de forme 3D pré-existants.

COMBINAISON DES MESURES DE DISSIMILARITÉ

6.1/ INTRODUCTION

Comme nous l'avons mentionné précédemment, dans le domaine de la recherche par forme clef, une seule méthode utilisée pour déterminer la similarité des formes reste loin d'être universelle, et capable de trouver des résultats satisfaisants pour toutes sortes d'objets. Toute méthode donne de bons résultats pour un certain jeu de données, et de moins bons pour d'autres.

Dans ce chapitre, nous présentons une nouvelle méthode permettant de combiner n'importe quelles méthodes utilisées pour déterminer la dissimilarité des objets. Cette combinaison permet d'améliorer les résultats du processus de recherche (la précision).

En effet, quelque soit la méthode utilisée, le principe pour calculer la dissimilarité entre objets est toujours le même : un descripteur de formes induit une distance entre formes. Pour un ensemble de descripteurs et de distances donnés, notre méthode calcule la meilleure combinaison linéaire convexe des distances : elle considère le graphe complet, dont les sommets sont les formes de la base de données ou la requête, et les arêtes sont étiquetées par plusieurs distances, une par descripteur ; ensuite cette méthode calcule par programmation linéaire la combinaison convexe des distances qui maximise soit la longueur des plus courts chemins entre la requête et les autres objets de la base, soit le plus court chemin entre la requête et un objet donné ; ce dernier représente l'objet actuellement comparé à la requête. Nous utilisons la bibliothèque de programmation linéaire GLPK (*GNU Linear Programming Kit*).

La méthode est complètement automatique, et permet d'améliorer la performance de n'importe quelle mesure de dissimilarité (distance).

Le reste de ce chapitre est organisé comme suit. La Section 6.2 décrit le contexte de notre travail ainsi que les grandes lignes de notre méthode. La description de notre approche est présentée dans la Section 6.3. La section 6.4 présente les résultats expérimentaux.

6.2/ VUE D'ENSEMBLE

La détermination de la similarité des objets est liée à une distance ou une mesure de dissimilarité calculée entre les descripteurs de formes associés aux objets [Cheng et al., 2013, Veltkamp et al., 2001]. A l'heure actuelle, aucune distance n'est considérée comme étant parfaite. Cette incapacité est due à la grande variété de représentations d'objets. Par exemple, certains types de distances considèrent le spectre de la forme, en utilisant les opérateurs de Laplace [Coifman et al., 2005] ; par contre des formes différentes peuvent être isospectrales [Reuter et al., 2006]. Une façon naturelle d'améliorer une méthode donnée est de la combiner avec une autre méthode, dans le but de bénéficier de leurs avantages. Cette combinaison peut être supervisée, c-à-d quelle dépend d'une classification préalable de la base de données, ou alors non supervisée, c-à-d automatique.

La méthode que nous proposons dans ce chapitre est complètement automatique. La combinaison de n distances résulte en une seule distance ; aucune classification préalable des objets de la base de données n'est effectuée. Du point de vue algorithmique, elle nécessite seulement une résolution d'un problème de programmation linéaire.

Nous notons d_1, d_2, \dots, d_n , les n distances utilisées pour mesurer la dissimilarité des formes. Nous supposons que ces distances sont normalisées ($d_i \in [0, 1]$), et symétriques ($d_i(A, B) = d_i(B, A)$). \mathcal{D} est une base de données ; nous définissons un graphe G , dont les noeuds sont les objets dans $\mathcal{D} \cup \{Q\}$, et Q représente la requête. Le poids d'une arête (A, B) , ou sa longueur d dans le graphe G est une combinaison des distances d_i entre les noeuds A et B . La distance d est définie comme suit :

$$\forall A, B \in \mathcal{D} \cup \{Q\}, \quad d(A, B) = \sum_{i=1}^n \alpha_i \times d_i(A, B), \quad \sum_{i=1}^n \alpha_i = 1$$

où chaque $\alpha_i \in [0, 1]$ est le coefficient associé à la distance d_i , et la somme de ces coefficients est égale à 1. Les coefficients α_i sont calculés en maximisant un critère ρ par la résolution d'un problème de programmation linéaire. Ce critère peut être ρ_1 la somme des longueurs des plus courts chemins entre Q et tous les objets de \mathcal{D} , ou ρ_2 la longueur du plus court chemin dans le graphe G entre Q et un objet S . Pour trouver les m objets les plus proches de Q , la distance D entre Q et S est définie comme étant le longueur du plus court chemin entre Q et S dans le graphe G .

Nous comparons notre méthode avec des méthodes de fusion tels que le minimum, le maximum, le produit et la fusion linéaire en utilisant les trois bases de données TOSCA [YA. M. Bronstein, 2007], SHREC'11 [Li et al., 2012], et ESB [Jayanti et al., 2006].

6.2.1/ DESCRIPTEURS DE FORMES UTILISÉS

Nous utilisons les descripteurs suivants : diamètre local LFD (*The local diameter*) [Gal et al., 2007], descripteur de géométrie de diffusion GD [Coifman et al., 2005] , descripteur de courbures DE (*Discret Energy*) [Meyer et al., 2003], descripteur de distribution D2 (*shape distribution*) [Osada et al., 2001], et l'enveloppe convexe CH (*Convex Hull*) [Kalyanaraman et al., 2005].

Notez que la contribution de cette méthode ne consiste pas à évaluer un descripteur en particulier ou à le comparer avec les autres, mais à montrer l'efficacité de la combinaison

de cet ensemble de descripteurs. De plus, le choix des descripteurs n'a aucune importance : d'autres descripteurs peuvent être utilisés. Les descripteurs que nous utilisons sont choisis pour leur efficacité et leur simplicité.

6.2.2/ RÈGLES DE FUSION

Les études ont montré que le processus de recherche par forme clef peut être amélioré par la combinaison de plusieurs descripteurs [Kittler et al., 1998]. La fusion est la technique la plus utilisée pour la combinaison des descripteurs. Dans notre chapitre, nous comparons notre méthode avec différentes règles de fusion tels que : maximum, minimum, la moyenne, et la fusion linéaire [Kittler et al., 1998].

Étant données deux formes A et B , pour n distances, nous définissons les règles de fusion comme suit :

- **Maximum** : $d(A, B) = \max(d_1(A, B), d_2(A, B), \dots, d_n(A, B))$.
- **Minimum** : $d(A, B) = \min(d_1(A, B), d_2(A, B), \dots, d_n(A, B))$.
- **Moyenne** : $d(A, B) = \frac{1}{n}(d_1(A, B) + d_2(A, B) + \dots + d_n(A, B))$.
- **Produit** : $d(A, B) = d_1(A, B) \times d_2(A, B) \times \dots \times d_n(A, B)$.
- **Linéaire** : cette règle de fusion combine des distances en tenant compte des poids c_i associés à ces derniers. La fusion linéaire est donnée comme suit :

$$d(A, B) = \sum_{i=1}^n c_i \times d_i(A, B)$$
 les c_i sont calculés en utilisant des techniques d'apprentissage [Chahooki et al., 2012].

6.3/ NOTRE APPROCHE

6.3.1/ NOTATIONS

d est une distance donnée, et \mathcal{D} est un ensemble d'objets. (d, \mathcal{D}) définit un graphe complet G , dont les sommets sont les objets dans \mathcal{D} , et le poids de chaque arête (A, B) est la distance d entre A et B . La distance d est une combinaison linéaire convexe d'un ensemble de distances $d_i, i = 1, \dots, n$. Pour toutes les paires de sommets $A \in \mathcal{D}, B \in \mathcal{D}$, la distance $d(A, B)$ est donnée par :

$$d(A, B) = \sum_{i=1}^n \alpha_i \times d_i(A, B), \quad \sum_{i=1}^n \alpha_i = 1, \quad 0 \leq \alpha_i \leq 1 \quad (6.1)$$

Nous notons d_i^* la distance géodésique induite par le couple (d_i, \mathcal{D}) , tel que $d_i^*(A, B)$ est la longueur du plus court chemin entre A et B dans un graphe G_i . La longueur d'une arête (A, B) dans le graphe G_i est la distance d_i . Nous pouvons constater que $d_i^*(A, B)$ peut bel et bien être inférieure à $d_i(A, B)$, et en aucun cas elle ne peut être plus grande. Le calcul du plus court chemin dans un graphe est un problème en temps polynomial [Cormen et al., 2009].

6.3.2/ DESCRIPTION DE NOTRE APPROCHE

Notre méthode consiste en deux phases : la première est indépendante de la requête Q , et est effectuée une seule fois sur la base de données \mathcal{D} ; dans une phase de pré-traitement, toutes les distances $d_i(A, B), i = 1, \dots, n$ pour toutes les paires d'objets A et B dans \mathcal{D} sont calculées. La deuxième phase dépend de la requête Q ; dans cette phase, nous calculons une nouvelle distance $D(Q, S)$ entre la requête Q et toutes les formes S dans \mathcal{D} . La distance D est la longueur de plus court chemin entre les deux sommets Q et S dans le graphe G . Pour définir la distance D entre Q et les formes dans \mathcal{D} , les valeurs des poids α_i sont calculés en maximisant le pouvoir de séparation¹ dans le graphe. Du point de vue algorithmique, les valeurs de α_i sont calculées pour qu'un certain critère ρ_1 or ρ_2 soit maximal. ρ_1 est la somme des longueurs de tous les plus courts chemins entre Q et tous les sommets B dans \mathcal{D} . ρ_2 est la longueur du plus court chemin de Q à une forme donnée $S \in \mathcal{D}$, tel que S est la forme actuellement comparée à Q . Le calcul des coefficients $\alpha = (\alpha_i)$ est équivalent à un problème de programmation linéaire. Les inconnues, les contraintes et la fonction objectif de ce programme sont spécifiés dans ce qui suit. En plus des inconnues α_i , il y a encore une inconnue par sommet dans le graphe G : à chaque noeud B , à l'exception de Q , on associe un potentiel $\pi_B = D(Q, B)$ représentant la longueur du plus court chemin entre Q et B dans le graphe G . Habituellement, π_B est appelé le potentiel de B [Bernhard et al., 2008]. Par définition, le potentiel π_Q de Q est connu et sa valeur vaut zéro. Il existe aussi une inconnue $d(A, B)$ pour chaque arête dans le graphe ; $d(A, B)$ est une inconnue représentant la longueur de l'arête (A, B) .

Les contraintes du problème linéaire sont les suivantes. Chaque arête $(A, B) \in E$ est associée avec deux contraintes d'inégalités linéaires :

$$\pi_B \leq \pi_A + d(A, B), \quad \pi_A \leq \pi_B + d(A, B), \quad \forall (A, B) \in E$$

et la contrainte :

$$d(A, B) = \sum_{i=1}^n \alpha_i \times d_i(A, B)$$

toutes les inconnues $\alpha_i, d(A, B), \pi_A$ et π_B sont positives ; de plus $\sum_{i=1}^n \alpha_i = 1$.

L'objectif du problème est ensuite de maximiser soit ρ_1 soit ρ_2 selon les contraintes précédentes avec :

$$\rho_1 = \sum_{B \in \mathcal{D}} \pi_B \quad \text{et} \quad \rho_2 = \pi_S$$

Pour ρ_2 , la forme $S \in \mathcal{D}$ est la forme comparée à la requête Q .

Nous pouvons conclure cette section avec les remarques suivantes :

- Toutes les inconnues : les potentiels π_B , les distances $d(A, B)$ et les coefficients α_i dépendent de la requête Q .
- Au lieu d'utiliser :

1. ou le pouvoir de résolution. Nous avons choisi cette appellation par analogie avec le pouvoir de résolution, ou de séparation, des instruments d'optique, microscopes ou télescopes.

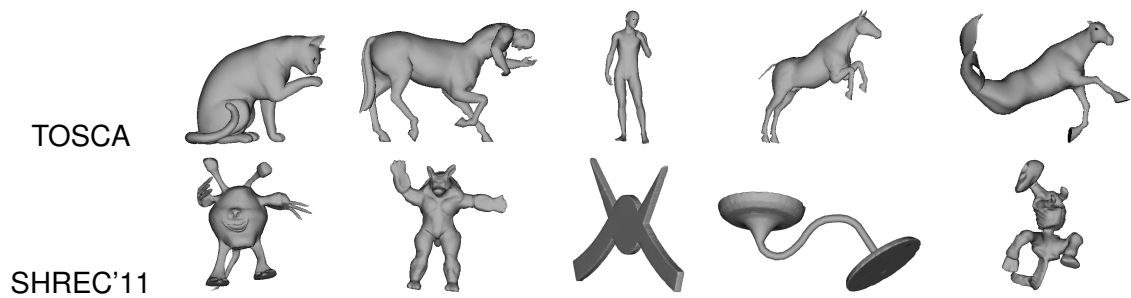


FIGURE 6.1 – Exemples de modèles 3D des bases de données *TOSCA* [Young et al., 2007] et *SHREC'11* [Li et al., 2012].

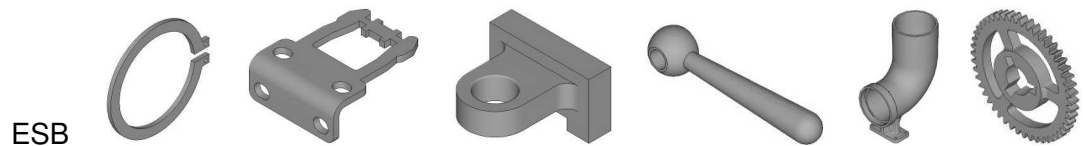


FIGURE 6.2 – Exemples de modèles CAD de la base de données *ESB* [Jayanti et al., 2006].

$$d(A, B) = \sum_{i=1}^n \alpha_i \times d_i(A, B)$$

nous pouvons aussi utiliser :

$$d(A, B) = \sum_{i=1}^n \alpha_i \times d_i^*(A, B)$$

— La variante de notre méthode qui utilise $\sum_i \alpha_i d_i^*$ au lieu de $\sum_i \alpha_i d_i$ nécessite beaucoup plus de temps de calcul, et donne des résultats proches, donc nous ne la mentionnons pas dans la partie expérimentale.

ρ_1 et ρ_2 donnent eux aussi des résultats proches. Les résultats illustrés dans la section suivante sont tous donnés en utilisant ρ_1 .

6.4/ RÉSULTATS

Pour évaluer la performance de notre approche, une série de tests est présentée dans cette section. Chaque méthode basée sur une mesure de dissimilarité donnée est comparée avec notre méthode. Nous considérons dans les comparaisons, ces différentes méthodes ainsi que leurs améliorations via la fusion, en utilisant la règle de maximum, minimum, etc . . .

6.4.1/ DESCRIPTEURS DE FORMES ET BASES DE DONNÉES 3D UTILISÉES

Pour évaluer notre méthode et faire les comparaisons nécessaires, nous utilisons différentes bases de données 3D. Nous utilisons les bases de données [Young et al., 2007]

et SHREC'11 [Li et al., 2012], et une base de modèles CAD (*Engineering Shape Benchmark* ESB) [Jayanti et al., 2006]. Ces bases de données sont décrites en détails dans le deuxième chapitre de cette thèse. Quelques modèles des bases de données utilisées sont illustrés dans les Figures 6.1 et 6.2.

Pour évaluer notre méthode en utilisant les différentes bases de données, nous utilisons les deux bases de données TOSCA et SHRECK'11 pour les descripteurs LFD, GD, DE, et D2. Les descripteurs CH et D2 sont utilisés pour évaluer les résultats de notre méthode appliquée à des modèles de la base de données ESB. Notons que d'autres types de descripteurs peuvent être utilisés pour ce type de modèles [Jayanti et al., 2006]. Les deux descripteurs que nous utilisons dans ce travail sont simples et efficaces.

Les Figures de 6.3 à 6.9 montrent les courbes de précision-rappel obtenues dans les expériences que nous avons menées. Ces résultats expérimentaux sont discutés dans la section suivante.

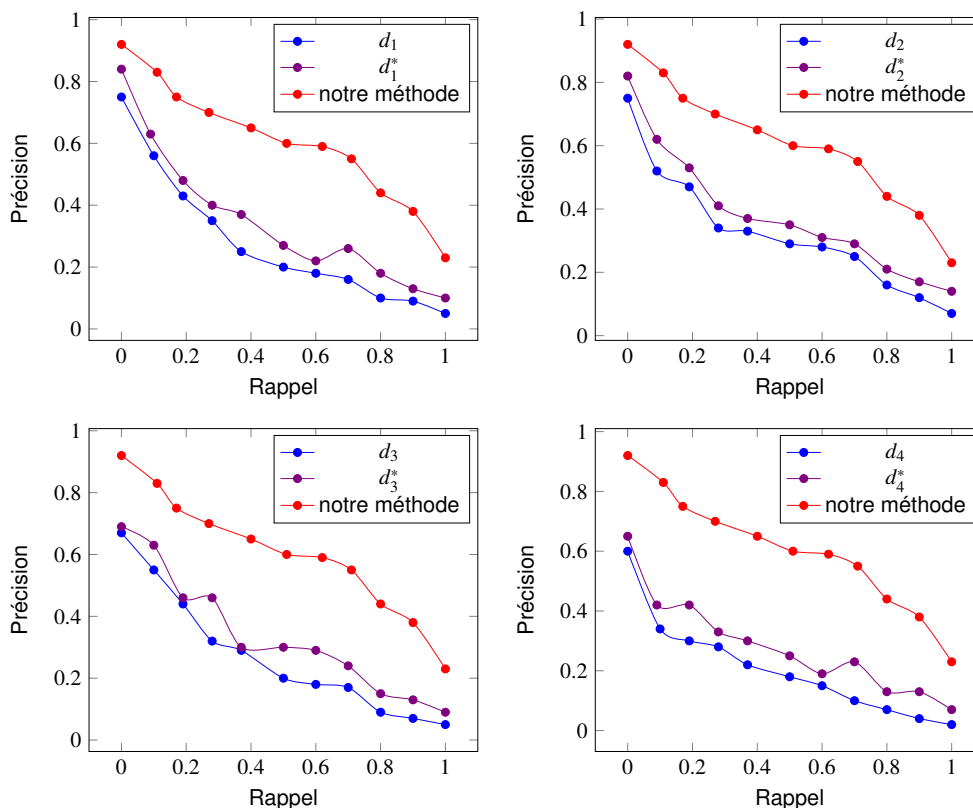


FIGURE 6.3 – Les courbes précision-rappel des éléments de la base de données TOSCA. Dans chaque graphique, les courbes bleu correspondent aux distances (d_i) induites par les descripteurs utilisés, et les courbes en violet correspondent aux distances géodésiques d_i^* . La courbe en rouge correspond aux résultats de notre méthode.

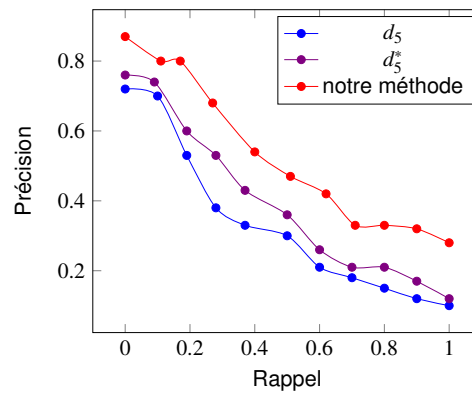


FIGURE 6.4 – Les courbes précision-rappel des éléments de la base de données ESB. Les courbes en bleu correspondent à la distance d_5 induite par le descripteur CH, et les courbes en violet à la distance géodésique d_5^* (longueur du plus court chemin) dans le graphe G_5 . La courbe en rouge correspond aux résultats de notre méthode.

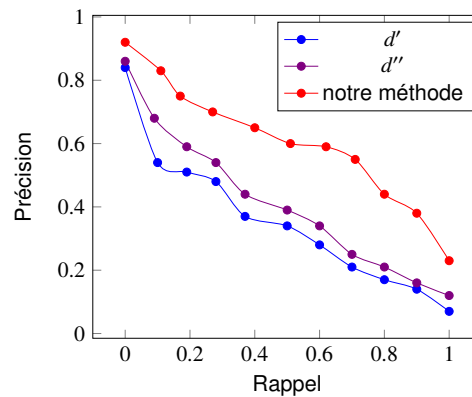


FIGURE 6.5 – Courbes de précision-rappel des objets de la base de données TOSCA. La courbe en bleu correspond aux résultats de combinaison des distances d_i , la courbe en violet des distances d_i^* . La courbe en rouge correspond aux résultats de combinaison des distances induites par les descripteurs LFD, GD, DE, et D2 en utilisant notre méthode.

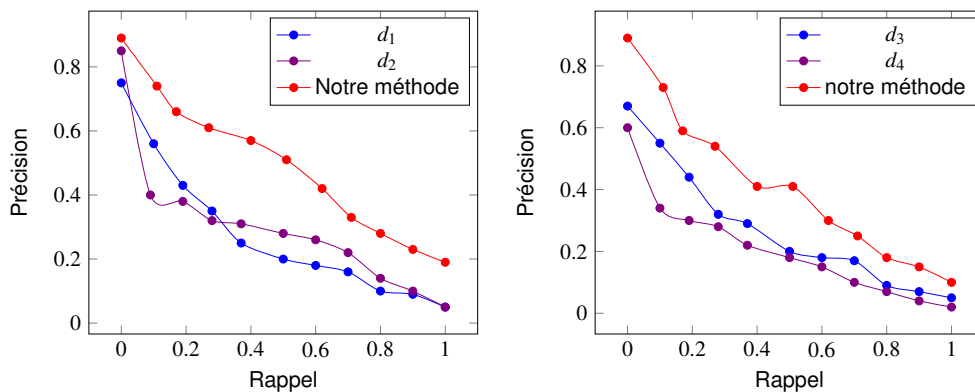


FIGURE 6.6 – Courbes de précision-rappel calculées pour des objets de la base de données TOSCA. Graphique à gauche : la courbe en rouge correspond à la combinaison des distance d_1 et d_2 avec notre méthode. A droite : la combinaison des distances d_3 et d_4 .

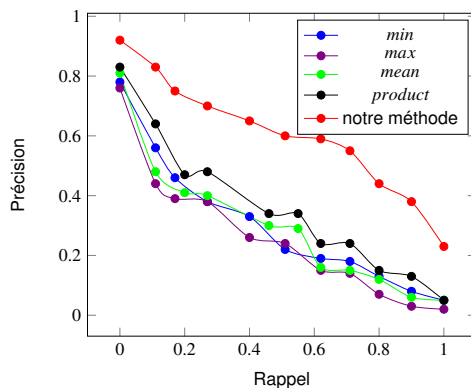


FIGURE 6.7 – Courbes de précision-rappel pour un ensemble d'objets de la base de données TOSCA, calculées en utilisant les différentes règles de fusion pour les distances d_i .

6.4.2/ RÉSULTATS ET COMPARAISONS

Dans cette section, nous donnons les résultats d'évaluation de notre méthode, ainsi que les résultats des comparaisons avec les différentes méthodes considérées dans ce chapitre. Les résultats sont exprimés en terme des courbes de précision-rappel. Dans cette section, les distances d_1, d_2, d_3, d_4 et d_5 font respectivement référence aux distances induites par les descripteurs : LFD, GD, DE, D2, et CH.

Les premiers résultats de tests concernent les comparaisons des distances d_i avec les distances d_i^* . Nous rappelons que la distance $d_i^*(A, B)$ est la longueur du plus court chemin entre A et B dans le graphe G_i . Les résultats de comparaison sont illustrés dans les Figures 6.3 et 6.4. Ces figures montrent une performance supérieure de notre nouvelle distance D comparée aux distances d_i^* .

En outre, pour comparer la performance des distances utilisées dans le contexte de la fusion, nous utilisons une combinaison linéaire des distances d_i^* . Nous notons $d' = \sum_{i=1}^n c_i \times$

d_i et $d'' = \sum_{i=1}^n c_i \times d_i^*$ la combinaison linéaire respectivement des distances d_i et d_i^* . Les coefficients c_i associés aux distances sont définis en terme des moyennes des précisions calculées sur un échantillon de $T = 100$ éléments sélectionnés aléatoirement dans la base de données. Nous avons obtenu $c_1 = 0.2648$, $c_2 = 0.3142$, $c_3 = 0.2232$, et $c_4 = 0.2078$ pour les distances d_1, d_2, d_3 et d_4 , et $c_1 = 0.2756$, $c_2 = 0.3042$, $c_3 = 0.2258$, et $c_4 = 0.1935$ pour les distances d_1^*, d_2^*, d_3^* et d_4^* . La Figure 6.5 montre les résultats de différents tests. Notre nouvelle distance D est meilleure.

La deuxième partie des tests concerne l'évaluation de la performance de notre nouvelle distance D , comparée aux distances d_i , ainsi que les différentes règles de fusion. La Figure 6.6 présente les résultats de la combinaison respective des distances d_1 et d_2 , et des distances d_3 et d_4 . Nous rappelons que notre méthode peut combiner n'importe quel nombre de distances, et non pas seulement deux distances. Les courbes illustrées dans ces figures montrent que la combinaison des distances améliore la précision du processus de recherche. Le Tableau 7.4 montre des exemples de résultats de précision pour des modèles sélectionnés aléatoirement dans les différentes classes de la base de données ESB. La combinaison des distances d_4 et d_5 par notre méthode (appelée D

dans le tableau 7.4) augmente leurs performances, pour tous les objets illustrés dans ce tableau.

TABLE 6.1 – Exemples des résultats de précisions pour des modèles choisis aléatoirement dans les différentes classes de la base de données ESB

Requête	Précision		Coefficients		Précision notre méthode (D)
	D2 (d_4)	CH (d_5)	α_4	α_5	
90 degree elbows	0.811	0.624	0.818546	0.181454	0.854
Bearling like parts	0.892	0.814	0.676229	0.323771	0.945
Container like parts	0.609	0.771	1	0	0.811
Cylindrical parts	0.614	0.842	0.0512844	0.948716	0.892
Discs	0.737	0.609	0.985294	0.0147057	0.811
Intersecting pipes	0.909	0.828	0.558892	0.441108	0.938
Long pins	0.846	0.602	0.743	0.256623	0.923
Miscellaneous	0.800	0.771	0.985294	0.0147057	0.845
Non-90 degree elbows	0.701	0.652	1	0	0.771
Posts	0.834	0.699	0.558892	0.441108	0.892
Pulley like parts	0.701	0.808	1	0	0.845
Round change at end	0.737	0.701	0.612684	0.387316	0.811
Simple pipes	0.938	0.689	0.558892	0.441108	0.955
Spoked wheels	0.669	0.925	0.985294	0.0147057	0.945
Bearing blocks	0.767	0.628	0.931738	0.0682622	0.846
Contoured surfaces	0.702	0.618	0.760019	0.239981	0.743
L Blocks	0.846	0.767	0.362104	0.637896	0.892
Machined blocks	0.589	0.702	0.79817	0.20183	0.735
Machined plates	0.907	0.767	0.439447	0.560553	0.945
Miscellaneous	0.874	0.811	0.0591479	0.940852	0.924
Motor bodies	0.702	0.853	0	1	0.907
Rocker arms	0.804	0.838	0.558892	0.441108	0.892
Slender links	0.708	0.734	0.879136	0.120864	0.797
Small machined blocks	0.907	0.863	0.805	0.195	0.924
T shaped parts	0.734	0.874	0.333112	0.666888	0.892
Thick slotted plates	0.673	0.79817	0.20183	0.816328	0.734
Backdoors	0.688	0.734	0.615845	0.384155	0.795
Bracket like parts	0.701	0.811	0.383436	0.616564	0.846
Contact switches	0.938	0.826	0.985294	0.0147057	0.955
Curved housings	0.892	0.528	0.0313121	0.968688	0.907
Miscellaneous	0.883	0.846	0.340035	0.659965	0.892
Rectangular housings	0.548	0.724	0.558892	0.441108	0.846
Thin plates	0.669	0.846	1	0	0.863

Nous comparons aussi la méthode proposée avec les différentes règles de fusion des distances d_i . La Figure 6.7 montre ces résultats en terme des courbes précision-rappel. Encore une fois, la nouvelle méthode est meilleure.

Les courbes de la Figures 6.8 montrent les résultats de tests pour tous les éléments des deux bases de données TOSCA et SHREC'11, en considérant différentes distances d_i .

Les graphiques de la Figure 6.9 montrent les résultats pour les trois catégories de la

base de donnée ESB. Les courbes présentées dans ces figures montrent une haute performance de la nouvelle distance D .

Ces expériences ont montré que la nouvelle distance D améliore toujours les résultats des distances originales d_i .

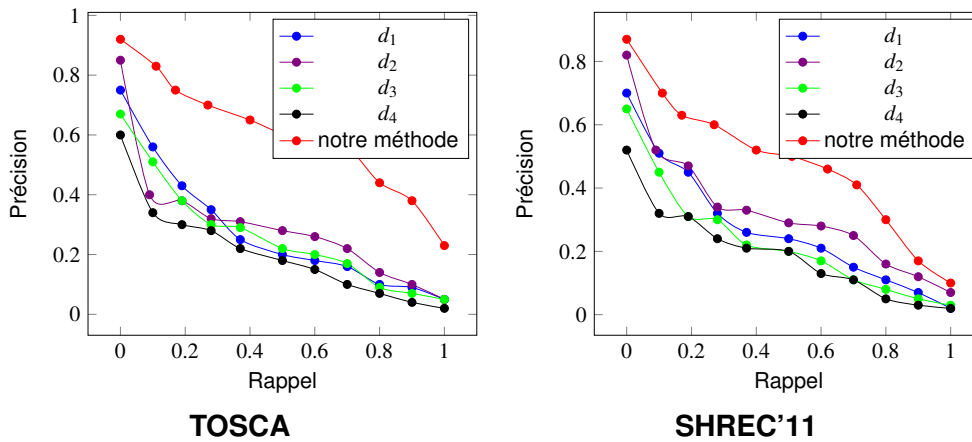


FIGURE 6.8 – Les courbes précision-rappel pour les éléments des bases de données TOSCA et SHREC'11 en utilisant notre méthode.

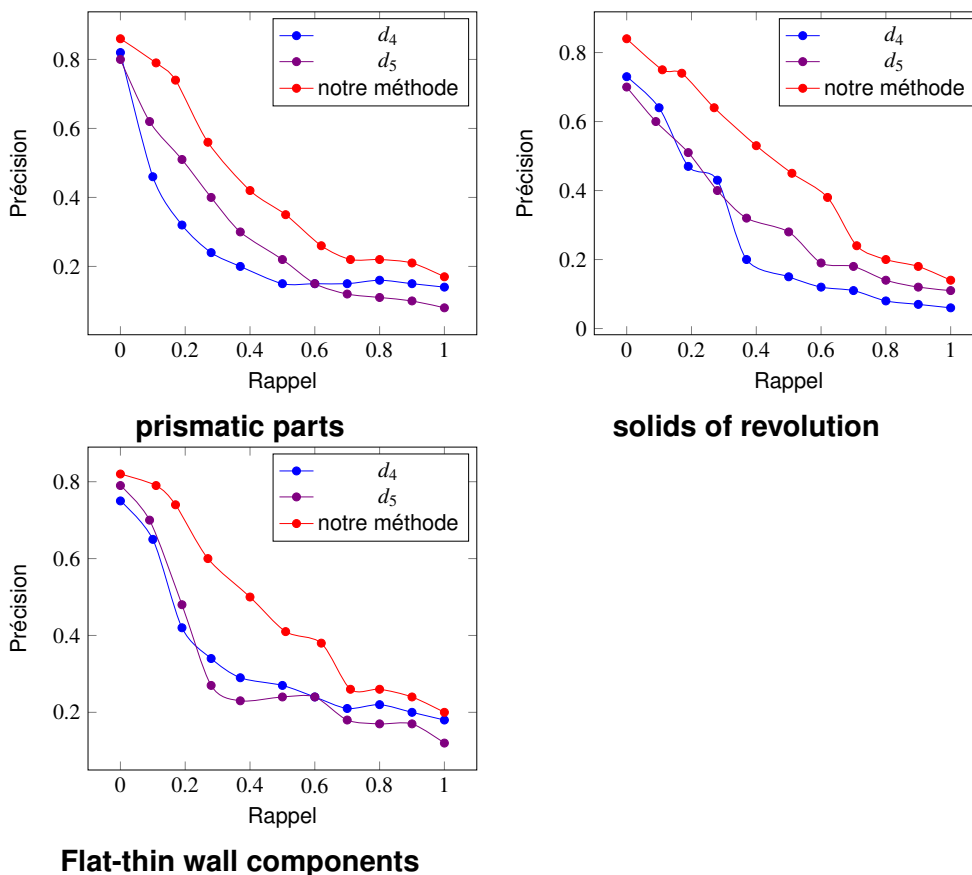


FIGURE 6.9 – Les courbes précision-rappel des modèles des trois catégories de la base de données ESB, en utilisant la combinaison des distances d_4 et d_5 .

6.5/ CONCLUSION

Nous avons présenté dans ce chapitre une nouvelle façon de combiner différentes distances induites par différents descripteurs de formes. Les résultats expérimentaux ont montré que cette nouvelle approche permet d'améliorer considérablement la performance des différents descripteurs, par le biais de leurs distances induites. Elle peut être généralisée pour être utilisée avec d'autres descripteurs. La nouvelle distance proposée est une combinaison convexe de n'importe quel nombre n de distances pré-existantes.

Les coefficients de cette combinaison sont calculés pour que le pouvoir de séparation (les potentiels) dans le graphe représentant une base de données soit maximal : elle maximise par programmation linéaire soit la somme des longueurs des plus courts chemins entre la requête et les objets de la base de données, soit la longueur du plus court chemin entre la requête et l'objet comparé à la requête. Les sommets du graphe sont les formes de la base de données ou la requête, et les arêtes sont étiquetées par plusieurs distances, une par descripteur.

La méthode est entièrement automatique : il n'y a pas besoin d'une connaissance préalable des classes des formes dans la base de données. La méthode proposée est en temps polynomial. Néanmoins, il y a un problème de complexité algorithmique que nous ne n'avons pas considéré jusqu'à présent : le nombre d'arêtes du graphe complet G croît quadratiquement avec le nombre des formes dans la base de données. C'est un sujet de recherche future.

LE CLONAGE AMÉLIORE LA RECONNAISSANCE

7.1/ INTRODUCTION

Rappelons que l'objectif du processus de recherche par forme clef est de trouver dans une base de données les objets similaires à une requête, en utilisant un descripteur de forme donné et sa distance induite. Toutefois, aucun descripteur de forme n'est suffisant pour accomplir correctement une telle tâche pour toutes les requêtes possibles. La façon standard pour surmonter ce problème est la fusion de plusieurs descripteurs de formes en un seul, c-à-d l'amélioration des descripteurs de formes.

Dans ce chapitre, nous proposons une alternative : au lieu d'améliorer les descripteurs de formes, nous améliorons la requête en utilisant un algorithme génétique. Cette méthode s'applique à tous les descripteurs de formes.

La requête donne naissance à des descendants, appelés clones. Les clones sont des copies de la requête, mais certains sommets sont légèrement perturbés. Un algorithme génétique (GA) fait évoluer les clones, de sorte qu'ils se rapprochent des formes dans la base de données (pour un descripteur de forme donnée et sa distance induite), bien que les clones et la requête restent indiscernables pour l'oeil humain.

Pour trouver les objets les plus similaires à la requête, nous utilisons donc l'objet lui même ainsi que ses clones, à chaque génération de l'algorithme génétique. Le meilleur clone (ou requête) est celui qui permet d'optimiser une fonction aptitude (en anglais *fitness*). Nous utilisons deux fonctions aptitudes.

Dans nos expériences, nous utilisons à titre d'exemple plusieurs descripteurs de formes. Les descripteurs de formes utilisés sont : *Vertex Normal Descriptor* (VND) [Thürner et al., 1998], *Discrete Energy* (DE) [Meyer et al., 2003] *Local Shape Distribution* (LSD) [Li et al., 2012], *Temperature Distribution* (TD) [Fang et al., 2011] et *Local Diameter Function* (LFD) [Gal et al., 2007]. Une étape de normalisation des formes est nécessaire avant de calculer les descripteurs sur les formes, ainsi que leurs distances induites.

Cette méthode peut être appliquée à toutes les techniques utilisées dans la recherche par forme clef pour calculer la dissimilarité $d(A, B)$ entre deux formes A et B . De plus, cette méthode est facile à mettre en oeuvre et massivement parallèle.

Le contexte de notre travail est présenté dans la Section 7.2 de ce chapitre. Nous dé-

crivons les concepts d'amélioration de la recherche, l'utilisation du hasard, ainsi que le principe de notre algorithme génétique utilisé pour améliorer la précision de recherche par forme clef. Dans la section 7.3, nous présentons en détails la technique de clonage de forme utilisé pour améliorer le processus de recherche. Une extension de notre algorithme génétique en une architecture distribuée est présentée dans la Section 7.4. Les résultats expérimentaux sont présentés dans la Section 7.5.

7.2/ CONTEXTE ET PRINCIPES

7.2.1/ L'AMÉLIORATION DU PROCESSUS DE RECHERCHE PAR FORME CLEF

Le calcul de dissimilarité des formes dépend fortement de la caractérisation des objets (quels caractères de l'objet sont utilisés ?), en l'occurrence les descripteurs de formes. Plusieurs descripteurs de formes ont été proposés dans la littérature, mais aucun de ces descripteurs n'est suffisant pour atteindre des résultats corrects de recherche pour toutes sortes d'objets [Gal et al., 2007, Dao et al., 2006b, Tangelder et al., 2008, Hu et al., 2009, Toldo et al., 2009]. Les approches classiques utilisées pour résoudre ce problème consiste à créer de nouveaux descripteurs en combinant plusieurs par des techniques de fusion [Akgül et al., 2008, Mc Donald et al., 2005, Chahooki et al., 2012, Kuncheva, 2002, Snoek et al., 2005]. Dans la plupart des méthodes de fusion, des règles de fusion ou des techniques d'apprentissage sont utilisées. Chahooki *et al.* [Chahooki et al., 2012] propose une méthode d'apprentissage pour fusionner des caractéristiques de l'objet afin d'améliorer le processus de recherche, en tenant compte de la surface et de l'intérieur de la forme. La méthode proposée par Akgül *et al.* [Akgül et al., 2008] combine des mesures de dissimilarité opérant sur différentes caractéristiques de la forme. Cette méthode calcule la combinaison optimale de ces mesures de dissimilarité en minimisant le critère de risque d'erreur de classement empirique. Surmonter les limites de certaines méthodes pré-existantes est une autre façon d'améliorer le processus de recherche. La méthode présentée dans [Bronstein et al., 2010] assure l'invariance aux changements d'échelle pour le descripteur du noyau de la chaleur proposé précédemment dans [Sun et al., 2009]. La méthode proposée dans [Ling et al., 2005] a pour but de rendre le contexte de forme [Belongie et al., 2000] résistant aux articulations : elle remplace la distance euclidienne, utilisée pour calculer le descripteur du contexte de la forme, par la distance géodésique. D'autres méthodes calculent la distance entre des formes dans le contexte de toutes les autres pour améliorer la précision de la recherche : la base de données est représentée par un graphe, dont les sommets sont les objets de la base de données, et les arêtes portent les distances entre les objets. Ce graphe est appelé graphe de transduction. La distance entre les formes est donc définie comme la longueur du plus court chemin dans le graphe associé à la base de données. La méthode proposée dans [Yang et al., 2008] permet d'améliorer la mesure de dissimilarité en utilisant le graphe de transduction.

7.2.2/ L'UTILISATION DU HASARD, ET DU BRUIT

Les propriétés du hasard sont utilisées dans de très nombreux algorithmes stochastiques, ou probabilistes. Citons, sans être exhaustif :

- La randomisation : c'est un choix aléatoire. Un exemple est le tri rapide : le pivot est choisi au hasard [Cormen, 2009].
- La factorisation des grands entiers avec la méthode rho de Pollard. Cette méthode utilise le paradoxe des anniversaires : il suffit de tirer n éléments au hasard parmi n^2 éléments pour que le tirage deux fois du même élément devienne probable ; plusieurs algorithmes pour résoudre le problème du logarithme discret (dont celui du petit pas, grand pas, *small step, big step*) utilisent le même principe.
- La planification de trajectoire pour la robotique, par carte probabiliste [Latombe, 2012].
- L'intégration par méthode de Monte Carlo.
- Le recuit simulé (en anglais *simulated annealing*), et d'autres méta-heuristiques ou algorithmes évolutionnaires.
- Les tables de hachage [Donald, 1999].
- La méthode de Rabin-Karp [Karp et al., 1987] pour la détection d'une chaîne de caractères dans un texte.
- Hors de l'informatique, les sondages d'opinion, électoraux par exemple.

La perturbation aléatoire est aussi utilisée dans le domaine de l'arithmétique stochastique [Vignes, 1993], pour améliorer la robustesse des calculs numériques. Dans le phénomène de résonance stochastique [Rousseau et al., 2006, Chapeau-Blondeau et al., 2009], le bruit améliore la transmission des informations et la détection des signaux faibles. Récemment, les méthodes présentées dans [Karbasi et al., 2014, Chen et al., 2014] montrent que les calculs bruités améliorent les mémoires associatives (des réseaux de neurones artificiels multi-couches). En géométrie algorithmique, des petites perturbations aléatoires de données géométriques enlèvent avec une probabilité de 1 toutes les dégénérescences comme, en 2D, trois points alignés ou quatre co-cycliques. Ceci simplifie considérablement les algorithmes géométriques, parce que seul un petit nombre de cas génériques doit être considéré, tandis que le nombre de cas dégénérés augmente de façon exponentielle avec la dimension géométrique du problème : considérer par exemple l'intersection de deux d -simplexes dans \mathbb{R}^{d+1} (deux segments dans le plan, deux triangles en 3D, etc). La littérature sur l'algorithmique probabiliste déborde.

Toutefois, le hasard ou le bruit semble rarement utilisé pour améliorer la reconnaissance et le processus de recherche par forme clef. Le bruit, le hasard sont habituellement considérés comme une source d'erreurs, une gêne, plutôt que comme une opportunité ou une ressource. Néanmoins, dans certains travaux de recherches, la perturbation ou le bruit ont été utilisés comme une stratégie efficace permettant d'améliorer les descripteurs de forme, et par conséquent la précision du processus de recherche. Le travail présenté dans [Jiang et al., 2009] propose une technique de perturbation pour améliorer la fiabilité du calcul de similarité entre les objets. Une autre technique basée sur la perturbation est utilisée dans [Thompson et al., 2010] pour la reconnaissance de l'iris de l'oeil. Cette technique de segmentation tente d'extraire l'iris à partir d'une image, en trouvant des limites circulaires qui s'approchent du cercle entourant l'iris. Une perturbation est effectuée en changeant une ou plusieurs des valeurs des paramètres de la segmentation. Une autre méthode basée sur la perturbation est utilisée pour améliorer la performance de reconnaissance des caractères manuscrits chinois dans [Yin et al., 2013]. Cette méthode est basée sur une technique de perturbation pour identifier les déformations des caractères.

7.2.3/ LES ALGORITHMES GÉNÉTIQUES

L'algorithme génétique est inspiré des phénomènes biologiques liés à la génétique et à l'évolution naturelle [Goldberg et al., 1988, Holland, 1975] : sélections, croisements, mutations, etc. Dans l'organisme humain, les cellules comportent une série de chaînes ADN appelés *chromosomes*. Le rôle de ces chromosomes est de coder les fonctionnalités et les spécificités de l'organisme, et de représenter des éléments essentiels dans la phase de reproduction, afin de faire évoluer de manière progressive les populations au sein d'un organisme donnant lieu à plusieurs opérations génétiques. Chaque chromosome est constitué d'un ensemble de *gènes*, appelé génotype. Lors de la phase de reproduction, des opérations génétiques donnent lieu à la fusion des chromosomes pour assurer l'évolution des populations. Les différentes opérations génétiques sont :

- **La sélection** : c'est le processus naturel permettant de déterminer les individus les plus adaptés, et qui permet d'obtenir les meilleures progénitures. Les individus les plus adaptés ont plus de chance de se reproduire. La sélection dépend du phénotype.
- **Le croisement** : cette opération génétique permet d'échanger des parties entre deux *chromosomes*, pour donner lieu à de nouveaux *chromosomes*. Les deux *chromosomes* échangent des *gènes* selon un point de croisement.
- **La mutation** : c'est un phénomène rare qui permet de modifier des *gènes* d'une façon aléatoire. Certains algorithmes génétiques n'utilisent que la mutation, et aucun croisement.

Ce principe de l'évolution naturelle a donné naissance à des algorithmes évolutionnaires, qui peuvent être utilisés pour résoudre des problèmes d'optimisation. L'algorithme génétique est utilisé dans différents domaines tels que la conception des réseaux de neurones artificiels [Harp et al., 1989], la vision par ordinateur [Bhanu et al., 1991], l'apprentissage automatique [Grefenstette, 1988, De Jong, 1988], la segmentation d'images [Bhanu et al., 1995, Melkemi et al., 2006], etc.

En effet, l'utilisation des algorithmes génétiques remonte aux années soixante, à l'origine aux travaux de recherche de John Holland sur la transposition de la génétique aux systèmes artificiels [Holland, 1992]. L'ouvrage de David Goldberg [Goldberg, 1989] sur l'utilisation des algorithmes évolutionnaires dans la recherche, l'optimisation et l'apprentissage automatique a permis une large diffusion des algorithmes génétiques. Les algorithmes génétiques se basent sur le principe de Charles Darwin : "la survie du plus apte" ("*the survival of the fittest*") pour résoudre un problème en faisant évoluer un ensemble de solutions, appelé population, vers un optimum. Un algorithme génétique est un processus itératif qui fait évoluer une population (solutions possibles). Il associe à chaque individu (*chromosome*) de chaque population une valeur, permettant de mesurer sa qualité (son aptitude). A chaque itération, appelée une génération, l'aptitude de chaque individu de la population précédente est évaluée, et la nouvelle population est formée en se basant sur les valeurs de l'aptitude des individus de la génération précédente. Les individus les plus adaptés ont plus de chance de se reproduire que les individus les moins adaptés. La reproduction utilise les mécanismes du croisement et de la mutation. Ce processus se répète jusqu'à l'optimum, ou après un nombre donné d'itérations. Ce processus est résumé dans la figure 7.1.

Les algorithmes génétiques ont été utilisés comme une stratégie efficace pour améliorer la précision de la reconnaissance et du processus de recherche par forme clef, tels que les systèmes de recherche d'information [Fan et al., 2004], et la recherche de docu-

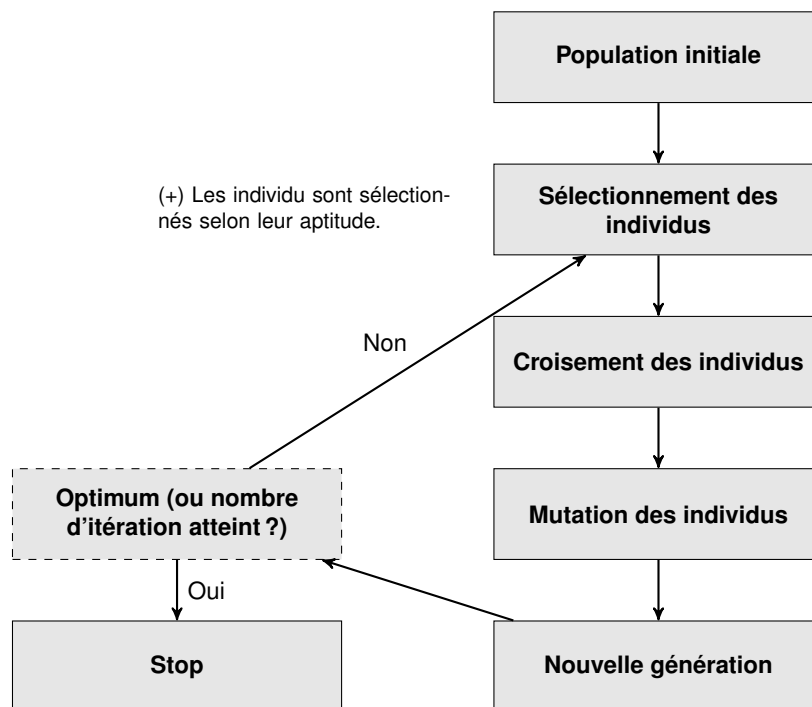


FIGURE 7.1 – Processus de l’algorithme génétique.

ments sur internet [Thada et al., 2013]. Dans le domaine de la vision par ordinateur, les algorithmes génétiques sont également utilisés avec des algorithmes de mesure de dissimilarité des données visuelles. De nombreuses approches basées sur les algorithmes génétiques sont utilisées pour l’amélioration des recherches par contenu visuel, ou CBIR (*Content-Based image Retrieval*). Dans [Syam et al., 2013], les auteurs proposent une mesure de dissimilarité basée sur un algorithme génétique : l’algorithme génétique intègre des caractéristiques d’image distinctes afin de trouver les images qui ressemblent le plus à une image donnée. La méthode de recherche par contenu visuel basée sur un algorithme génétique proposée dans [Aparna, 2013] vise à fusionner les mesures de dissimilarité : elle calcule le poids adéquat associé à chaque mesure de dissimilarité. Une autre méthode [Chan et al., 1999] combine différents vecteurs caractéristiques en utilisant un algorithme génétique afin de trouver les valeurs appropriées des poids associés aux caractéristiques prises en compte.

7.2.4/ PRINCIPE DE NOTRE MÉTHODE

Dans le domaine de l’appariement et de la recherche de formes, des artefacts visuellement imperceptibles ou des irrégularités des maillages 3D empêchent les objets d’être correctement reconnus. Pour résoudre ce problème, nous proposons de perturber le maillage 3D de la requête afin que celle-ci se rapproche de ses m objets les plus proches dans la base de données. Ces perturbations sont optimisées avec l’utilisation d’un algorithme génétique. La classe des objets dans la base de données n’a pas à être connue, et la classe de la requête non plus, bien sûr.

La méthode proposée fonctionne comme suit. Soit Q une requête et O un objet dans la base de données ; les deux sont décrits par des maillages triangulaires ; de plus, ils sont

normalisés (la somme des aires de tous les triangles vaut 1). La distance entre Q et O est $d(Q, O)$. Pour simplifier, nous disons que $d(Q, O)$ est une distance entre Q et O , même si d peut ne pas satisfaire l'inégalité triangulaire. Nous supposons que d est non négatif, et que $d(S, S)$ est zéro pour tout objet S . Pour plus de commodité, nous supposons également que $d(Q, O)$ appartient à l'intervalle $[0, 1]$. Un algorithme génétique est appliqué à la forme Q pour générer un ensemble de clones. Nous notons $Q = \{Q_0 = Q, Q_1, Q_2, \dots, Q_{K-1}\}$ l'ensemble des clones et la requête $Q_0 = Q$. Les clones Q_i sont visuellement indiscernables de la requête Q . Soit Q^* le meilleur clone dans Q . La distance $d(Q, O)$ est donc $d(Q^*, O)$. La distance améliorée $d(Q^*, O)$ ne peut pas être supérieure à $d(Q, O)$. Ainsi, l'algorithme génétique ne peut que diminuer la distance d . L'algorithme génétique crée $G = 15$ générations de $K = 20$ clones. $P_1 = \{Q_1, \dots, Q_{K-1}\}$ est obtenue par mutation de Q . Ces K mutants Q_1, Q_2, \dots, Q_{K-1} sont similaires à A , mais les coordonnées 3D d'un sous-ensemble aléatoire de leurs sommets sont légèrement et aléatoirement perturbées. Pour renouveler la population, les opérateurs génétiques standard sont appliqués aux individus sélectionnés pour générer de nouveaux clones. Ces nouveaux clones remplacent leur parents dans la génération suivante. A chaque itération de l'algorithme génétique, et pour chaque Q_i de la population courante, l'ensemble $M(Q_i)$ des m objets de la base qui sont les plus proches de Q_i est calculé. Chaque Q_i est associé avec une fonction d'aptitude $f(Q_i)$. Par conséquent, Q^* est le clone Q_i avec la meilleure $f(Q_i)$ dans la génération actuelle. Q^* et son ensemble $M^* = M(Q_i)$ sont enregistrés. Finalement, $M(Q)$ est l'ensemble associé au meilleur clone Q^* dans toutes les générations.

Notez qu'il n'y a aucune connaissance préalable de la classe d'objets de la requête, et aucune connaissance préalable des classes de la base de données. La connaissance de la classe est seulement utilisée pour mesurer les performances (rappel et de précision) de notre méthode.

Une autre façon possible d'améliorer le processus de recherche est de lisser la forme de la requête, ainsi que les formes de la base de données. Le lissage est simple et rapide ; il est en particulier plus rapide que l'algorithme génétique. Une brève discussion comparant les performances du lissage et de l'algorithme génétique est donnée dans la partie expérimentale.

La section suivante présente en détails notre méthode.

7.3/ CLONAGE DE FORMES ET ALGORITHME GÉNÉTIQUE

7.3.1/ INITIALISATION DE L'ALGORITHME GÉNÉTIQUE : CLONAGE DE FORMES

Chaque objet est un maillage triangulaire ayant V sommets. Pour cloner un maillage à partir d'une requête Q , nous générons un ensemble Q de K clones. Le premier clone est la requête Q elle-même ; les autres $K - 1$ clones Q_1, \dots, Q_{K-1} sont les clones calculés à partir de la requête Q . Les clones sont obtenus en perturbant quelques points sur la surface de la requête Q .

Soit P la probabilité qu'un sommet $v = (v_x, v_y, v_z)$ soit légèrement perturbé, et V le nombre de sommets de la requête. Le nombre de sommets à perturber est noté $\mu = \lceil PV \rceil$. C'est le même pour tous les clones de Q . Les indices des sommets perturbés sont stockés dans un tableau non trié de μ éléments. Si un sommet v est perturbé, ses nouvelles

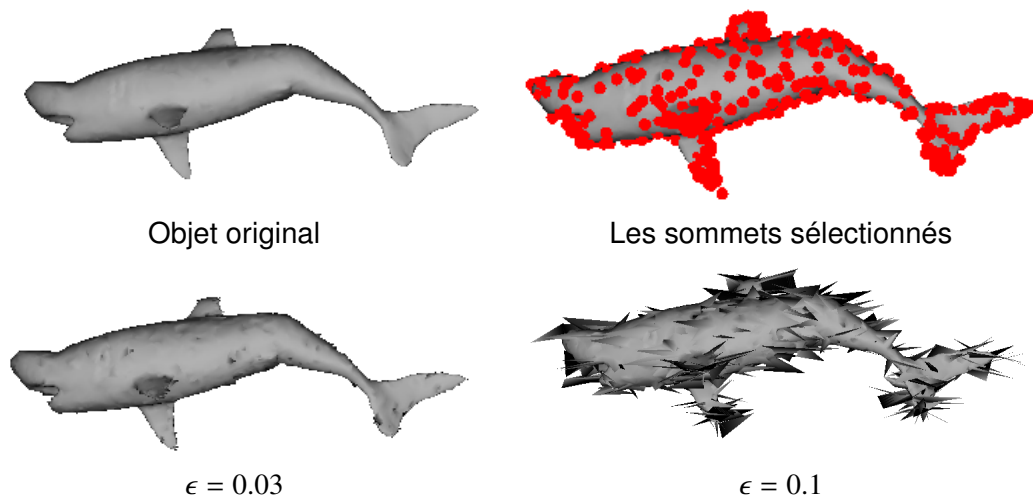


FIGURE 7.2 – Un exemple de la technique de perturbation appliquée sur un modèle (Shark, TOSCA [YA. M. Bronstein, 2007]). En haut à gauche : le modèle original. En haut à droite : les sommets sélectionnés pour être perturbés. En bas à gauche : les sommets sélectionnés perturbés avec $\epsilon = 0.03$. En bas à droite : les sommets sélectionnés perturbés avec $\epsilon = 0.1$.

coordonnées sont :

$$v'_x = v_x \pm \epsilon_x, \quad v'_y = v_y \pm \epsilon_y, \quad v'_z = v_z \pm \epsilon_z \quad (7.1)$$

où ϵ_x , ϵ_y , et ϵ_z sont trois nombres pseudo-aléatoires uniformément répartis dans l'intervalle $[-\epsilon, +\epsilon]$. D'une part, la valeur du paramètre ϵ doit être assez petite pour que les clones soient visuellement très proches de la requête, et d'autre part, une valeur nulle de ϵ , ou une valeur trop faible, ne peut pas améliorer la requête. Il est donc très important de bien estimer la valeur d' ϵ .

La Figure 7.2 montre plusieurs configurations pour un modèle 3D (*Shark*, TOSCA [YA. M. Bronstein, 2007]), pour différentes valeurs d' ϵ . A partir de cette figure, nous pouvons constater que pour $\epsilon = 0.03$ (en bas à gauche de la Figure 7.2), le clone ressemble toujours au modèle original (sous-figure en haut à gauche de la Figure 7.2), tandis que pour $\epsilon = 0.1$, le clone ne ressemble plus au modèle original (grands changements sur la surface).

La Figure 7.3 montre pour les différents descripteurs considérés, la distance entre un modèle original David 1 et les clones d'une autre posture du même modèle (David 2). Les clones du modèle David 2 sont générés à partir du modèle original (David 2) pour différentes valeurs d' ϵ . Les valeurs de l'axe des abscisses dans les graphiques de la Figure 7.3 correspondent aux différents clones. A partir des graphiques présentés dans la Figure 7.3, nous pouvons extraire deux informations : a) la distance diminue (selon les descripteurs, ex. pour les descripteur VND $\epsilon \in]0, 0.07]$ et $P = 0.25$ $d = 0.08$), elle est inférieure à la distance entre les deux objets eux mêmes.

b) ensuite la distance entre un clone et le modèle (David 1) augmente quand la valeur d' ϵ augmente ;

Cela signifie qu'il y a des clones qui permettent d'avoir une plus petite distance par rapport au modèle lui même (David 2) ; dans ce cas, les clones sont reconnaissables. Cependant, au-delà d'une certaine valeur ($\epsilon > 0.07$, pour le VND), la dissimilarité entre les

formes augmente de façon remarquable.

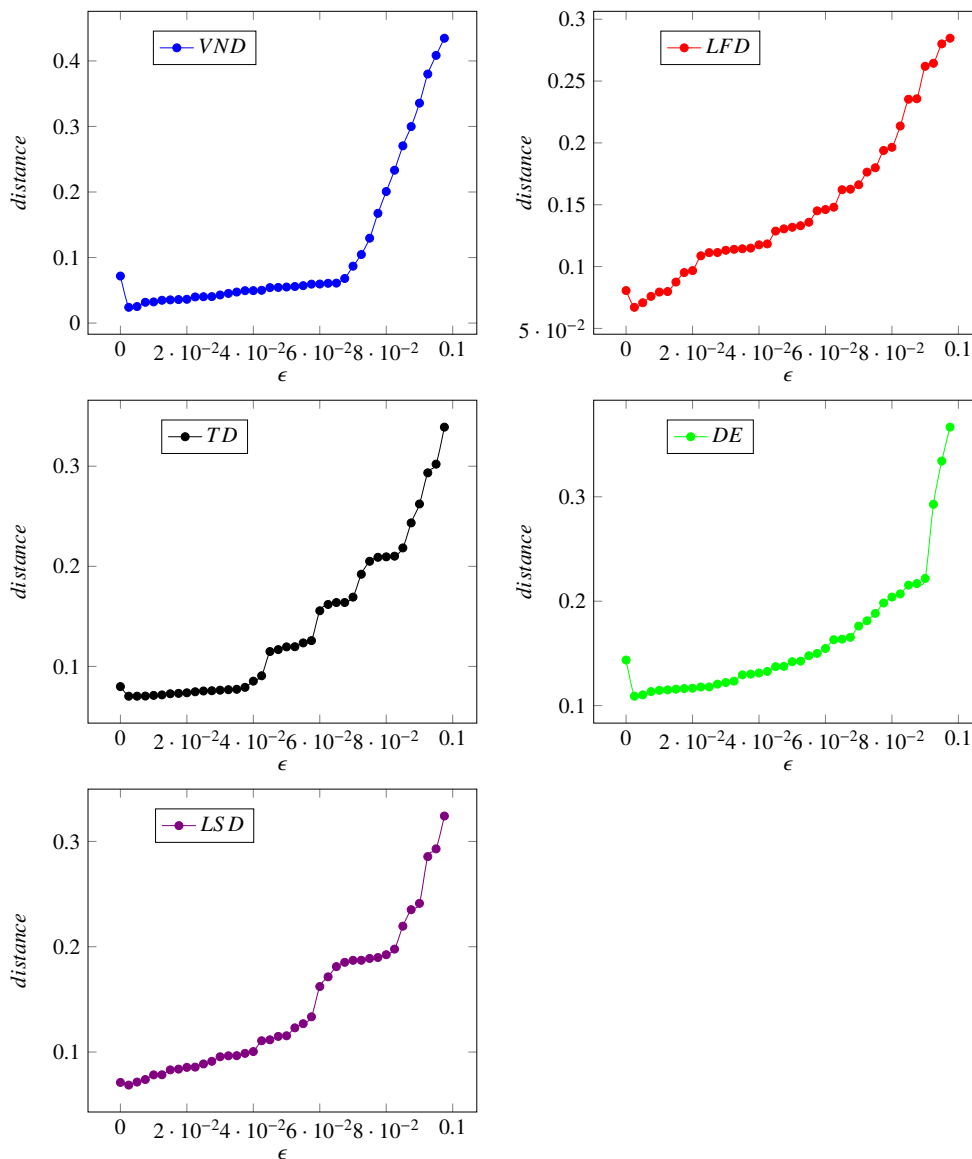


FIGURE 7.3 – Les distances entre un modèle David 1 et les clones d'un autre modèle David 2 générés en considérant différentes valeurs d' ϵ , et en utilisant les descripteurs VND, LFD, TD, DE, et LSD. En abscisse, la valeur d' ϵ ; en ordonné, la distance entre un modèle (David 1) et des clones d'un autre modèle, David 2, de la même classe. Pour $\epsilon = 0$, la distance n'est pas nulle, et vaut la distance entre David 1 et l'original (non perturbé) David 2. Pour de petites valeurs de ϵ , la distance est inférieure à D_0 , et augmente lentement, proportionnellement à ϵ , et fait par atteindre D_0 . Puis, au delà d'une certaine valeur, la distance augmente avec une perte nettement plus grande.

En effet, le degré de perturbation ϵ des sommets lors de génération des clones joue un rôle très important dans l'amélioration du processus de recherche, en utilisant l'objet lui même ainsi que ces clones comme requêtes. La Figure 7.4 présente la précision moyenne des objets de la base de donnée TOSCA [YA. M. Bronstein, 2007], en fonction des valeurs d' ϵ . Nous pouvons remarquer que les précisions varient en fonction des

valeurs d' ϵ , ainsi que des descripteurs utilisés. Dans la plupart des cas (VND, SD, DGC, DMC, et DE), des petites valeurs d' ϵ permettent d'améliorer la précision des descripteurs.

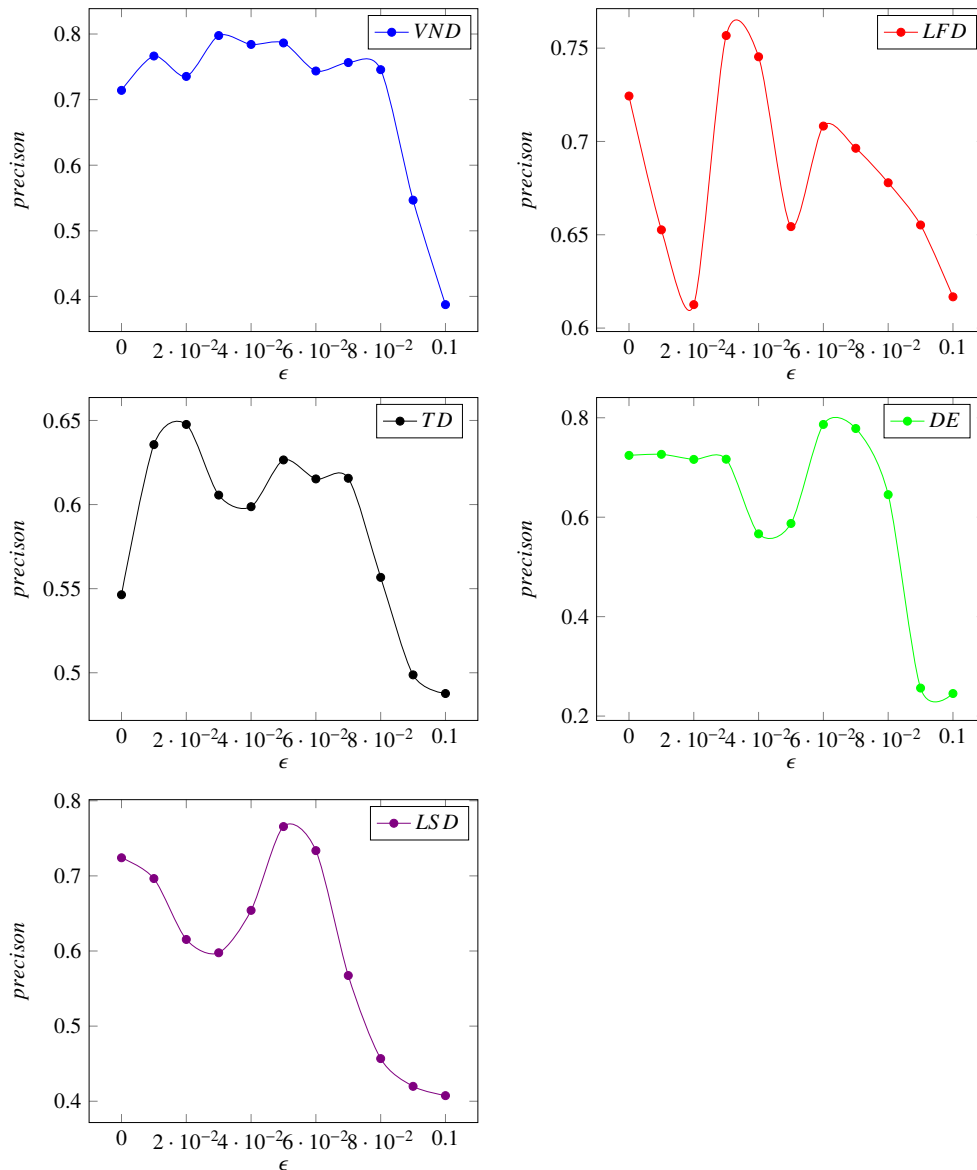


FIGURE 7.4 – La précision moyenne de toute la base de données (TOSCA [YA. M. Bronstein, 2007]), en fonction du coefficient de perturbation ϵ , en utilisant les différents descripteurs de formes.

En outre, la probabilité P de perturber un sommet est un autre paramètre important. La Figure 7.5 montre la variation de la moyenne des distances entre d'une part les clones de tous les objets dans la base de données, et d'autre part les m objets les plus proches de chaque requête. Les clones des objets sont obtenus en faisant varier le nombre de sommets perturbés $\mu = \lceil PV \rceil$ (axe des abscisses des graphiques). Les différents graphiques présentés dans cette figure (Figure 7.5) donnent les résultats de différents descripteurs utilisés dans notre travail. Nous remarquons que les distances varient de façon irrégulière. La meilleure valeur de μ est celle qui permet d'obtenir la plus petite distance ; elle dépend de chaque descripteur (ex. pour DMC, $P = 0.05$).

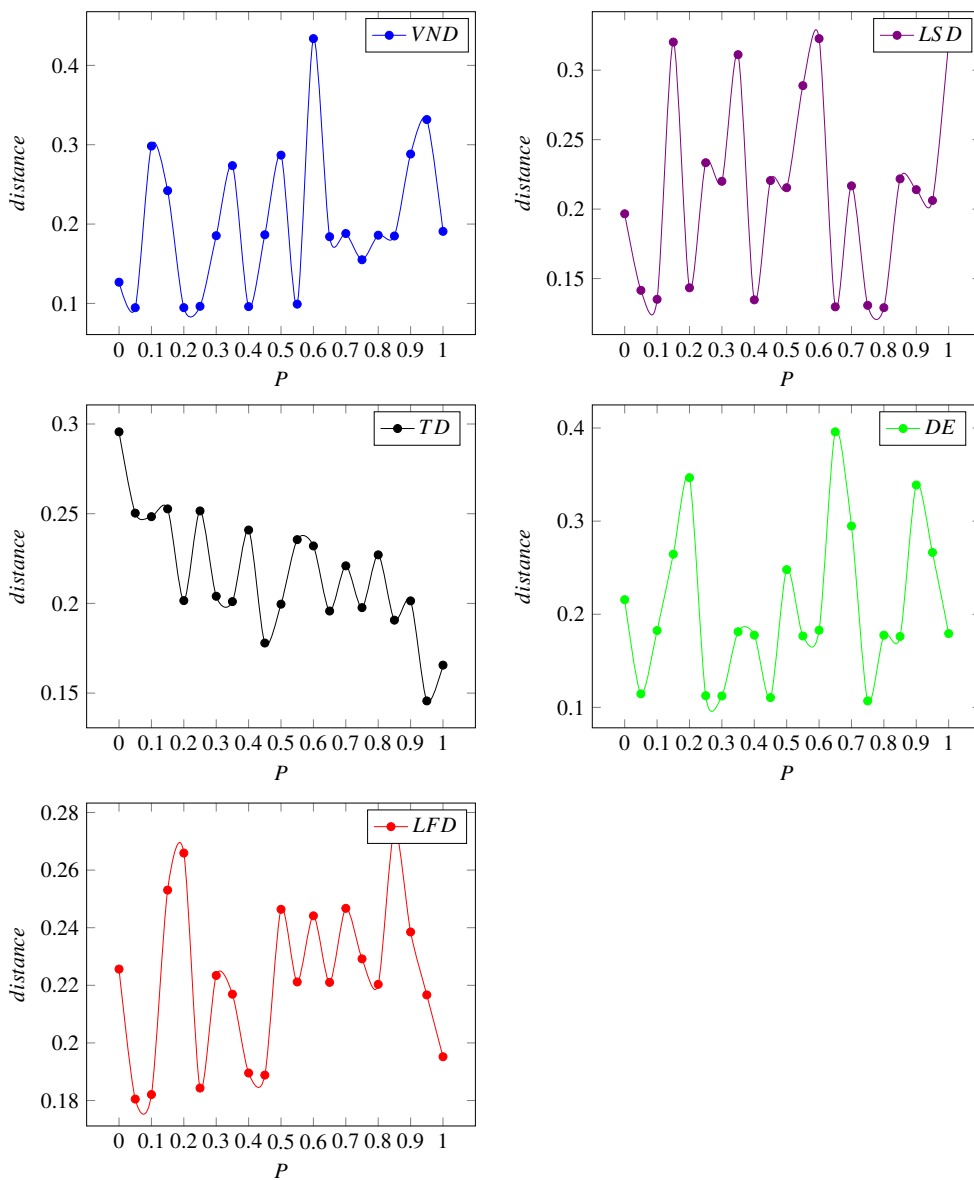


FIGURE 7.5 – La moyenne des distances (mesurée avec d) entre Q_i et d'une part, les clones de toutes les requêtes possibles dans la base de données ([YA. M. Bronstein, 2007]), et d'autre part les m formes dans la base de données les plus proches de Q_i . La courbe montre la variation de la moyenne des distances en fonction du nombre de sommets perturbés $\mu = [PV]$ (nous avons utilisé l' ϵ optimal pour chaque descripteur de forme, par exemple $\epsilon = 0.03$ pour le descripteur VND).

7.3.2/ LE CALCUL DES PARAMÈTRES ϵ ET P

Dans une phase de pré-traitement, les valeurs optimales des paramètres ϵ et P sont calculées pour chaque descripteur de formes utilisé. Nous utilisons la base de données TOSCA [YA. M. Bronstein, 2007]. Un sous-ensemble des modèles de cette base de données est sélectionné. Notez que cette étape a besoin de connaître la classe des objets dans la base de données pour enregistrer la précision du processus de recherche lors du calcul des paramètres ϵ et P .

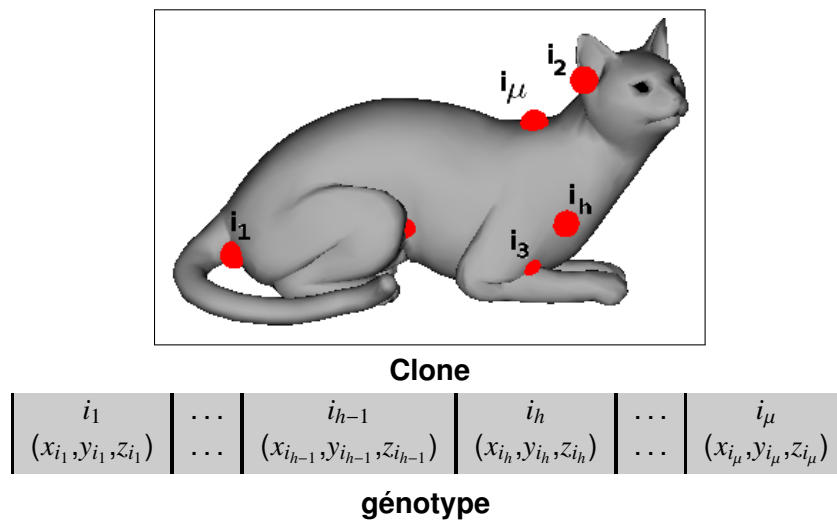


FIGURE 7.6 – Un exemple de codage du génotype d'un modèle : sommets perturbés en rouge (la sous-figure en haut), le génotype (sous-figure du bas).

La figure 7.4 montre les valeurs d' ϵ pour chaque descripteur de formes, enregistré avec la même valeur de $P \approx 0,25$. $P \approx 0,25$ est la meilleure valeur commune pour tous les descripteurs utilisés. Il est également possible d'affiner P pour chaque descripteur de formes. Les valeurs optimales d' ϵ pour VND, LFD, TD, DE, et LSD sont respectivement 0.05, 0.03, 0.02, 0.08 et 0.07.

7.3.3/ NOTRE ALGORITHME GÉNÉTIQUE

Nous présentons en détails dans cette section l'algorithme génétique utilisé.

La méthode calcule le clone optimal Q^* de la requête Q : Q^* est à distance minimale de ses m objets les plus proches dans la base de données. Nous avons essayé deux fonctions aptitudes.

7.3.3.1/ POPULATION DE L'ALGORITHME ALGORITHME

Le processus de l'algorithme génétique commence à partir d'une première population P_0 , et se termine après un certain nombre de générations G , avec une population finale P_{G-1} . La population initiale correspond à l'ensemble des clones $P_0 = \{Q_1, \dots, Q_{K-1}\}$ générés à partir de la requête Q . La représentation, ou le génotype de chaque clone, est un tableau non trié de taille μ . C'est un tableau à une dimension, contenant les indices i ainsi que les coordonnées 3D des sommets perturbés : (i, x_i, y_i, z_i) . Chaque tuple (i, x_i, y_i, z_i) du tableau représente un gène. Il est facile d'obtenir les coordonnées des sommets d'un clone à partir des coordonnées des sommets du modèle original et de son génotype. La Figure 7.6 montre un exemple d'un génotype obtenu à partir des indices des sommets perturbés sur la surface d'un clone calculé pour un modèle donné (Cat, TOSCA [YA. M. Bronstein, 2007]). Les valeurs numériques marquant les sommets rouges sur la surface de l'objet sont les indices des sommets perturbés.

TABLE 7.1 – Intervalle de sélection.

i	$f(Q_i)^{-1}$	l_i	Intervalle de sélection
1	0.31	0.202	[0, 0.202]
2	0.25	0.163]0.202, 0.365]
3	0.42	0.274]0.365, 0.639]
4	0.43	0.281]0.639, 0.92]
5	0.12	0.078]0.92, 1]

7.3.3.2/ OPÉRATEURS DE L'ALGORITHME GÉNÉTIQUE

A partir de la génération initiale, les individus (les clones) évoluent au cours des générations afin de produire de nouvelles générations d'individus (clones). Différentes opérations génétiques sont appliquées sur les clones. Les opérateurs de l'algorithme génétique sont :

- **La sélection.**
- **Le croisement.**
- **La mutation.**

En effet, la première étape dans le processus d'évolution génétique consiste à déterminer les couples de clones permettant de générer d'autres clones dans la génération suivante. L'aptitude $f(Q_i)$ des clones joue un rôle important dans le processus de sélection. Elle permet de déterminer les clones susceptibles de se reproduire. Un clone avec une meilleure valeur de $f(Q_i)$ a plus de chance de se reproduire.

Pour former une nouvelle génération de clones, nous sélectionnons à partir de la génération initiale $k/2$ paires de clones, en nous basons sur leurs valeurs de $f(Q_i)$. Soit F la somme des inverses des valeurs de $f(Q_i)$ de tous les clones dans la génération courante :

$$F = \sum_{i=1}^K f(Q_i)^{-1} \quad (7.2)$$

Remarque : nous considérons l'inverse de $f(Q_i)$ car nous souhaitons minimiser f .

Les $K/2$ clones sont sélectionnés avec une probabilité proportionnelle à f_i définie ci-dessous. Nous formons donc K intervalles, où la largeur l_i de chaque intervalle correspondant à un clone Q_i est donnée par :

$$l_i = \frac{f(Q_i)^{-1}}{F} \quad (7.3)$$

Pour sélectionner les paires de clones, nous tirons au hasard k valeurs comprises entre 0 et 1. Si la valeur sélectionnée appartient à un intervalle i , alors le clone Q_i est sélectionné. Le tableau montre un exemple calculant les intervalles de sélection selon les valeurs de $f(Q_i)$ de 5 clones.

A partir du tableau 7.1, et si la valeur sélectionnée aléatoirement est égale à 0.45 à titre d'exemple, alors c'est le clone 3 qui est sélectionné ($0.45 \in]0.365, 0.639]$).

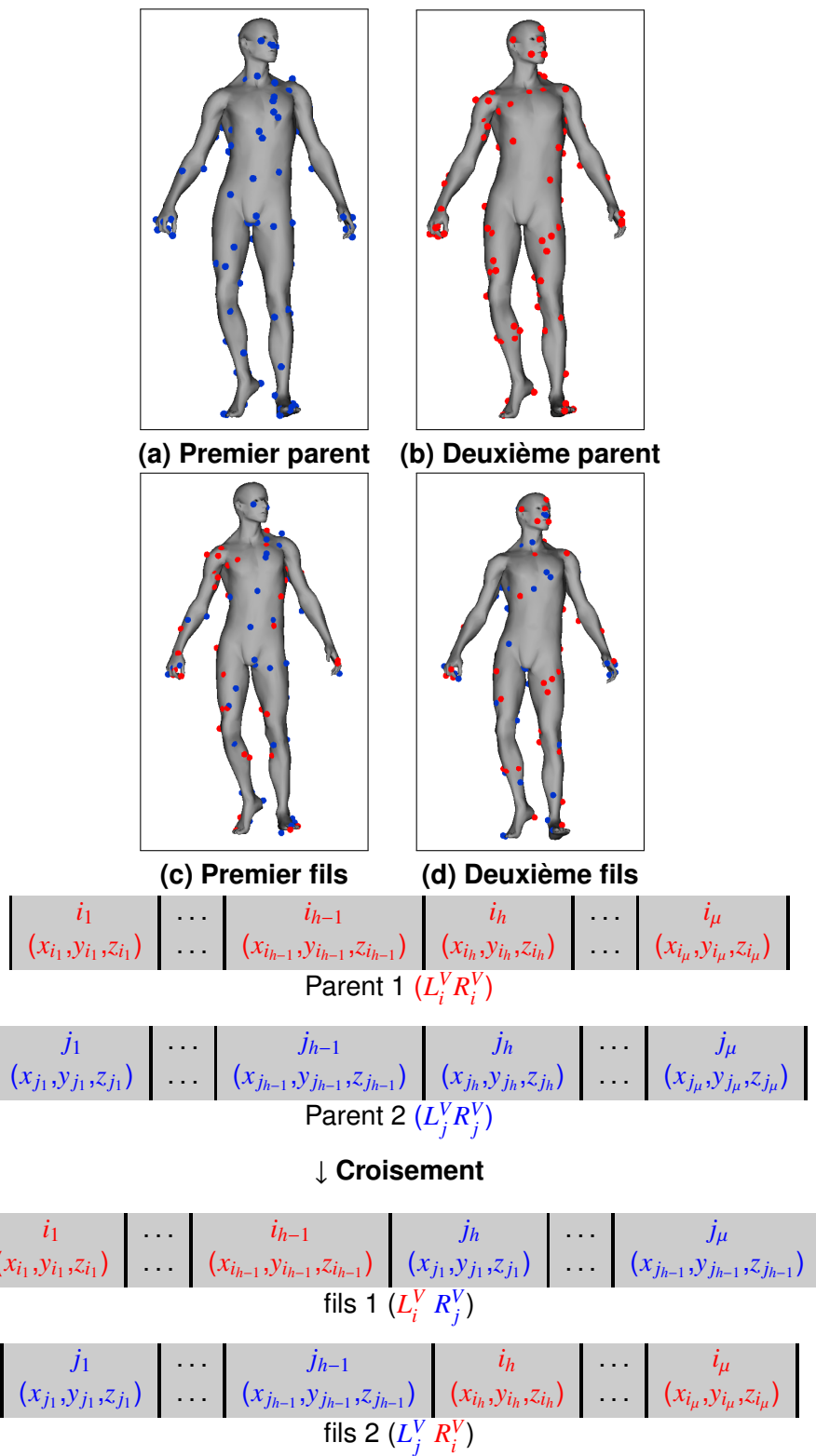


FIGURE 7.7 – Un exemple de l’opération de croisement pour deux clones d’une posture du modèle David (TOSCA [YA. M. Bronstein, 2007]).

Après avoir sélectionné les paires de clones pour le couplage, chaque paire de clones génère deux nouveaux clones en appliquant une opération de croisement entre leurs

deux génotypes. Le génotype d'un clone Q_i est noté Q_i^V . Q_i^V est la concaténation d'une partie gauche L_i^V et une partie droite R_i^V . Le croisement entre $Q_i^V = L_i^V R_i^V$ et $Q_j^V = L_j^V R_j^V$ donne deux nouveaux génotypes $L_i^V R_j^V$ et $L_j^V R_i^V$. Ce sont les deux enfants des clones Q_i et Q_j .

Pour chaque accouplement, la position de croisement est choisie au hasard (voir la Figure 7.7).

La mutation est un événement rare mais extrêmement important dans les algorithmes génétiques. Chaque clone peut subir une mutation : avec une probabilité de 0.01, chaque gène, c-à-d chaque sommet perturbé (i, x_i, y_i, z_i) est remplacé par un nouveau sommet $(j, x_j + \epsilon_x, y_j + \epsilon_y, z_j + \epsilon_z)$, où j est choisi au hasard et $\epsilon_x, \epsilon_y, \epsilon_z$ sont des valeurs pseudo aléatoires dans $[-\epsilon, \epsilon]$.

La Figure 7.8 montre un exemple d'une mutation d'un gène, où le sommet vert dans le modèle à gauche est enlevé après l'application d'une mutation et remplacé par un nouveau sommet (sommet vert dans le modèle à droite).

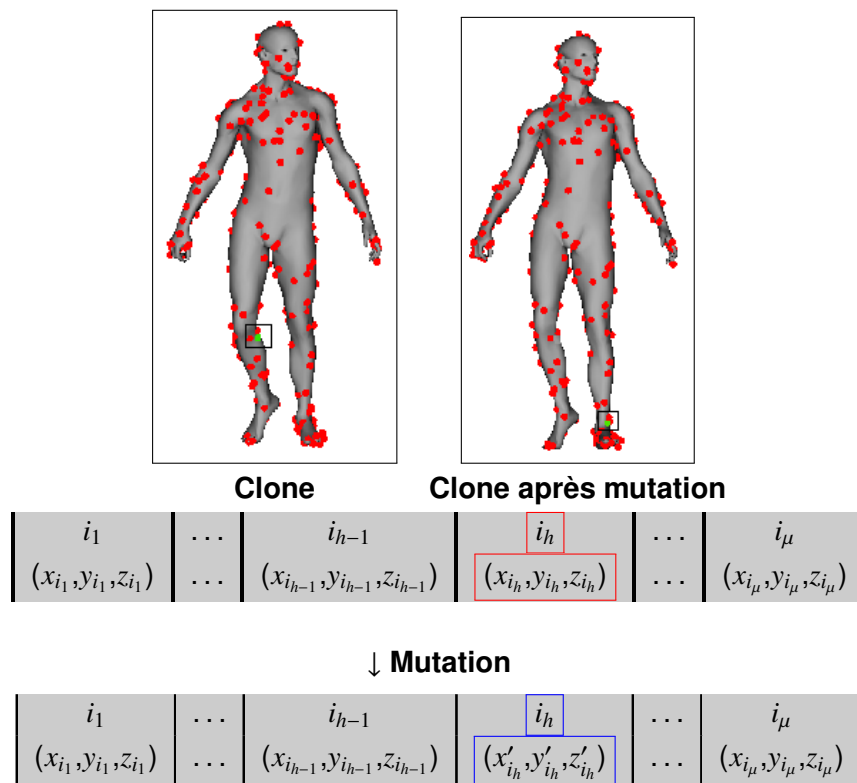


FIGURE 7.8 – Un exemple de mutation appliquée sur un modèle. Le sommet vert sur le modèle à gauche représente le sommet muté (remplacé par un nouveau sommet dans le modèle à droite).

7.3.3.3/ LA QUALITÉ D'UN CLONE : SON APTITUDE

L'aptitude (en Anglais *fitness*) est une fonction utilisée pour mesurer la performance ou la qualité d'une solution fournie par un algorithme génétique. Elle représente un paramètre très important permettant d'une part d'évaluer la qualité des solutions données par les

populations de l'algorithme génétique, et d'autre part de mesurer la qualité d'un génotype, ce qui détermine sa probabilité de se reproduire. Plusieurs fonctions d'aptitudes ont été proposées dans le domaine de la recherche d'information utilisant des algorithmes génétiques. Thada *et al.* [Thada et al., 2013] donnent une étude comparative des coefficients de dissimilarité utilisés pour trouver la meilleure valeur de l'aptitude, afin de chercher les documents textuels les plus pertinents pour un ensemble de mots clés. Un autre travail propose une nouvelle fonction approximative de l'aptitude dans le domaine de la recherche par mot [Radwan et al., 2008].

En effet, pour déterminer la valeur de l'aptitude d'un génotype, nous utilisons deux fonctions.

Definition 7. *Étant donnée une base de données \mathcal{B} , dont les éléments sont les objets $\{x_1, x_2, \dots, x_n\}$, pour un descripteur de formes donné, la distance entre deux objets x_i et x_j est noté $d(x_i, x_j)$.*

Nous proposons d'utiliser les deux fonctions suivantes.

Fonction d'aptitude #1 :

Nous notons $R(Q_i) := \{Q_i\} \cup M(Q_i)$ l'ensemble des m objets les plus proches la requête Q_i , et la requête Q_i elle-même. La première fonction $f(Q_i, d)$ est donnée par :

$$f(Q_i, d) := \frac{1}{m \times (m + 1)} \sum_{x_j \in R(Q_i)} \sum_{x_k \in R(Q_i)} d(x_j, x_k) \quad (7.4)$$

C'est la moyenne des distances entre les formes de $R(Q_i)$. Plus $f(Q_i, d)$ est petit, et meilleur est Q_i .

Fonction d'aptitude #2 :

La deuxième fonction d'aptitude est donnée par :

$$f(Q_i, d) := \frac{1}{m} \sum_{x_k \in M(Q_i)} d(Q_i, x_k) \quad (7.5)$$

C'est la moyenne des distances entre Q_i et les objets dans $M(Q_i)$. Les constantes multiplicatives $\frac{1}{m \times (m + 1)}$ et $\frac{1}{m}$ peuvent être ignorées.

Les précisions de recherche dans la base de données TOSCA, en utilisant les deux fonctions sont illustrées dans la figure 7.9. La première fonction aptitude est meilleure ; elle n'est pas forcément n fois plus coûteuse que la deuxième, car ce sont souvent les mêmes qui se trouvent dans $M(Q_i)$, et il est facile de stocker les distances entre formes de la base de données dans une table de hachage.

En effet, l'objectif de l'utilisation de l'algorithme génétique dans ce travail est de trouver le clone Q^* le plus proche des éléments $M(Q^*)$. Cela est réalisé en optimisant au fil des générations, la fonction d'aptitude. La meilleure solution de la recherche pour un clone Q_i correspond à celle qui donne une valeur de l'aptitude $f(Q_i, d)$ proche de zéro (minimal). La meilleure solution (meilleur clone) de l'algorithme génétique est le clone, dont l'aptitude est optimale. Les tableaux 7.3 et 7.5 montrent l'évolution des fonctions d'aptitude aux cours des générations. Les valeurs de la précision du processus de recherche correspondant sont illustrées dans les tableaux 7.2 et 7.4.

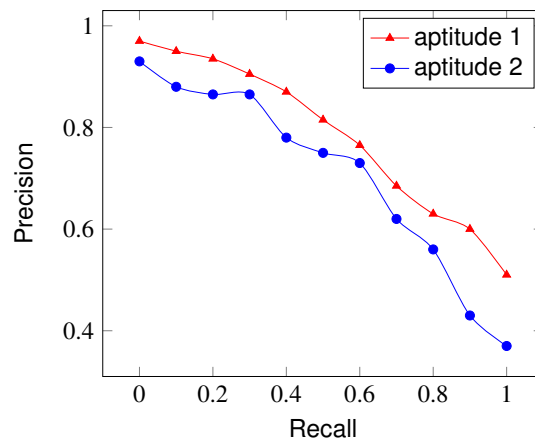


FIGURE 7.9 – Comparaison des précisions de notre algorithme selon les deux fonctions d'aptitude.

TABLE 7.2 – La précision du meilleur clone à chaque itération de l'algorithme génétique : la fonction aptitude #1 est utilisée. On indique en caractère gras les meilleures précision dans chaque génération.

Descripteur	Précision au cours des générations											
	1	2	3	4	5	6	7	8	9	10	11	12
VND	0.8125	0.75	0.6875	0.6875	0.875	0.75	0.937	0.6875	0.625	0.8125	0.95	0.6875
LFD	0.875	0.937	0.875	0.8125	0.5625	0.75	0.8125	0.875	0.75	0.625	0.8125	0.75
DE	0.75	0.625	0.75	0.875	0.625	0.5625	0.875	0.8125	0.6875	0.6875	0.6875	0.625
TD	0.8125	0.875	0.8125	0.75	0.75	0.8125	0.875	0.8125	0.75	0.75	0.8125	0.75
LSD	0.875	0.6875	0.8125	0.75	0.6875	0.75	0.937	0.875	0.8125	0.6875	0.8125	0.5625
MAS-SR-GA	0.875	0.937	0.875	0.875	0.875	0.75	0.937	0.875	0.8125	0.8125	0.95	0.75

TABLE 7.3 – L'évolution de la fonction d'aptitude du meilleur clone Q^* au cours des générations de l'algorithme génétique : la fonction aptitude#1 est utilisée.

descripteur	Les valeurs de la fonction aptitude#1											
	1	2	3	4	5	6	7	8	9	10	11	12
VND	0.142	0.204	0.349	0.219	0.085	0.282	0.022	0.383	0.312	0.122	0.035	0.345
LFD	0.108	0.056	0.128	0.136	0.366	0.224	0.12	0.114	0.22	0.35	0.092	0.240
TD	0.125	0.225	0.125	0.251	0.432	0.324	0.125	0.522	0.135	0.11	0.256	0.188
DE	0.284	0.398	0.308	0.110	0.353	0.384	0.081	0.342	0.335	0.21	0.356	0.388
LSD	0.090	0.375	0.138	0.199	0.420	0.339	0.007	0.152	0.213	0.320	0.095	0.405

Jusqu'ici, par simplicité, nous avons supposé qu'un seul descripteur, et donc une seule distance, était utilisé pour traiter une requête. Mais l'algorithme génétique proposé dans ce chapitre fonctionne tout aussi bien en utilisant plusieurs descripteurs, et leurs distances induites, pour traiter une requête. La recherche dans une base de données des m éléments les plus proches d'un clone Q_i pour une distance d rend un ensemble $M_d(Q_i)$. En passant, on comprend ici pourquoi il est important que toutes les distances soient normalisées. Les mêmes fonctions d'aptitude f_1 et f_2 sont utilisables. Remarquons que dans tout ensemble $M(Q_i)$, c'est toujours la même distance d qui est utilisée pour calculer la distance entre les éléments de $\{Q_i\} \cup M(Q_i)$.

TABLE 7.4 – La précision du meilleur clone à chaque itération de l’algorithme génétique : la fonction aptitude#2 est utilisée.

descripteur	Précision au cours des générations											
	1	2	3	4	5	6	7	8	9	10	11	12
VND	0.8125	0.6875	0.75	0.75	0.6875	0.75	0.875	0.6875	0.625	0.875	0.75	0.875
LFD	0.75	0.875	0.8125	0.8125	0.625	0.75	0.8125	0.75	0.75	0.8125	0.625	0.8125
TD	0.75	0.8125	0.75	0.8125	0.8125	0.875	0.75	0.75	0.8125	0.875	0.8125	0.75
DE	0.625	0.75	0.625	0.875	0.75	0.625	0.875	0.8125	0.625	0.75	0.6875	0.75
LSD	0.75	0.6875	0.8125	0.75	0.5625	0.75	0.8125	0.8125	0.8125	0.937	0.875	0.6875
MAS-SR-GA	0.8125	0.875	0.8125	0.875	0.8125	0.875	0.875	0.8125	0.8125	0.937	0.875	0.875

TABLE 7.5 – L’évolution de la fonction d’aptitude du meilleur clone Q^* au cours des générations de l’algorithme génétique : la fonction aptitude#1 est utilisée.

descripteur	Les valeurs de la fonction aptitude#2											
	1	2	3	4	5	6	7	8	9	10	11	12
VND	0.143	0.275	0.165	0.202	0.223	0.298	0.136	0.421	0.423	0.110	0.174	0.095
LFD	0.199	0.156	0.199	0.110	0.336	0.312	0.098	0.165	0.235	0.105	0.386	0.165
TD	0.112	0.254	0.245	0.095	0.140	0.152	0.156	0.254	0.287	0.145	0.0.135	0.154
DE	0.286	0.232	0.185	0.136	0.212	0.185	0.156	0.098	0.221	0.165	0.220	0.212
LSD	0.225	0.321	0.098	0.256	0.350	0.256	0.105	0.125	0.168	0.071	0.068	0.355

7.4/ ALGORITHME GÉNÉTIQUE DISTRIBUÉ POUR LE CLONAGE DE FORMES

Dans cette section, nous proposons une architecture multi-agents et parallèle pour notre algorithme génétique. Nous notons cette extension MAS-SR-GA (*Multi Agent System based on Genetic Algorithm for 3D Shape Retrieval*). MAS-SR-GA peut incorporer plusieurs descripteurs de formes présentés sous forme d’agents, qui travaillent en coopération afin d’obtenir une meilleure précision du processus de recherche. Il permet de bénéficier des avantages de chaque descripteur utilisé, et d’améliorer leur performance par leur utilisation conjointe.

Un système multi agent MAS (*Multi Agent System*) est un système composé de plusieurs modules autonomes appelés agents interagissant entre eux afin de résoudre une tâche donnée [Ferber, 1999, Florea, 1998]. La plupart du temps, ils utilisent des techniques issues de l’intelligence artificielle. Les MAS ont été largement utilisés dans diverses applications telles que la vision par ordinateur [Stone et al., 2000], les réseaux sociaux [Sabater et al., 2002], la gestion de la circulation et du transport [Chen et al., 2010, Burmeister et al., 1997], etc.

7.4.1/ DESCRIPTION DE L’ALGORITHME DISTRIBUÉ PROPOSÉ

Nous décrivons le MAS utilisé dans notre travail. Le MAS-SR-GA est composé d’un agent coordinateur AC et de K agents ou processus de recherche par forme : SR_1, SR_2, \dots, SR_K , $K = 20$. Ces K agents utilisent les descripteurs de forme dans $\mathcal{D} = \{ \text{VND, LSD, TD, DE, LFD} \}$. Par abus de notation, nous donnons le même nom d au descripteur de forme et à sa distance induite. Chaque agent SR_i traite une paire $(d \in \mathcal{D}, Q_i)$ où Q_i est un

clone généré à partir de la requête Q (ou la requête elle-même). L'agent SR_i calcule en fonction de la distance d l'ensemble $M(Q_i, d) = \{m_1, \dots, m_m\}$ des m formes dans la base de données les plus proches de Q_i (la base de données est le même pour tous les agents SR_i). Chaque agent SR_i envoie à l'AC le tuple : $(Q_i, d, M(Q_i, d), f(Q_i, d))$.

Nous présentons maintenant l'AC. Au début, l'AC lit Q , la forme de la requête. Il génère $K - 1$ individus $\{Q_1, \dots, Q_{K-1}\}$ de Q ($K = 20$), par mutation (expliqué dans la Section 7.3.3.3). $\{Q_0 = Q, Q_1, \dots, Q_{K-1}\}$ est la première population P_1 . La première population P_1 est spéciale, car elle est générée avec des mutations seulement.

l'AC entre dans une boucle (pour $g = 1$ à $G = 12$) :

- L'AC demande aux agents SR_i de traiter tous les paires (Q_i, d) , pour les descripteurs de formes $d \in \mathcal{D}$, tel que Q_i est un individu dans P_g . En effet, tous ces calculs sont indépendants et peuvent être réalisés en parallèle. Les agents SR_i envoient leurs résultats à l'AC.
- Après la réception des réponses envoyées par les agents SR_i , l'AC met à jour le tuple $(Q_i, d, M(Q_i, d), f(Q_i, d))$ du meilleur clone Q^* , c-à-d le clone avec la meilleure $f(Q, d)$.
- Si $g < G$, l'AC crée la génération suivante P_{g+1} en utilisant les opérateurs génétiques définis dans la Section 7.3.3.3 (la sélection, le croisement, et la mutation).

Après la boucle, le meilleur clone Q^* , ainsi que son tuple associé $(Q^*, d^*, M(Q^*, d^*), f(Q^*, d^*))$ sont désignés par l'AC.

L'algorithme 4 résume notre algorithme génétique distribué.

Algorithm 4 Le pseudo algorithme du MAS-SR-GA

Initialisation :

L'AC reçoit Q ;

for $i = 1$ to K **do**

Il perturbe et crée un nouveau clone Q_i ;

Il envoie Q_i et d aux SR_i ;

end for

for $g = 1$ to G **do**

for $i = 1$ to K **do**

SR_i : calcule $M(Q_i, d)$ et $f(Q_i, d)$;

SR_i : envoie le tuple $(Q_i, d_i, M(Q_i, d), f(Q_i, d))$ à l'AC ;

end for

L'AC enregistre l'ensemble $M(Q^*, d)$ du meilleur clone Q^* ;

L'AC applique des opérateurs génétiques sur les clones Q_i pour générer des nouveaux clones ;

for $i = 1; i \leq K$ **do**

l'AC envoie Q_i à l'agent SR_i ;

end for

end for

Output : $M(Q^*, d), f(Q^*, d)$. $M(Q^*, d)$ est le résultat de rechercher pour la requête Q .

7.4.2/ LA STRUCTURE DU MAS-SR-GA

En effet, la structure de MAS peut être définie par plusieurs topologies. Dans la littérature, différentes topologies sont présentées. Nous pouvons par exemple faire référence à la topologie en anneau [Cantú-Paz, 1998] et à la topologie aléatoire [Tang et al., 2004]. Dans le premier modèle, les agents sont reliés entre eux pour former un cercle fermé. La topologie du second modèle est définie de manière aléatoire, de sorte que la propagation de l'information change selon l'organisation des agents. Dans notre modèle, nous utilisons une topologie simplifiée, dans laquelle les agents SR_i sont indépendants les uns des autres et sont en interaction avec un agent coordonnateur AC.

L'AC commence par la création des individus Q_i de la requête Q . Ensuite, il envoie (Q_i, d) aux agents SR_i . Chaque agent effectue un processus de recherche avec la paire (Q_i, d) , calcule $f(Q_i, d)$, et trouve $M(Q_i, d)$. L'AC reçoit tous les tuples $(Q_i, d, M(Q_i, d), f(Q_i, d))$ envoyés par tous les agents SR_i . Au cours des générations, l'AC compare tous les $f(Q_i, d)$, sélectionne le meilleur Q^* , et enregistre son $M(Q^*, d)$ correspondant et sa fonction d'appétitude $f(Q_i, d)$. Pour diversifier la population, et créer une nouvelle génération P_{g+1} de clones à partir de la génération actuelle P_g , l'AC applique les opérateurs de croisement et de mutation. Ce processus est répété pour G itérations. Finalement, le meilleur $M(Q^*, d)$ dans toutes les générations donne les résultats de recherche pour la requête Q (voir tableaux 7.2 et 7.4). La structure de notre MAS-SR-GA est illustrée dans la Figure 7.10.

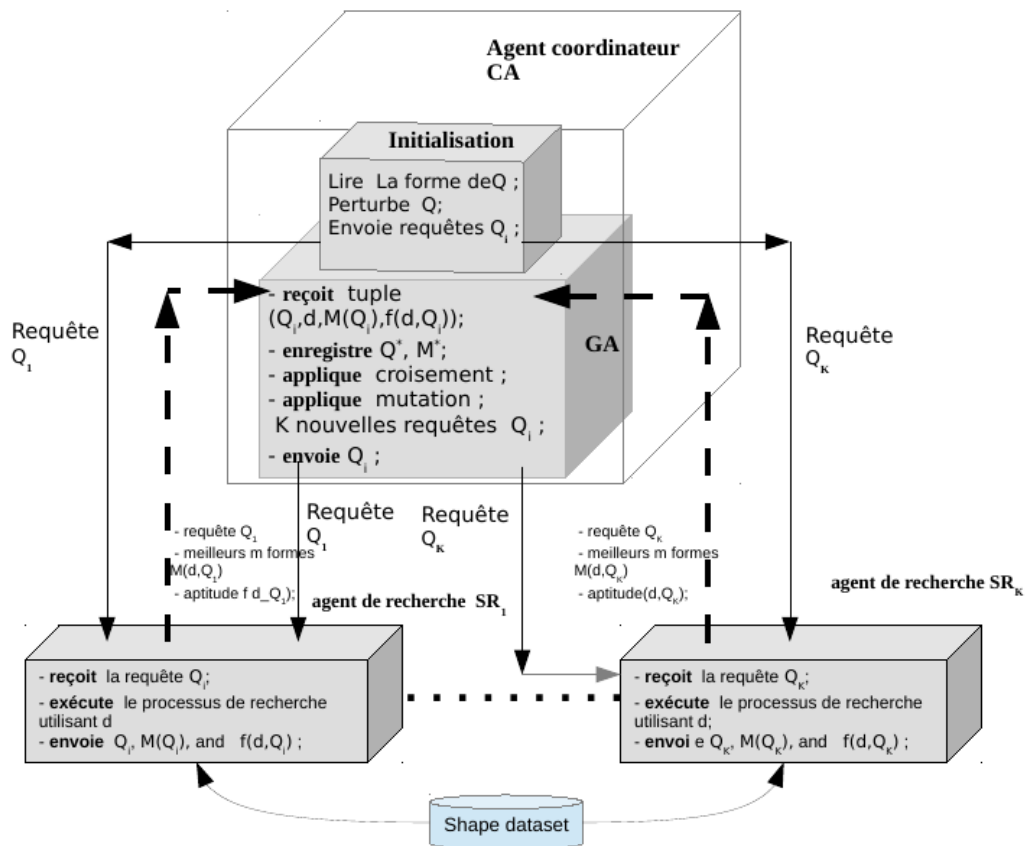


FIGURE 7.10 – L'architecture de notre MAS-SR-GA.

7.5/ RÉSULTATS EXPÉRIMENTAUX

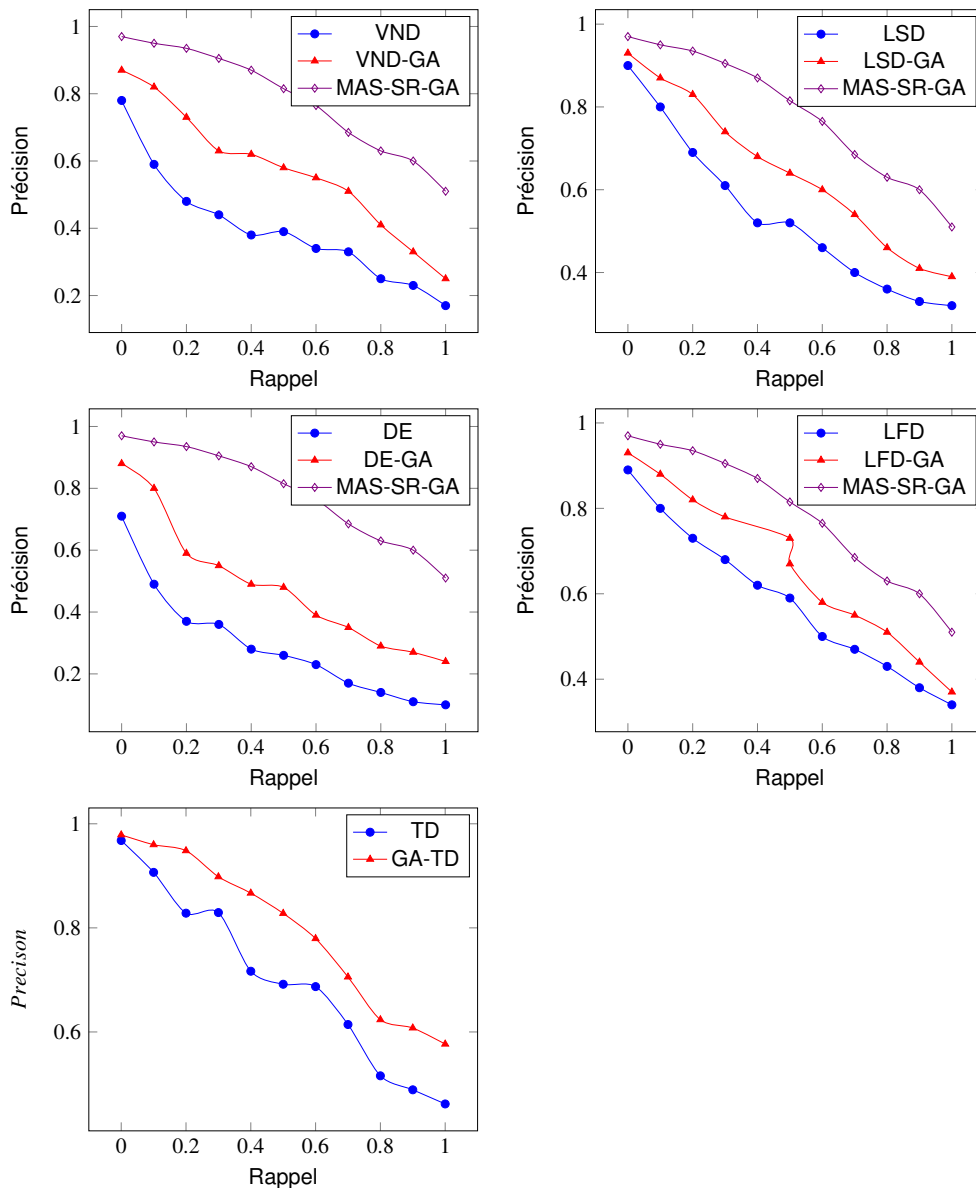


FIGURE 7.11 – Les courbes précision-rappel comparant les différents descripteurs avec le GA appliqué à ces derniers. La base de données est SHREC'11.

7.5.1/ RÉSULTATS

Habituellement, le processus de recherche renvoie les m formes les plus proches de la requête Q . Pour le cas des descripteurs améliorés par notre algorithme génétique, les formes sont les m formes dans la base de données les plus proches du meilleur clone Q^* , c-à-d le clone avec la meilleure valeur d'aptitude.

La figure 7.11 montre la performance des descripteurs VND, LSD, TD, DE, et LFD, par rapport à leurs homologues en appliquant notre algorithme génétique (notés *descripteur-GA*). Nous pouvons clairement constater que l'algorithme génétique améliore de manière

significative les performances de tous les descripteurs. A partir de cette figure, nous pouvons également constater que l'algorithme génétique distribué (MAS-SR-GA) améliore la précision de tous les descripteurs utilisés simultanément.

7.5.2/ COMPARAISON DE L'ALGORITHME GÉNÉTIQUE ET DU LISSAGE

Pour mesurer l'efficacité de la technique de clonage proposée, nous comparons les résultats obtenus avec notre algorithme à ceux obtenus en utilisant le lissage.

Pour lisser un maillage donné, chaque sommet dans le maillage est remplacé par de nouvelles coordonnées. Elles sont données par l'équation 7.6.

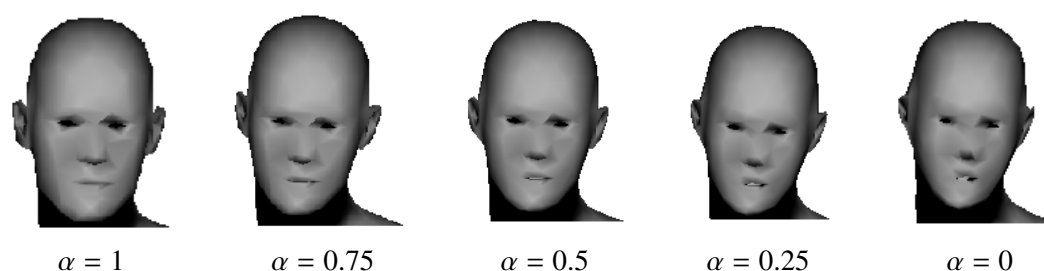


FIGURE 7.12 – Un exemple d'un modèle lissé considérant différentes valeurs de α .

$$v' = \alpha v + (1 - \alpha)g \quad (7.6)$$

où g est le centre de gravité des voisins du sommet v , et α est un paramètre donné. La distance lissée entre deux formes A et B est $d' = d(s(A), s(B))$, où $s(\cdot)$ est l'opérateur de lissage. La Figure 7.12 montre un exemple d'un modèle lissé, en considérant différentes valeurs de α .

Les graphiques de la figure 7.13 montre les courbes de distances moyennes entre des requêtes, et le reste des objets dans la base, en utilisant différents descripteurs de formes, et selon les valeurs du paramètre α . La meilleure valeur de α est celle qui permet d'obtenir une distance minimale. Chaque descripteur a une valeur appropriée de α .

La Figure 7.14 montre les courbes rappel/précision correspondant aux différents descripteurs appliqués à des surfaces lisses comparés aux descripteurs améliorés par notre algorithme génétique. Cette figure montre que l'algorithme génétique fonctionne mieux que les descripteurs appliqués à des surfaces lisses.

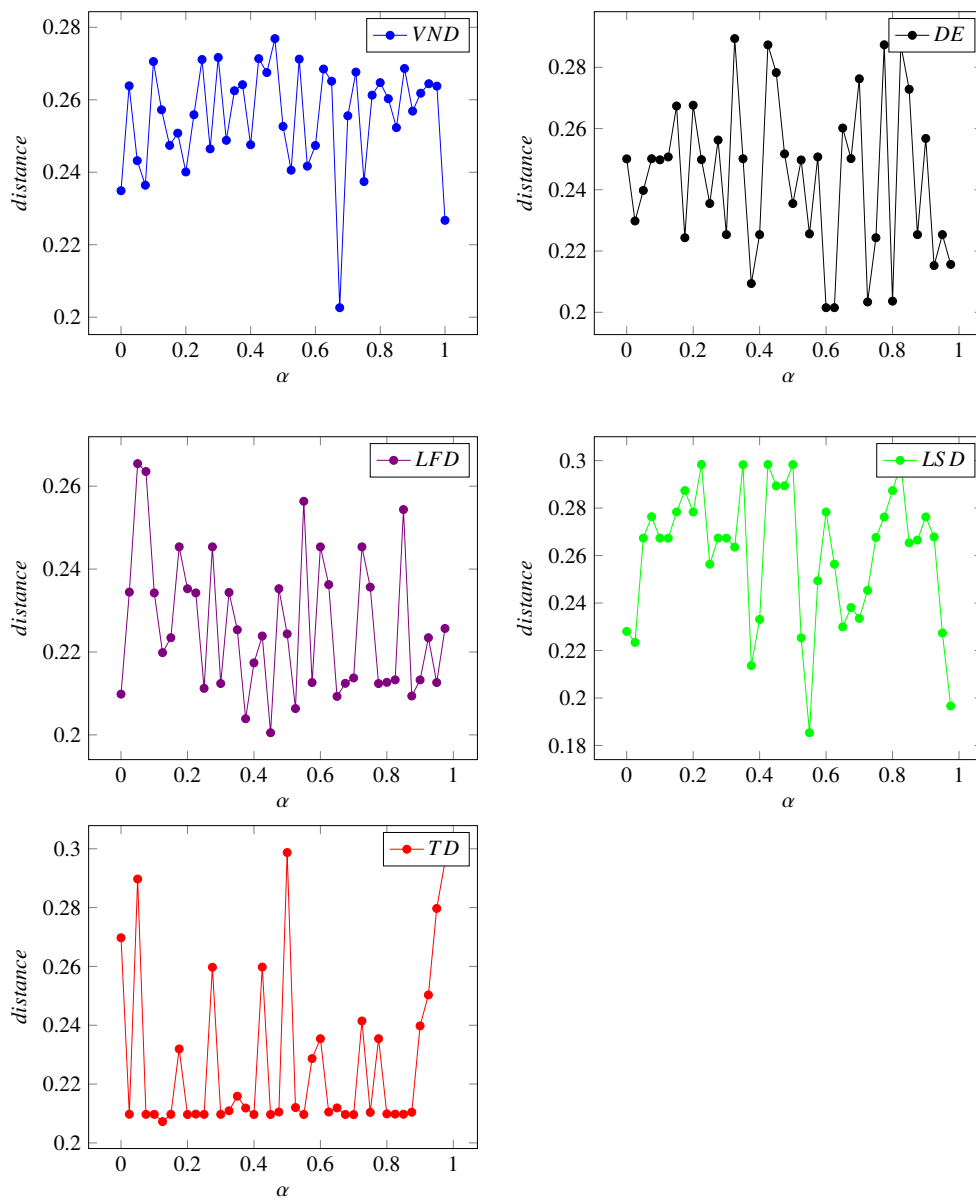


FIGURE 7.13 – Les distances moyennes entre des requêtes et les objets de la base de données TOSCA, en fonction du paramètre α utilisé pour le lissage.

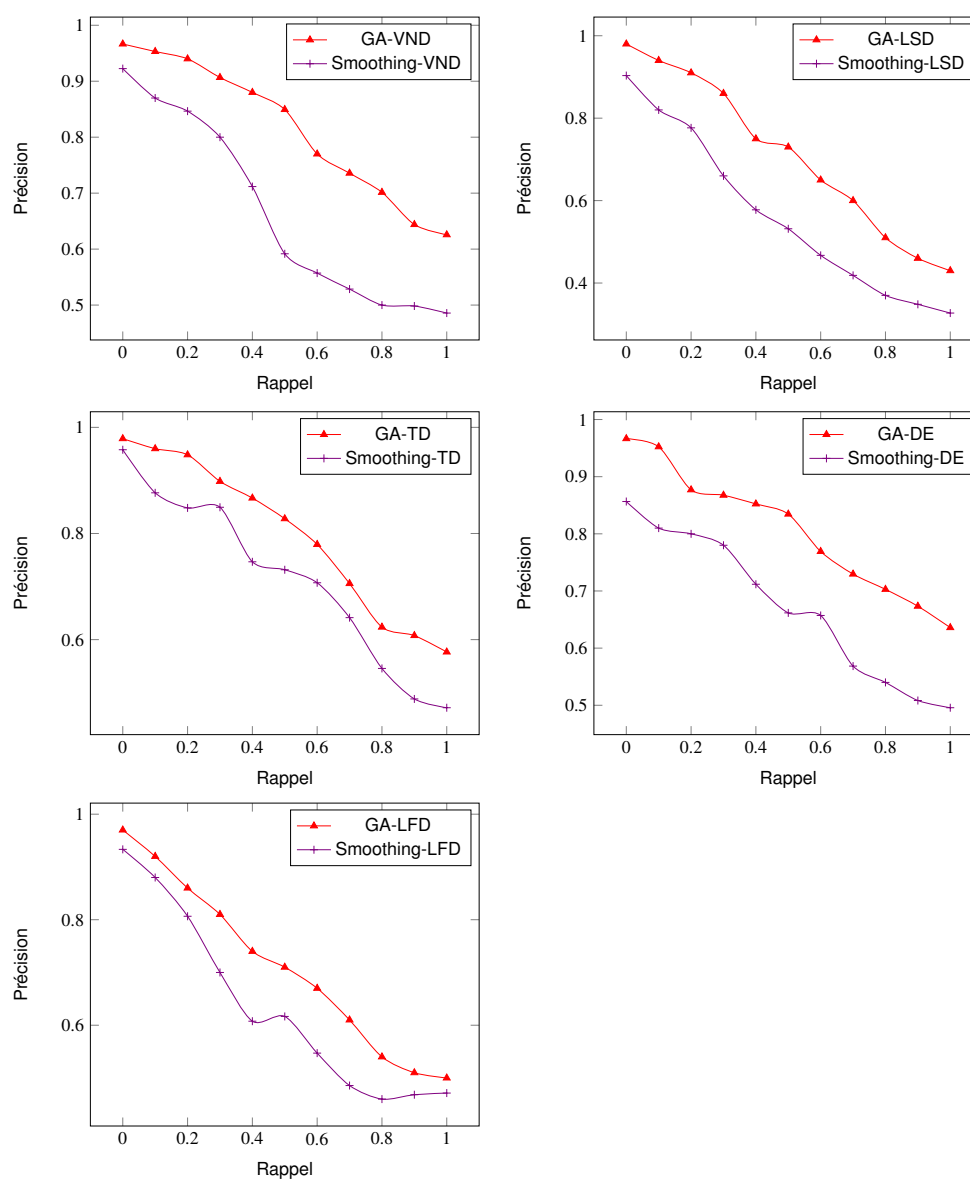


FIGURE 7.14 – Les courbes précision-rappel comparant les résultats de notre algorithme avec ceux du lissage.

7.5.3/ COMPARAISON DES RÉSULTATS

Dans cette section, la performance de notre méthode est comparée aux différents descripteurs présentés dans la littérature.

- **D2** : ce descripteur [Osada et al., 2001] est basé sur la distribution des distances entre les sommets sur la surface d'un modèle 3D.
- **GD** : la diffusion géométrique (GD) est un descripteur basée sur la distribution des distances de diffusion entre les sommets dans le graphe du maillage triangulaire d'un objet 3D. La distance de diffusion [Coifman et al., 2005] représente la longueur moyenne du trajet d'une marche aléatoire à travers les sommets du graphe.

- **AVG** : ce descripteur est basé sur la longueur moyenne des trajectoires géodésiques entre les sommets du maillage [Gal et al., 2007].
- **HSD-MGS** : c'est l'une des méthodes de fusion les plus efficaces, utilisées pour améliorer la précision du processus de recherche [Li et al., 2014].
- **GD-GT** : c'est la méthode d'amélioration de la recherche par forme clef présentée dans [Yang et al., 2008], appliquée avec le descripteur GD.

Les résultats des comparaisons sont illustrés dans la Figure 7.15. Notre méthode est meilleure.

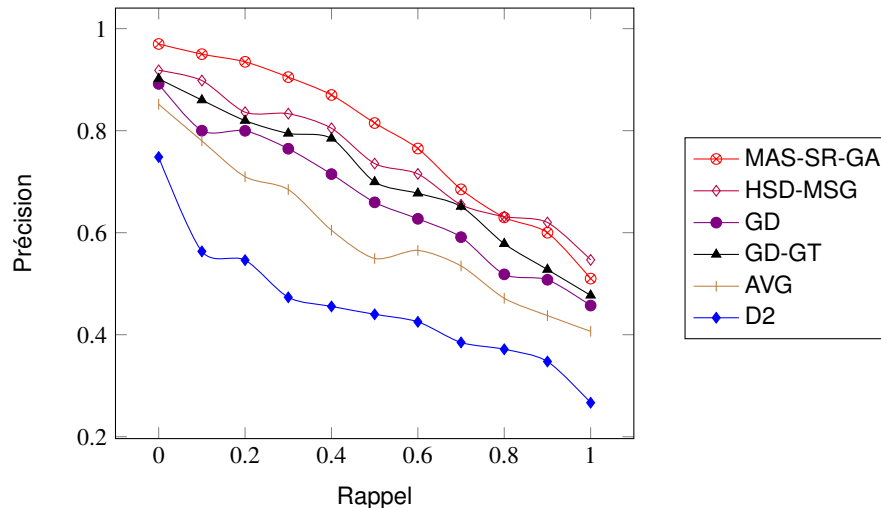


FIGURE 7.15 – La comparaison des résultats de notre méthode avec d'autres méthodes proposées dans la littérature. Les courbes sont tracées en fonction des résultats rapportés sur des objets de la base de données TOSCA.

7.5.4/ PRÉ-TRAITEMENT ET TEMPS DE CALCUL

Le tableau 7.6 donne le temps de calcul des descripteurs utilisés par notre méthode. Dans la pratique, les descripteurs de formes des objets de la base de données sont calculés en même temps que ceux de la requête lors du processus de recherche : pour mesurer la dissimilarité entre la requête et chaque forme dans la base de données, il faut calculer les descripteurs de forme pour chacune d'entre elles.

Afin d'accélérer le processus de recherche, le descripteur de chaque forme dans la base de données est calculé la première fois que c'est nécessaire, et stocké dans un fichier. Lors du calcul de $d(Q_i, O)$, nous cherchons le fichier correspondant au descripteur de O ; s'il existe, alors il est utilisé, sinon le descripteur de la forme O est calculé, puis stocké dans un nouveau fichier.

Dans la structure distribuée de notre algorithme génétique, le processus de recherche prend moins de temps grâce au parallélisme. Tous les calculs des $d(Q_i, O)$ ont besoin de moins d'une seconde (voir le tableau 7.6) lorsque le descripteur de la forme O est connu. Quand il n'est pas connu, le calcul des deux descripteurs peut également être fait en parallèle, et donc moins d'une seconde est suffisante dans ce cas également. Dans une architecture totalement distribuée, où tous les calculs $d(Q_i, O)$ sont effectués en parallèle, chaque itération de l'AG a besoin d'une seconde ou moins. Si G est le nombre d'itéra-

TABLE 7.6 – Temps des calculs.

Temps (en secondes)			
Processus de recherche pour Q_i			
Descripteur	$d(Q_i, O)$	Real time	Pre-processing
LSD	0.8998	539.886	1.0258
DE	0.7359	441.51	0.8619
VND	0.6327	379.62	0.7595
LFD	0.9215	552.93	1.0465
TD	1.254	909.7	1.145
Création Q_i		0.0322	
N		600	

tions de l'AG, et T le temps maximal nécessaire pour le calcul d'une distance $d(Q_i, O)$ ($T = 1$ seconde dans notre implémentation actuelle), alors la durée de notre algorithme de recherche est de GT . Il est indépendant du nombre de formes dans la base de données. Le facteur G est inévitable : pour calculer la génération $g + 1$, nous avons besoin de connaître la génération g . Si N est le nombre de formes dans la base de données, nous calculons Nd descripteurs de formes dans la base de données, GKd descripteurs (histogrammes) correspondants aux requêtes, et $NGKd$ distances entre une requête Q_i et une forme dans la base de données. Le total est $(N + 1)GKd$ calculs, chacun a besoin de T temps ou moins. Lorsqu'il n'y a pas suffisamment de processeurs (pour calculer en temps GT), il est raisonnable de penser que la complexité de notre méthode est proche de $(N + 1)GKd/P$ où P est le nombre de processeurs disponibles.

Notez que les expériences ont été effectuées sur un ordinateur portable (Intel (R) Core (TM) i5 CPU M 560 2.67GHz, 6 Go de RAM), en utilisant C++ sous Arch Linux GNU / Linux 3.16.

7.6/ CONCLUSION

Les artefacts invisibles, le bruit ou l'irrégularité dans les maillages 3D réduit la performance des méthodes de recherche par forme clef. En règle générale, les approches classiques d'amélioration de descripteurs de forme utilisent des techniques de fusion de plusieurs descripteurs de formes.

Au lieu de cela, la méthode proposée dans ce chapitre améliore la requête plutôt que le descripteur de formes. L'algorithme génétique effectue des perturbations invisibles sur la surface de la requête. Il donne des clones qui s'approchent de plus en plus des formes dans la base de données, quel que soit le ou les descripteurs de formes utilisés. Les classes des objets dans la base de données n'ont pas besoin d'être connues par notre algorithme.

Les résultats expérimentaux ont montré que cette amélioration de la requête améliore également la précision de la recherche. Ils montrent que l'algorithme génétique permet d'obtenir de meilleurs résultats que les méthodes pré-existantes. De plus, il est meilleur que le lissage.

L'algorithme génétique est lent, par exemple beaucoup plus lent que le lissage, mais il est massivement parallèle. Nous avons proposé une architecture parallèle de notre algo-

rithme en utilisant un système multi agents. Cette architecture parallèle permet d'utiliser plusieurs descripteurs, sans augmenter les temps d'exécution.

A la suite de ce chapitre, nous pouvons conclure que les descripteurs de formes sont numériquement instables, ou autrement dit, sensibles aux détails invisibles à l'oeil humain, même pour les plus robustes d'entre eux. Cette sensibilité pose un problème (elle rend plus difficile le processus de recherche) et en même temps, elle fournit une solution : le clonage.

CONCLUSION

Dans cette thèse de doctorat, nous avons développé plusieurs contributions au domaine de l'appariement de formes et la recherche par forme clef.

Dans un premier temps, dans le premier chapitre, nous avons introduit les notions de base de l'appariement et de la recherche par forme clef. Le chapitre donne un état de l'art des méthodes utilisées pour la représentation et l'appariement de formes, selon leurs différentes catégories.

Dans le deuxième chapitre, nous avons présenté les différents concepts et outils utilisés dans les chapitres restants. Nous avons donc introduit les différentes représentations d'objets, les bases de données utilisées, ainsi que les mesures d'évaluation du processus de recherche par forme clef.

Nous avons par la suite présenté dans le troisième chapitre une nouvelle méthode pour la décomposition d'un objet, représenté par un nuage de points (pixels ou voxels), en squelette d'ellipsoïdes. La méthode calcule la meilleure décomposition en différentes parties ou classes ou ellipsoïdes. La similarité entre deux formes A et B, décomposées chacune en k ellipsoïdes est équivalente à un couplage optimal dans un graphe biparti avec $2k$ sommets. La méthode est facile à mettre en oeuvre, résistante au bruit et aux aléas d'échantillonnage.

Afin d'introduire les contributions restantes de cette thèse, nous avons décrit les différents descripteurs de formes utilisés par ces contributions, ainsi que la mesure de dissimilarité utilisée pour définir la similarité des objets.

Nous avons par la suite proposé une nouvelle méthode basée sur la fusion de deux descripteurs de formes, l'un local, l'autre global. La méthode fusionne des caractéristiques calculées en chaque sommet du maillage décrivant l'objet ; des caractéristiques sont extraites en utilisant un premier descripteur local basé sur les courbures, et un descripteur global basé sur la géométrie de la diffusion. Différentes règles de fusion peuvent être utilisées pour fusionner les deux descripteurs. Les résultats expérimentaux ont montré que la règle du produit est la plus simple, et la plus efficace.

Une autre méthode de fusion a été ensuite présentée. Cette nouvelle méthode est basée sur la fusion des mesures de dissimilarité plutôt que sur les descripteurs. Elle considère le graphe complet, dont les sommets sont les formes de la base ou la requête, et les arêtes sont étiquetées par plusieurs distances, une par descripteur ; ensuite cette méthode calcule par programmation linéaire la combinaison convexe des distances qui maximise soit la somme des longueurs des plus courts chemins entre la requête et les objets de la base de données, soit la longueur du plus court chemin entre la requête et l'objet comparé à la requête. Cette méthode peut être généralisée pour être utilisée avec d'autres descripteurs de formes.

Dans le dernier chapitre, nous avons présenté la dernière contribution de cette thèse ; elle décrit une nouvelle façon pour fusionner et améliorer les descripteurs de formes.

Cette méthode améliore les formes plutôt que les descripteurs : elle consiste à perturber la requête avec un algorithme génétique pour la rapprocher des formes de la base de données, pour un descripteur donné, ou plusieurs ; cette méthode est massivement parallèle, et une architecture multi-agent est proposée. L'algorithme génétique fait évoluer plusieurs générations de clones de la requête. Les clones sont visuellement identiques à la requête. Le meilleur clone est celui qui optimise une fonction distance (aptitude) ; nous considérons plusieurs distances. Nous comparons les différents résultats obtenus en utilisant les différentes fonctions d'aptitude.

Une leçon du chapitre sur le clonage est l'instabilité des descripteurs de formes : ils sont sensibles à des détails invisibles à l'oeil humain, même pour les plus robustes d'entre eux. Cette sensibilité pose un problème (elle rend plus difficile le processus de recherche) et en même temps, elle fournit une solution : le clonage.

Concluons sur les travaux futurs, en fait des travaux que nous n'avons pas pu mener à bien faute de temps :

- Comparer et éventuellement combiner les trois méthodes proposées dans cette thèse, aux chapitres 5, 6, 7 ;
- Perturber le Laplacien plutôt que les coordonnées cartésiennes, dans la méthode de clonage, au chapitre 7 ;
- Améliorer la complexité de l'algorithme de combinaison linéaire convexe des distances du chapitre 6.

Nous avons envisagé plusieurs méthodes que nous n'avons pas eu le temps de tester.

Mentionnons en une. L'idée est de fusionner plusieurs chaînes de Markov sur le même graphe, chaque chaîne utilisant sa propre matrice de coûts. L'idée est de calculer la combinaison linéaire convexe des chaînes (il s'agit bien d'une chaîne de Markov) qui a la plus grande vitesse de mélange, autrement dit celle qui a la valeur propre λ_2 la plus petite en valeur absolue. Rappelons que la vitesse de mélange, autrement dit la vitesse de convergence vers la distribution stationnaire, est déterminée par λ_2 , la deuxième valeur propre de la matrice de transition de la chaîne de Markov, quand elles sont ordonnées par module décroissant : $\lambda_1 = 1 \geq |\lambda_2| \geq \dots$; plus $|\lambda_2|$ est petite et plus cette convergence, ou mélange, est rapide. Cette méthode est donc une méthode inédite de fusion précoce. On peut envisager que cette méthode puisse aussi être utilisée pour le contrôle optimal de processus stochastiques, chacun d'eux étant représenté par une chaîne de Markov (toujours sur le même graphe).

Mentionnons en une seconde. Notre idée était d'utiliser des chaînes de Markov absorbantes. A chaque état de la chaîne, est associé un état terminal : un seul arc, réflexif et de probabilité un, est issu de cet état terminal. Par exemple, ce type de chaînes peut modéliser le trafic automobile sur un réseau routier, et les états terminaux permettent de comptabiliser les accidents mortels. Ces chaînes de Markov peuvent aussi être utilisées pour des calculs d'éclaircissements diffus par les méthodes "de radiosité" (*radiosity*) : un photon est émis par une source lumineuse, rebondit sur des surfels (éléments de surface) représentés par les états de la chaîne ; quand il est absorbé par un surfel (ou vu par un spectateur virtuel), il termine dans l'état terminal associé. Le suivi de quelques milliers ou millions de photons virtuels permet de calculer la radiosité de chaque surfel : elle est proportionnelle au nombre de photons piégés dans l'état terminal du surfel. L'idée est donc d'utiliser l'accidentologie, ou pour être moins sinistre la géométrie de la propagation de la lumière plutôt que la géométrie de la diffusion de la chaleur.

Finalement, nous avons utilisé deux modèles différents, mais intuitivement équivalents, pour la géométrie de diffusion : d'une part, le Laplacien d'un maillage triangulaire, et d'autre part les chaînes de Markov. Les deux modèles font appel à l'algèbre linéaire, en particulier à la décomposition de matrices (le Laplacien, ou la matrice de Markov) en valeurs et vecteurs propres. Faute de temps, nous n'avons pas comparé ces deux modèles, et nous n'avons pas étudié leur éventuelle connexion, comme le lien entre les valeurs et vecteurs propres de la Laplacienne du graphe d'un maillage, et les valeurs et vecteurs propres des matrices de Markov associées à ce même graphe. Il semble y avoir peu de littérature sur cette question. Remarquons pour conclure que la modélisation géométrique [Stanko et al., 2016] utilise les dérivées seconde et troisième du Laplacien (par exemple pour contraindre, concevoir ou reconstruire des maillages), mais les dérivées du Laplacien discret ne sont pas encore utilisées dans le domaine de la recherche par forme clef.

BIBLIOGRAPHIE

- [Abdullah et al., 2009] Abdullah, A., Veltkamp, R., et Wiering, M. (2009). **An ensemble of deep support vector machines for image categorization**. Dans *Soft Computing and Pattern Recognition, 2009. SOCPAR '09. International Conference of*, pages 301–306.
- [Ahamed Shafeeq, 2012] Ahamed Shafeeq, K. H. (2012). **Dynamic clustering of data with modified k-means algorithm**. Dans *International Proceedings of Computer Science & Information Tech*, volume 27, page 221.
- [Akgül et al., 2008] Akgül, C. B., Sankur, B., Yemez, Y., et Schmitt, F. (2008). **Similarity score fusion by ranking risk minimization for 3d object retrieval**. Dans *Proceedings of the 1st Eurographics conference on 3D Object Retrieval*, pages 41–48. Eurographics Association.
- [Aparna, 2013] Aparna, K. (2013). **Retrieval of digital images based on multi-feature similarity using genetic algorithm**. *International Journal of Engineering Research and Applications (IJERA)*, 3(4) :1486–1499.
- [Aubry et al., 2011] Aubry, M., Schlickewei, U., et Cremers, D. (2011). **The wave kernel signature : A quantum mechanical approach to shape analysis**. Dans *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE.
- [Banégas et al., 2001] Banégas, F., Jaeger, M., Michelucci, D., et Roelens, M. (2001). **The ellipsoidal skeleton in medical applications**. Dans *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, SMA '01*, pages 30–38, New York, NY, USA. ACM.
- [Barber et al., 1996] Barber, C. B., Dobkin, D. P., et Huhdanpaa, H. (1996). **The quickhull algorithm for convex hulls**. *ACM Transactions on Mathematical Software (TOMS)*, 22(4) :469–483.
- [Basri et al., 1998] Basri, R., Costa, L., Geiger, D., et Jacobs, D. (1998). **Determining the similarity of deformable shapes**. *Vision Research*, 38(15) :2365–2385.
- [Belongie et al., 2000] Belongie, S., Malik, J., et Puzicha, J. (2000). **Shape context : A new descriptor for shape matching and object recognition**. Dans *NIPS*, volume 2, page 3.
- [Berg et al., 2005] Berg, A. C., Berg, T. L., et Malik, J. (2005). **Shape matching and object recognition using low distortion correspondences**. Dans *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 26–33. IEEE.
- [Bernhard et al., 2008] Bernhard, H., Korte, B., et Vygen, J. (2008). **Combinatorial optimization : Theory and algorithms**. Heidelberg : Springer-Verlag.
- [Bernier et al., 2003] Bernier, T., et Landry, J.-A. (2003). **A new method for representing and matching shapes of natural objects**. *Pattern Recognition*, 36(8) :1711–1723.

- [Bhanu et al., 1995] Bhanu, B., Lee, S., et Ming, J. (1995). **Adaptive image segmentation using a genetic algorithm**. *Systems, Man and Cybernetics, IEEE Transactions on*, 25(12) :1543–1567.
- [Bhanu et al., 1991] Bhanu, B., Lee, S., et Ming, J. C. (1991). **Self-optimizing image segmentation system using a genetic algorithm**. Dans *ICGA*, pages 362–369.
- [Blum et al., 2012] Blum, M., Springenberg, J. T., Wülfing, J., et Riedmiller, M. (2012). **A learned feature descriptor for object recognition in rgb-d data**. Dans *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1298–1303. IEEE.
- [Boumzaid et al., 2009] Boumzaid, Y., Lanquetin, S., et Neveu, M. (2009). **A reverse scheme for quadrilateral meshes**. Dans *GraphiCon*, volume 9, pages 25–26.
- [Bracewell, 1965] Bracewell, R. (1965). **The fourier transform and its applications**. *New York*, 5.
- [Brennecke et al., 2004] Brennecke, A., et Isenberg, T. (2004). **3d shape matching using skeleton graphs**. Dans *SimVis*, pages 299–310. Citeseer.
- [Bronstein et al., 2011] Bronstein, A. M., Bronstein, M. M., Guibas, L. J., et Ovsjanikov, M. (2011). **Shape google : Geometric words and expressions for invariant shape retrieval**. *ACM Transactions on Graphics (TOG)*, 30(1) :1.
- [Bronstein et al., 2010] Bronstein, M. M., et Kokkinos, I. (2010). **Scale-invariant heat kernel signatures for non-rigid shape recognition**. Dans *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1704–1711. IEEE.
- [Buch et al., 2016] Buch, A. G., Petersen, H. G., et Krüger, N. (2016). **Local shape feature fusion for improved matching, pose estimation and 3d object recognition**. *SpringerPlus*, 5(1) :1.
- [Burmeister et al., 1997] Burmeister, B., Haddadi, A., et Matylis, G. (1997). **Application of multi-agent systems in traffic and transportation**. Dans *Software Engineering. IEE Proceedings-[see also Software, IEE Proceedings]*, volume 144, pages 51–60. IET.
- [Cantú-Paz, 1998] Cantú-Paz, E. (1998). **A survey of parallel genetic algorithms**. *Calculateurs paralleles, reseaux et systems repartis*, 10(2) :141–171.
- [Chahooki et al., 2012] Chahooki, M., et Charkari, N. (2012). **Shape retrieval based on manifold learning by fusion of dissimilarity measures**. *Image Processing, IET*, 6(4) :327–336.
- [Chan et al., 1999] Chan, D. Y.-M., et King, I. (1999). **Genetic algorithm for weights assignment in dissimilarity function for trademark retrieval**. Dans *Visual Information and Information Systems*, pages 557–565. Springer.
- [Chang et al., 2011] Chang, M.-C., et Kimia, B. B. (2011). **Measuring 3d shape similarity by graph-based matching of the medial scaffolds**. *Computer Vision and Image Understanding*, 115(5) :707 – 720. Special issue on 3D Imaging and Modelling.
- [Chao et al., 2011] Chao, M.-W., Lin, C.-H., Chang, C.-C., et Lee, T.-Y. (2011). **A graph-based shape matching scheme for 3d articulated objects**. *Computer Animation and Virtual Worlds*, 22(2-3) :295–305.
- [Chapeau-Blondeau et al., 2009] Chapeau-Blondeau, F., et Rousseau, D. (2009). **Raising the noise to improve performance in optimal processing**. *Journal of Statistical Mechanics : Theory and Experiment*, 2009(01) :P01003.

- [Chaudhari et al., 2014] Chaudhari, A. J., Leahy, R. M., Wise, B. L., Lane, N. E., Badawi, R. D., et Joshi, A. A. (2014). **Global point signature for shape analysis of carpal bones**. *Physics in medicine and biology*, 59(4) :961.
- [Chazal et al., 2013] Chazal, F., Guibas, L. J., Oudot, S. Y., et Skraba, P. (2013). **Persistence-based clustering in riemannian manifolds**. *Journal of the ACM (JACM)*, 60(6) :41.
- [Chen et al., 2010] Chen, B., et Cheng, H. H. (2010). **A review of the applications of agent technology in traffic and transportation systems**. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2) :485–497.
- [Chen et al., 2002] Chen, D.-Y., et Ouhyoung, M. (2002). **A 3d object retrieval system based on multi-resolution reeb graph**. *Computers and Graphics*.
- [Chen et al., 2003] Chen, D.-Y., Tian, X.-P., Shen, Y.-T., et Ouhyoung, M. (2003). **On visual similarity based 3d model retrieval**. Dans *Computer graphics forum*, volume 22, pages 223–232. Wiley Online Library.
- [Chen et al., 2014] Chen, H., Varshney, L. R., et Varshney, P. K. (2014). **Noise-enhanced information systems**. *Proceedings of the IEEE*, 102(10) :1607–1621.
- [Cheng et al., 2013] Cheng, Z.-Q., Chen, Y., Martin, R., Lai, Y.-K., et Wang, A. (2013). **Supermatching : Feature matching using supersymmetric geometric constraints**. *Visualization and Computer Graphics, IEEE Transactions on*, 19(11) :1885–1894.
- [Coifman et al., 2005] Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F., et Zucker, S. W. (2005). **Geometric diffusions as a tool for harmonic analysis and structure definition of data : Diffusion maps**. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21) :7426–7431.
- [Cormen, 2009] Cormen, T. H. (2009). **Introduction to algorithms**. MIT press.
- [Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., et Stein, C. (2009). **Introduction to Algorithms, Third Edition**. The MIT Press, 3rd édition.
- [Dao et al., 2006a] Dao, M. N., et De Amicis, R. (2006a). **A new method for boundary-based shape matching and retrieval**. Dans *Image Processing, 2006 IEEE International Conference on*, pages 1485–1488. IEEE.
- [Dao et al., 2006b] Dao, M.-S., et de Amicis, R. (2006b). **A new method for boundary-based shape matching and retrieval**. Dans *Image Processing, 2006 IEEE International Conference on*, pages 1485–1488.
- [De Jong, 1988] De Jong, K. (1988). **Learning with genetic algorithms : An overview**. *Machine Learning*, 3(2-3) :121–138.
- [Dey et al., 2010] Dey, T. K., Li, K., Luo, C., Ranjan, P., Safa, I., et Wang, Y. (2010). **Persistent heat signature for pose-oblivious matching of incomplete models**. Dans *Computer Graphics Forum*, volume 29, pages 1545–1554. Wiley Online Library.
- [Dickinson et al., 2014] Dickinson, S. J., Shokoufandeh, A., et Siddiqi, K. (2014). **Shock graph**. Dans *Computer Vision*, pages 729–737. Springer.
- [Diday, 1971] Diday, E. (1971). **Une nouvelle méthode en classification automatique et reconnaissance des formes, la méthode des nuées dynamiques**. *Revue de Statistique Appliquée*, 19(2) :19–33.
- [Dixon, 1993] Dixon, M. (1993). **Gift wrapping**. US Patent 5,186,988.
- [Do Carmo et al., 1976] Do Carmo, M. P., et Do Carmo, M. P. (1976). **Differential geometry of curves and surfaces**, volume 2. Prentice-hall Englewood Cliffs.

- [Donald, 1999] Donald, E. K. (1999). **The art of computer programming**. *Sorting and searching*, 3 :426–458.
- [Douros et al., 2002] Douros, I., et Buxton, B. F. (2002). **Three-dimensional surface curvature estimation using quadric surface patches**. *Scanning*.
- [Dubrovina et al., 2010] Dubrovina, A., et Kimmel, R. (2010). **Matching shapes by eigendecomposition of the laplace-beltrami operator**. Dans *Proc. 3DPVT*, volume 2.
- [El-Mehalawi et al., 2003] El-Mehalawi, M., et Miller, R. A. (2003). **A database system of mechanical components based on geometric and topological similarity. part ii : indexing, retrieval, matching, and similarity assessment**. *Computer-Aided Design*, 35(1) :95–105.
- [Fan et al., 2004] Fan, W., Gordon, M. D., et Pathak, P. (2004). **A generic ranking function discovery framework by genetic programming for information retrieval**. *Information Processing & Management*, 40(4) :587–602.
- [Fang et al., 2011] Fang, Y., Sun, M., et K., R. (2011). **Temperature distribution descriptor for robust 3D shape retrieval**. Dans *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 9–16.
- [Farin, 2002] Farin, G. E. (2002). **Curves and surfaces for CAGD : a practical guide**. Morgan Kaufmann.
- [Ferber, 1999] Ferber, J. (1999). **Multi-agent systems : an introduction to distributed artificial intelligence**, volume 1. Addison-Wesley Reading.
- [Florea, 1998] Florea, A. M. (1998). **Introduction to multi-agent systems**. *International Summer School on Multi-Agent Systems, Bucharest*, pages 1–11.
- [Folkers et al., 2002] Folkers, A., et Samet, H. (2002). **Content-based image retrieval using fourier descriptors on a logo database**. Dans *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 521–524. IEEE.
- [Funayama et al., 2012] Funayama, R., Yanagihara, H., Van Gool, L., Tuytelaars, T., et Bay, H. (2012). **Robust interest point detector and descriptor**. US Patent 8,165,401.
- [Funkhouser et al., 2006] Funkhouser, T., et Shilane, P. (2006). **Partial matching of 3 d shapes with priority-driven search**. Dans *ACM International conference proceeding series*, volume 256, pages 131–142. Citeseer.
- [Gal et al., 2006] Gal, R., et Cohen-Or, D. (2006). **Salient geometric features for partial shape matching and similarity**. *ACM Transactions on Graphics (TOG)*, 25(1) :130–150.
- [Gal et al., 2007] Gal, R., Shamir, A., et Cohen-Or, D. (2007). **Pose-oblivious shape signature**. *Visualization and Computer Graphics, IEEE Transactions on*, 13(2) :261–271.
- [Galvez, 2012] Galvez, J. (2012). **A graph-theoretical approach to calculate vibrational energies of atomic and subatomic systems**. *Open Journal of Physical Chemistry*, 2(4) :204–211.
- [Goldberg, 1989] Goldberg, D. E. (1989). **Genetic Algorithms in Search, Optimization and Machine Learning**. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st édition.
- [Goldberg et al., 1988] Goldberg, D. E., et Holland, J. H. (1988). **Genetic algorithms and machine learning**. *Machine learning*, 3(2) :95–99.

- [Grefenstette, 1988] Grefenstette, J. J. (1988). **Credit assignment in rule discovery systems based on genetic algorithms**. *Machine Learning*, 3(2-3) :225–245.
- [Grundmann et al., 2008] Grundmann, M., Meier, F., et Essa, I. (2008). **3d shape context and distance transform for action recognition**. Dans *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE.
- [Guellil et al., 2009] Guellil, Z., et Zaoui, L. (2009). **Proposition d’une solution au problème d’initialisation cas du k -means**. Dans *CIIA*.
- [Harp et al., 1989] Harp, S. A., Samad, T., et Guha, A. (1989). **Towards the genetic synthesis of neural network**. Dans *Proceedings of the Third International Conference on Genetic Algorithms*, pages 360–369, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Harris et al., 1988] Harris, C., et Stephens, M. (1988). **A combined corner and edge detector**. Dans *Alvey vision conference*, volume 15, page 50. Citeseer.
- [Heider et al., 2011] Heider, P., Pierre-Pierre, A., Li, R., et Grimm, C. (2011). **Local shape descriptors, a survey and evaluation**. Dans *Proceedings of the 4th Eurographics conference on 3D Object Retrieval*, pages 49–56. Eurographics Association.
- [Hilaga et al., 2001] Hilaga, M., Shinagawa, Y., Kohmura, T., et Kunii, T. L. (2001). **Topology matching for fully automatic similarity estimation of 3d shapes**. Dans *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212. ACM.
- [Holland, 1975] Holland, J. H. (1975). **Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence**. U Michigan Press.
- [Holland, 1992] Holland, J. H. (1992). **Adaptation in Natural and Artificial Systems**. MIT Press, Cambridge, MA, USA.
- [<http://www.cs.princeton.edu/min/binvox/>,] <http://www.cs.princeton.edu/min/binvox/>.
- [Hu et al., 2009] Hu, J., et Hua, J. (2009). **Salient spectral geometric features for shape matching and retrieval**. *Vis. Comput.*, 25(5-7) :667–675.
- [Hyndman, 1995] Hyndman, R. J. (1995). **The problem with sturges rule for constructing histograms**. *Monash University*.
- [Jain et al., 2007] Jain, V., et Zhang, H. (2007). **A spectral approach to shape-based retrieval of articulated 3d models**. *Computer-Aided Design*, 39(5) :398–407.
- [Jayanti et al., 2006] Jayanti, S., Kalyanaraman, Y., Iyer, N., et Ramani, K. (2006). **Developing an engineering shape benchmark for cad models**. *Computer-Aided Design*, 38(9) :939–953.
- [Jiang et al., 2009] Jiang, Y., Edmiston, E., Wang, F., Blumberg, H., Papademetris, X., et Staib, L. (2009). **Improving the reliability of shape comparison by perturbation**. Dans *Biomedical Imaging : From Nano to Macro, 2009. ISBI '09. IEEE International Symposium on*, pages 686–689.
- [Kalyanaraman et al., 2005] Kalyanaraman, Y., Boutin, M., et Ramani, K. (2005). **Effectiveness of convex hull histograms for shape matching**. Rapport technique, PRE-CISE technical report. PRE-TR-2005-2, West Lafayette (IN) : Purdue University.
- [Karbasi et al., 2014] Karbasi, A., Salavati, A. H., Shokrollahi, A., et Varshney, L. R. (2014). **Noise facilitation in associative memories of exponential capacity**. *Neural computation*.

- [Karp et al., 1987] Karp, R. M., et Rabin, M. O. (1987). **Efficient randomized pattern-matching algorithms**. *IBM Journal of Research and Development*, 31(2) :249–260.
- [Kazhdan et al., 2002] Kazhdan, M., Chazelle, B., Dobkin, D., Finkelstein, A., et Funkhouser, T. (2002). **A reflective symmetry descriptor**. Dans *Computer Vision ECCV 2002*, pages 642–656. Springer.
- [Kazhdan et al., 2003] Kazhdan, M., Funkhouser, T., et Rusinkiewicz, S. (2003). **Rotation invariant spherical harmonic representation of 3 d shape descriptors**. Dans *Symposium on geometry processing*, volume 6, pages 156–164.
- [Khotanzad et al., 1990] Khotanzad, A., et Hong, Y. H. (1990). **Invariant image recognition by zernike moments**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(5) :489–497.
- [Kim et al., 2000] Kim, H.-K., et Kim, J.-D. (2000). **Region-based shape descriptor invariant to rotation, scale and translation**. *Signal Processing : Image Communication*, 16(1) :87–93.
- [Kittler et al., 1998] Kittler, J., Hatef, M., Duin, R. P. W., et Matas, J. (1998). **On combining classifiers**. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3) :226–239.
- [Knopp et al., 2010] Knopp, J., Prasad, M., Willems, G., Timofte, R., et Van Gool, L. (2010). **Hough transform and 3d surf for robust three dimensional classification**. Dans *Computer Vision–ECCV 2010*, pages 589–602. Springer.
- [Koenderink et al., 1992] Koenderink, J. J., et van Doorn, A. J. (1992). **Surface shape and curvature scales**. *Image and vision computing*, 10(8) :557–564.
- [Kokkinos et al., 2012] Kokkinos, I., Bronstein, M. M., Litman, R., et Bronstein, A. M. (2012). **Intrinsic shape context descriptors for deformable shapes**. Dans *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 159–166. IEEE.
- [Körtgen et al., 2003] Körtgen, M., Park, G.-J., Novotni, M., et Klein, R. (2003). **3d shape matching with 3d shape contexts**. Dans *The 7th central European seminar on computer graphics*, volume 3, pages 5–17.
- [Kuhn, 1955] Kuhn, H. W. (1955). **The hungarian method for the assignment problem**. *Naval research logistics quarterly*, 2(1-2) :83–97.
- [Kuncheva, 2002] Kuncheva, L. I. (2002). **A theoretical study on six classifier fusion strategies**. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(2) :281–286.
- [Laga et al., 2011] Laga, H., Schreck, T., Ferreira, A., Godil, A., Pratikakis, I., et others (2011). **Local shape descriptors, a survey and evaluation**.
- [Lai et al., 2009] Lai, Y.-K., Hu, S.-M., Martin, R. R., et Rosin, P. L. (2009). **Rapid and effective segmentation of 3d models using random walks**. *Computer Aided Geometric Design*, 26(6) :665–679.
- [Latombe, 2012] Latombe, J.-C. (2012). **Robot motion planning**, volume 124. Springer Science & Business Media.
- [Lavoué, 2011] Lavoué, G. (2011). **Bag of words and local spectral descriptor for 3d partial shape retrieval**. Dans *Eurographics workshop on 3D object retrieval*. Citeseer.
- [Lavoué, 2012] Lavoué, G. (2012). **Combination of bag-of-words descriptors for robust partial shape retrieval**. *The Visual Computer*, 28(9) :931–942.
- [Lee et al., 2005] Lee, C. H., Varshney, A., et Jacobs, D. W. (2005). **Mesh saliency**. Dans *ACM transactions on graphics (TOG)*, volume 24, pages 659–666. ACM.

- [Li et al., 2012] Li, B., Godil, A., Aono, M., Bai, X., Furuya, T., Li, L., López-Sastre, R. J., Johan, H., Ohbuchi, R., Redondo-Cabrera, C., et others (2012). **Shrec'12 track : Generic 3d shape retrieval**. Dans *3DOR*, pages 119–126.
- [Li et al., 2013] Li, B., Heng, L., Koser, K., et Pollefeys, M. (2013). **A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern**. Dans *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1301–1307. IEEE.
- [Li et al., 2014] Li, C., et Hamza, A. B. (2014). **Symmetry discovery and retrieval of nonrigid 3d shapes using geodesic skeleton paths**. *Multimedia tools and applications*, 72(2) :1027–1047.
- [Ling et al., 2005] Ling, H., et Jacobs, D. W. (2005). **Using the inner-distance for classification of articulated shapes**. Dans *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 719–726. IEEE.
- [Ling et al., 2007] Ling, H., et Jacobs, D. W. (2007). **Shape classification using the inner-distance**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2) :286–299.
- [Liu et al., 2004] Liu, R., et Zhang, H. (2004). **Segmentation of 3d meshes through spectral clustering**. Dans *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pages 298–305. IEEE.
- [Liu et al., 2007] Liu, R., et Zhang, H. (2007). **Mesh segmentation via spectral embedding and contour analysis**. Dans *Computer Graphics Forum*, volume 26, pages 385–394. Wiley Online Library.
- [Lowe, 1999] Lowe, D. G. (1999). **Object recognition from local scale-invariant features**. Dans *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. IEEE.
- [Lowe, 2004] Lowe, D. G. (2004). **Distinctive image features from scale-invariant keypoints**. *International journal of computer vision*, 60(2) :91–110.
- [Lv et al., 2009] Lv, L., Liu, Y., Zhang, W., Jiang, C., Chen, X., Chang, T., et Yuen, M. (2009). **Shape profile matching and its applications**. *Journal of Software*, 4(5) :413–421.
- [Mahmoudi et al., 2009] Mahmoudi, M., et Sapiro, G. (2009). **Three-dimensional point cloud recognition via distributions of geometric distances**. *Graphical Models*, 71(1) :22 – 31.
- [Manning et al., 2008] Manning, C. D., Raghavan, P., Schütze, H., et others (2008). **Introduction to information retrieval**, volume 1. Cambridge university press Cambridge.
- [Mc Donald et al., 2005] Mc Donald, K., et Smeaton, A. F. (2005). **A comparison of score, rank and probability-based fusion methods for video shot retrieval**. Dans *Image and video retrieval*, pages 61–70. Springer.
- [McWherter et al., 2001] McWherter, D., Peabody, M., Shokoufandeh, A. C., et Regli, W. (2001). **Database techniques for archival of solid models**. Dans *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 78–87. ACM.
- [Meagher, 1980] Meagher, D. J. (1980). **Otree encoding : A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer**. Electrical and Systems Engineering Department Rensselaer Polytechnic Institute Image Processing Laboratory.

- [Melkemi et al., 2006] Melkemi, K. E., Batouche, M., et Fougou, S. (2006). **A multiagent system approach for image segmentation using genetic algorithms and extremal optimization heuristics**. *Pattern Recognition Letters*, 27(11) :1230–1238.
- [Meyer et al., 2003] Meyer, M., Desbrun, M., Schröder, P., et Barr, A. H. (2003). **Discrete differential-geometry operators for triangulated 2-manifolds**. Dans *Visualization and mathematics III*, pages 35–57. Springer.
- [Mikolajczyk et al., 2005] Mikolajczyk, K., et Schmid, C. (2005). **A performance evaluation of local descriptors**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10) :1615–1630.
- [Murala et al., 2012] Murala, S., Maheshwari, R., et Balasubramanian, R. (2012). **Local tetra patterns : a new feature descriptor for content-based image retrieval**. *Image Processing, IEEE Transactions on*, 21(5) :2874–2886.
- [Nadler et al., 2005] Nadler, B., Lafon, S., Coifman, R. R., et Kevrekidis, I. G. (2005). **Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators**. Dans *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press.
- [Ohbuchi et al., 2010] Ohbuchi, R., et Furuya, T. (2010). **Distance metric learning and feature combination for shape-based 3d model retrieval**. Dans *Proceedings of the ACM workshop on 3D object retrieval*, pages 63–68. ACM.
- [Ohbuchi et al., 2008] Ohbuchi, R., Osada, K., Furuya, T., et Banno, T. (2008). **Salient local visual features for shape-based 3d model retrieval**. Dans *Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on*, pages 93–102. IEEE.
- [Ohkita et al., 2012] Ohkita, Y., Ohishi, Y., Furuya, T., et Ohbuchi, R. (2012). **Non-rigid 3d model retrieval using set of local statistical features**. Dans *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, pages 593–598. IEEE.
- [Osada et al., 2001] Osada, R., Funkhouser, T., Chazelle, B., et Dobkin, D. (2001). **Matching 3d models with shape distributions**. Dans *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 154–166.
- [Ovsjanikov et al., 2010] Ovsjanikov, M., Mérigot, Q., Mémoli, F., et Guibas, L. (2010). **One point isometric matching with the heat kernel**. Dans *Computer Graphics Forum*, volume 29, pages 1555–1564. Wiley Online Library.
- [Ovsjanikov et al., 2008] Ovsjanikov, M., Sun, J., et Guibas, L. (2008). **Global intrinsic symmetries of shapes**. Dans *Computer graphics forum*, volume 27, pages 1341–1348. Wiley Online Library.
- [Pascucci et al., 2007] Pascucci, V., Scorzelli, G., Bremer, P.-T., et Mascarenhas, A. (2007). **Robust on-line computation of reeb graphs : simplicity and speed**. Dans *ACM Transactions on Graphics (TOG)*, volume 26, page 58. ACM.
- [Pratikakis et al., 2010] Pratikakis, I., Spagnuolo, M., Theoharis, T., et Veltkamp, R. (2010). **A robust 3d interest points detector based on harris operator**. Dans *Eurographics workshop on 3D object retrieval*, volume 5. Citeseer.
- [Radwan et al., 2008] Radwan, A. A. A., Latef, B. A. A., Ali, A. M. A., et Sadek, O. A. (2008). **A new fitness function based genetic algorithm to improve information retrieval systems**. Dans *Third International Conference on Intelligent Computing and Information Systems*, volume 2, pages 715 – 721, Cairo, Egypt. World Academy of Science, Engineering and Technology.

- [Reuter et al., 2006] Reuter, M., Wolter, F.-E., et Peinecke, N. (2006). **Laplace–beltrami spectra as shape-dna of surfaces and solids**. *Computer-Aided Design*, 38(4) :342–366.
- [Riskin, 1984] Riskin, H. (1984). **Fokker-Planck Equation**. Springer.
- [Rousseau et al., 2006] Rousseau, D., Anand, G., et Chapeau-Blondeau, F. (2006). **Noise-enhanced nonlinear detector to improve signal detection in non-gaussian noise**. *Signal Processing*, 86(11) :3456–3465.
- [Rustamov, 2007] Rustamov, R. M. (2007). **Laplace-beltrami eigenfunctions for deformation invariant shape representation**. Dans *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233. Eurographics Association.
- [Sabater et al., 2002] Sabater, J., et Sierra, C. (2002). **Reputation and social network analysis in multi-agent systems**. Dans *Proceedings of the first international joint conference on Autonomous agents and multiagent systems : part 1*, pages 475–482. ACM.
- [Sari et al., 2015] Sari, K., Tighiouart, F., et Tighiouart, B. (2015). **Using unsupervised neural network approach to improve classification of satellite images**. *International Journal of Computer Applications in Technology*, 51(1) :3–8.
- [Sario et al., 1977] Sario, L., Nakai, M., Wang, C., et Chung, L. O. (1977). **Laplace-beltrami operator**. *Classification Theory of Riemannian Manifolds : Harmonic, quasi-harmonic and biharmonic functions*, pages 12–25.
- [Scilab Enterprises, 2012] Scilab Enterprises (2012). **Scilab : Le logiciel open source gratuit de calcul numérique**. Scilab Enterprises, Orsay, France.
- [Sharma et al., 2010] Sharma, A., Von Lavante, E., et Horaud, R. (2010). **Learning shape segmentation using constrained spectral clustering and probabilistic label transfer**. Dans *Computer Vision–ECCV 2010*, pages 743–756. Springer.
- [Skraba et al., 2010] Skraba, P., Ovsjanikov, M., Chazal, F., et Guibas, L. (2010). **Persistence-based segmentation of deformable shapes**. Dans *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 45–52. IEEE.
- [Smeets et al., 2009] Smeets, D., Fabry, T., Hermans, J., Vandermeulen, D., et Suetens, P. (2009). **Isometric deformation modelling for object recognition**. Dans *Computer Analysis of Images and Patterns*, pages 757–765. Springer.
- [Snoek et al., 2005] Snoek, C. G. M., Worring, M., et Smeulders, A. W. M. (2005). **Early versus late fusion in semantic video analysis**. Dans *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 399–402.
- [Song et al., 2014] Song, R., Liu, Y., Martin, R. R., et Rosin, P. L. (2014). **Mesh saliency via spectral processing**. *ACM Transactions on Graphics (TOG)*, 33(1) :6.
- [Stanko et al., 2016] Stanko, T., Hahmann, S., Bonneau, G.-P., et Saguin-Sprynski, N. (2016). **Smooth interpolation of curve networks with surface normals**. Dans *Eurographics 2016 Short Papers*.
- [Stone et al., 2000] Stone, P., et Veloso, M. (2000). **Multiagent systems : A survey from a machine learning perspective**. *Autonomous Robots*, 8(3) :345–383.
- [Sturges, 1926] Sturges, H. A. (1926). **The choice of a class interval**. *Journal of the american statistical association*, 21(153) :65–66.

- [Sudha et al., 2014] Sudha, L., et Bhavani, R. (2014). **A combined classifier knn–svm in gait-based biometric authentication system**. *International Journal of Computer Applications in Technology*, 49(2) :113–121.
- [Sun et al., 2009] Sun, J., Ovsjanikov, M., et Guibas, L. (2009). **A concise and provably informative multi-scale signature based on heat diffusion**. Dans *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library.
- [Sundar et al., 2003] Sundar, H., Silver, D., Gagvani, N., et Dickinson, S. (2003). **Skeleton based shape matching and retrieval**. Dans *Shape Modeling International, 2003*, pages 130–139. IEEE.
- [Syam et al., 2013] Syam, B., et Rao, Y. (2013). **An effective similarity measure via genetic algorithm for content based image retrieval with extensive features**. *International Arab Journal of Information Technology (IAJIT)*, 10(2).
- [Taati et al., 2011] Taati, B., et Greenspan, M. (2011). **Local shape descriptor selection for object recognition in range data**. *Computer Vision and Image Understanding*, 115(5) :681–694.
- [Tang et al., 2009] Tang, F., Lim, S. H., Chang, N. L., et Tao, H. (2009). **A novel feature descriptor invariant to complex brightness changes**. Dans *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2631–2638. IEEE.
- [Tang et al., 2004] Tang, J., Lim, M.-H., Ong, Y.-S., et Er, M. J. (2004). **Study of migration topology in island model parallel hybrid-ga for large scale quadratic assignment problems**. Dans *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, volume 3, pages 2286–2291. IEEE.
- [Tangelder et al., 2008] Tangelder, J., et Veltkamp, R. (2008). **A survey of content based 3d shape retrieval methods**. *Multimedia Tools and Applications*, 39(3) :441–471.
- [Tax et al., 1997] Tax, D. M., Duin, R. P., et Breukelen, M. V. (1997). **Comparison between product and mean classifier combination rules**. Dans *In Proc. Workshop on Statistical Pattern Recognition*, pages 165–170.
- [Thada et al., 2013] Thada, V., et Jaglan, V. (2013). **Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm**. *International Journal of Innovations in Engineering and Technology*.
- [Thayananthan et al., 2003] Thayananthan, A., Stenger, B., Torr, P. H., et Cipolla, R. (2003). **Shape context and chamfer matching in cluttered scenes**. Dans *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages 1–127. IEEE.
- [Thompson et al., 2010] Thompson, J., et Flynn, P. (2010). **A segmentation perturbation method for improved iris recognition**. Dans *Biometrics : Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–8.
- [Thürner et al., 1998] Thürner, G., et Wüthrich, C. A. (1998). **Computing vertex normals from polygonal facets**. *Journal of Graphics Tools*, 3(1) :43–46.
- [Tian et al., 2005] Tian, H., He, Y., et Wu, Y. (2005). **A new approach of progressive spherical parameterization**. Dans *Computer Aided Design and Computer Graphics, 2005. Ninth International Conference on*, pages 5–pp. IEEE.

- [Toldo et al., 2009] Toldo, R., Castellani, U., et Fusiello, A. (2009). **Visual vocabulary signature for 3d object retrieval and partial matching**. Dans *Proceedings of the 2Nd Eurographics Conference on 3D Object Retrieval*, EG 3DOR'09, pages 21–28, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- [Torres et al., 2004] Torres, R. d. S., Falcao, A. X., et Costa, L. d. F. (2004). **A graph-based approach for multiscale shape analysis**. *Pattern Recognition*, 37(6) :1163–1174.
- [Tung, 2005] Tung, T. (2005). **Indexation 3D de bases de données d'objets par graphes de Reeb améliorés**. PhD thesis, Télécom ParisTech.
- [Veltkamp et al., 2001] Veltkamp, R., et Hagedoorn, M. (2001). **State of the art in shape matching**. *Principles of visual information retrieval*, page 87.
- [Vignes, 1993] Vignes, J. (1993). **A stochastic arithmetic for reliable scientific computation**. *Mathematics and computers in simulation*, 35(3) :233–261.
- [West et al., 2001] West, D. B., et others (2001). **Introduction to graph theory**, volume 2. Prentice hall Upper Saddle River.
- [YA. M. Bronstein, 2007] YA. M. Bronstein, M. M. Bronstein, R. K. (2007). **Calculus of nonrigid surfaces for geometry and texture manipulation**. *Visualization and Computer Graphics, IEEE Transactions on*, 13/5 :902–913.
- [Yang et al., 2008] Yang, X., Bai, X., Latecki, L. J., et Tu, Z. (2008). **Improving shape retrieval by learning graph transduction**. Dans *Computer Vision—ECCV 2008*, pages 788–801. Springer.
- [Yin et al., 2013] Yin, F., Zhou, M.-K., Wang, Q.-F., et Liu, C.-L. (2013). **Style consistent perturbation for handwritten chinese character recognition**. Dans *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1051–1055.
- [Young et al., 2007] Young, S., Adelstein, B., et Ellis, S. (2007). **Calculus of non-rigid surfaces for geometry and texture manipulation**. *Visualization and Computer Graphics, IEEE Transactions on*, 13(5) :902–913.
- [Zaharia et al., 2002] Zaharia, T., et Preteux, F. (2002). **Indexation de maillages 3D par descripteurs de forme**. Dans *13ème Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA'2002)*, pages 48–57.
- [Zhang et al., 2002] Zhang, D., et Lu, G. (2002). **Shape-based image retrieval using generic fourier descriptor**. *Signal Processing : Image Communication*, 17(10) :825–848.
- [Zhang et al., 2003] Zhang, H., et Malik, J. (2003). **Learning a discriminative classifier using shape context distances**. Dans *null*, page 242. IEEE.
- [Zhang et al., 2015] Zhang, S., Zhang, G., et Zhao, X. (2015). **Robust facial expression recognition using improved sparse classifier**. *International Journal of Computer Applications in Technology*, 52(1) :59–70.
- [Zou et al., 2015] Zou, K., Zhang, Z., Zhang, J., et Zhang, Q. (2015). **A 3d model feature extraction method using curvature-based shape distribution**. Dans *Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on*, pages 1809–1813. IEEE.

TABLE DES FIGURES

1.1	Ces deux objets sont-ils similaires ?	4
1.2	Types de descripteurs de formes 3D.	5
1.3	Le contexte de forme pour un objet 3D [Körtgen et al., 2003].	6
1.4	Des exemples de modèles 3D avec leur descripteur de symétrie [Kazhdan et al., 2002].	7
1.5	Un exemple de descripteur de points d'intérêt [Pratikakis et al., 2010]. Les points en vert représentent les points d'intérêt.	8
1.6	La saillance de deux modèles 3D [Lee et al., 2005]. Les régions saillantes sont représentées avec des couleurs proches du rouge.	9
1.7	Le contexte de forme d'un modèle 3D utilisant les coordonnées log-polaires dans le voisinage d'un point [Kokkinos et al., 2012].	10
1.8	Un modèle 3D et son graphe de Reeb associé [Pascucci et al., 2007].	11
1.9	Graphe de Reeb multirésolution à 1, 3, et 11 composantes [Hilaga et al., 2001].	12
1.10	Exemples de modèles 3D et leur squelettes associés [Sundar et al., 2003].	13
1.11	Descripteur de points caractéristiques en utilisant une méthode basée sur la prise de vues pour un modèle 3D [Ohbuchi et al., 2008].	14
1.12	Distribution de formes D2 [Osada et al., 2001].	14
1.13	Descripteur de diamètre local [Li et al., 2012].	14
1.14	Sac de mots visuels [Lavoué, 2011].	15
1.15	Le descripteur de diffusion de chaleur calculé pour différents modèle [Sun et al., 2009]. Les valeurs du descripteur calculées pour les points sont représentées par une carte de couleurs graduée du bleu (petite valeur) au rouge (grande valeur).	17
1.16	Détection des symétries pour un objet articulé [Sharma et al., 2010].	18
1.17	Segmentation d'un modèle 3D en utilisant la méthode proposée dans [Liu et al., 2007]. A gauche : le modèle segmenté. A droite : la représentation spectrale du modèle.	19
1.18	Détection des symétries pour un objet avec changement de posture [Ovsjanikov et al., 2008].	20
1.19	Histogrammes des sommets calculés en utilisant le descripteur WKS [Aubry et al., 2011].	21

1.20	Le descripteur de diffusion de chaleur calculé pour différents modèles [Sun et al., 2009]. Les valeurs du descripteur calculées pour les points sont représentées par une carte de couleurs graduée du bleu (petite valeur) au rouge (grande valeur).	21
1.21	Calcul de similarité et symétrie en utilisant la méthode proposée dans [Ovsjanikov et al., 2010]. (a) : correspondance entre un modèle entier et une partie d'un modèle. (b) : détection de symétrie dans des modèles. . . .	22
1.22	Un exemple de segmentation en utilisant la méthode de combinaison de HKS et PBC [Skraba et al., 2010].	22
2.1	Un exemple d'images binaires représentant des modèles 2D.	26
2.2	Un exemple d'un objet polygonale. gauche : facettes triangulaires. droite : objet de synthèse obtenu en utilisant le lissage de Gouraud.	26
2.3	Un tore modélisé par la surface de Bézier http://morpheo.inrialpes.fr/people/Boyer/Teaching/RICM/c5.pdf	26
2.4	Exemple d'objets CSG construits en utilisant différentes opérations https://en.wikipedia.org/wiki/Constructiv_solid_geometry	27
2.5	Différents niveaux de subdivisions pour un maillage rectangulaire [Boumzaid et al., 2009]. haut droite : maillage original.	28
2.6	Représentation volumique d'un objet 3D. (a) : modèle polygonal. (b) : modèle volumique.	30
2.7	Les éléments de la base de données MPEG-7.	31
2.8	Les différents modèles de la base de données TOSCA.	31
2.9	Les différents modèles de la base de données SHREC'11.	32
3.1	Un objet 3D décomposé en 3, 4, 5, 7, 12, classes.	37
3.2	Un exemple de classification 1D. Gauche : un ensemble de nombres entiers. Droite : partitionnement des nombres entiers en classes.	39
3.3	Un exemple 2D de partitionnement en utilisant l'algorithme des nuées dynamiques. Nous enregistrons les différentes valeurs des variance intra/inter classes selon le nombre de classes k	40
3.4	Variation des valeurs intra/inter classes selon le nombre de classes k . $var_intra(k)$ est une fonction descendante, $var_inter(k)$ est une fonction unimodale.	41
3.5	Objets 2D décomposés en k classes optimales.	41
3.6	Squelette d'ellipsoïde d'un objet 2D.	42
3.7	Un exemple d'un partitionnement contenant des classes inconsistantes. . .	43
3.8	Décomposition d'une classe en sous classes.	43

3.9	Les centres utilisés pour initialiser l'algorithme des nuées dynamiques. P_1 et P_2 sont les deux points initiaux (les points les plus éloignés). (b) Les deux points les plus éloignés dans leur classe c_1 and c_2 . Si $d_1 < d_2$, P_1 est le nouveau centre, sinon c'est P_2	44
3.10	Décomposition adéquate d'objets en squelettes d'ellipsoïdes.	45
3.11	Exemples d'objets 3D décomposés en squelette d'ellipsoïdes. Ligne 1 : modèles 3D. Ligne 2 : squelette d'ellipsoïdes 3D.	45
3.12	Exemples d'objets 2D décomposés en squelette d'ellipsoïdes. Ligne 1 : modèles 2D. Ligne 2 : squelette d'ellipsoïdes 2D.	45
3.13	Un exemple d'un couplage optimal dans un graphe biparti pour deux formes 2D.	47
3.14	Résultats de recherche pour des exemples de requêtes.	49
4.1	Coût défini en termes des angles opposés à l'arête reliant deux sommets voisins [Meyer et al., 2003].	53
4.2	Différentes mesures de distance entre deux sommets sur la surface d'un modèle [Dubrovina et al., 2010].	54
4.3	Carte de couleurs des valeurs associées aux sommets calculée en utilisant le descripteur de la géométrie de diffusion (GD). En bleu les petites valeurs, et en rouge les grandes valeurs.	54
4.4	Carte de couleurs des valeurs associés aux sommets calculées en utilisant les différents descripteurs	58
4.5	Carte de couleurs représentant les valeurs et les vecteurs propres de la matrice Laplacienne [Dubrovina et al., 2010].	65
4.6	Graphe G_1 associé à une chaîne de Markov avec les mêmes probabilités pour les arêtes sortantes d'un sommet donné.	67
4.7	Graphe G_2 associé à une chaîne de Markov, considérant des probabilités distinctes pour les arêtes sortantes d'un sommet donné.	67
5.1	Chaîne de Markov à deux états représentant le degré de confiance en chaque descripteur.	74
5.2	Courbes de précision-rappel : résultats du processus de recherche pour la base de données TOSCA, en utilisant les différentes règles de fusion. . . .	75
5.3	Un exemple de descripteur S calculé pour différentes formes. A gauche : le modèle original. Au centre : la carte de couleurs correspondant aux valeurs du descripteur calculés en chaque sommet du modèle. A droite : les histogrammes correspondants.	76
5.4	Un exemple des histogrammes calculés en utilisant les différents descripteurs. A gauche : le descripteur S_1 . Au milieu ; le descripteur S_2 . A droite : La fusion des deux descripteurs.	77
5.6	Courbes de précision-rappel. (a) : résultats en utilisant les descripteurs S_1 , S_2 , et S	78

5.5	Résultats de recherche en utilisant S_1 , S_2 et S . Colonne 1 : une requête donnée. Colonnes i : la liste des objets trouvés pour une requête. Les erreurs de la recherche sont entourées en rouge. S_1 et S_2 commettent des erreurs, S n'en commet pas.	79
5.7	Matrice de distance de la base de donnée. Les objets dissimilaires sont représentés en clair.	79
5.8	Courbes précision-rappel pour la comparaison des descripteurs.	82
6.1	Exemples de modèles 3D des bases de données <i>TOSCA</i> [Young et al., 2007] et <i>SHREC'11</i> [Li et al., 2012].	89
6.2	Exemples de modèles CAD de la base de données ESB [Jayanti et al., 2006].	89
6.3	Les courbes précision-rappel des éléments de la base de données <i>TOSCA</i> . Dans chaque graphique, les courbes bleu correspondent aux distances (d_i) induites par les descripteurs utilisés, et les courbes en violet correspondent aux distances géodésiques d_i^* . La courbe en rouge correspond aux résultats de notre méthode.	90
6.4	Les courbes précision-rappel des éléments de la base de données ESB. Les courbes en bleu correspondent à la distance d_5 induite par le descripteur CH, et les courbes en violet à la distance géodésique d_5^* (longueur du plus court chemin) dans le graphe G_5 . La courbe en rouge correspond aux résultats de notre méthode.	91
6.5	Courbes de précision-rappel des objets de la base de données <i>TOSCA</i> . La courbe en bleu correspond aux résultats de combinaison des distances d_i , la courbe en violet des distances d_i^* . La courbe en rouge correspond aux résultats de combinaison des distances induites par les descripteurs LFD, GD, DE, et D2 en utilisant notre méthode.	91
6.6	Courbes de précision-rappel calculées pour des objets de la base de données <i>TOSCA</i> . Graphique à gauche : la courbe en rouge correspond à la combinaison des distance d_1 et d_2 avec notre méthode. A droite : la combinaison des distances d_3 et d_4	91
6.7	Courbes de précision-rappel pour un ensemble d'objets de la base de données <i>TOSCA</i> , calculées en utilisant les différentes règles de fusion pour les distances d_i	92
6.8	Les courbes précision-rappel pour les éléments des bases de données <i>TOSCA</i> et <i>SHREC'11</i> en utilisant notre méthode.	94
6.9	Les courbes précision-rappel des modèles des trois catégories de la base de données ESB, en utilisant la combinaison des distances d_4 et d_5	94
7.1	Processus de l'algorithme génétique.	101
7.2	Un exemple de la technique de perturbation appliquée sur un modèle (Shark, <i>TOSCA</i> [YA. M. Bronstein, 2007]). En haut à gauche : le modèle original. En haut à droite : les sommets sélectionnés pour être perturbés. En bas à gauche : les sommets sélectionnés perturbés avec $\epsilon = 0.03$. En bas à droite : les sommets sélectionnés perturbés avec $\epsilon = 0.1$	103

7.3	Les distances entre un modèle David 1 et les clones d'un autre modèle David 2 générés en considérant différentes valeurs d' ϵ , et en utilisant les descripteurs VND, LFD, TD, DE, et LSD. En abscisse, la valeur d' ϵ ; en ordonné, la distance entre un modèle (David 1) et des clones d'un autre modèle, David 2, de la même classe. Pour $\epsilon = 0$, la distance n'est pas nulle, et vaut la distance entre David 1 et l'original (non perturbé) David 2. Pour de petites valeurs de ϵ , la distance est inférieure à D_0 , et augmente lentement, proportionnellement à ϵ , et fait par atteindre D_0 . Puis, au delà d'une certaine valeur, la distance augmente avec une pente nettement plus grande.	104
7.4	La précision moyenne de toute la base de données (TOSCA [YA. M. Bronstein, 2007]), en fonction du coefficient de perturbation ϵ , en utilisant les différents descripteurs de formes.	105
7.5	La moyenne des distances (mesurée avec d) entre Q_i et d'une part, les clones de toutes les requêtes possibles dans la base de données ([YA. M. Bronstein, 2007]), et d'autre part les m formes dans la base de données les plus proches de Q_i . La courbe montre la variation de la moyenne des distances en fonction du nombre de sommets perturbés $\mu = \lceil PV \rceil$ (nous avons utilisé l' ϵ optimal pour chaque descripteur de forme, par exemple $\epsilon = 0.03$ pour le descripteur VND).	106
7.6	Un exemple de codage du génotype d'un modèle : sommets perturbés en rouge (la sous-figure en haut), le génotype (sous-figure du bas).	107
7.7	Un exemple de l'opération de croisement pour deux clones d'une posture du modèle David (TOSCA [YA. M. Bronstein, 2007]).	109
7.8	Un exemple de mutation appliquée sur un modèle. Le sommet vert sur le modèle à gauche représente le sommet muté (remplacé par un nouveau sommet dans le modèle à droite).	110
7.9	Comparaison des précisions de notre algorithme selon les deux fonctions d'aptitude.	112
7.10	L'architecture de notre MAS-SR-GA.	115
7.11	Les courbes précision-rappel comparant les différents descripteurs avec le GA appliqué à ces derniers. La base de données est SHREC'11.	116
7.12	Un exemple d'un modèle lissé considérant différentes valeurs de α	117
7.13	Les distances moyennes entre des requêtes et les objets de la base de données TOSCA, en fonction du paramètre α utilisé pour le lissage.	118
7.14	Les courbes précision-rappel comparant les résultats de notre algorithme avec ceux du lissage.	119
7.15	La comparaison des résultats de notre méthode avec d'autres méthodes proposées dans la littérature. Les courbes sont tracées en fonction des résultats rapportés sur des objets de la base de données TOSCA.	120

LISTE DES TABLES

3.1	Valeurs des variances <i>intra_inter</i> classes selon le nombre de classes k . . .	38
3.2	Partitionnement de la base de données. Le tableau montre la répartition des objets de la base de données en différents groupes, en utilisant notre algorithme.	48
3.3	Matrice de distances entre des objets similaires. Les valeurs sont petites, contrairement aux distances entre les objets dissimilaires (Tableaux 3.4, 3.5, 3.6.)	48
3.4	Matrice de distances pour des objets différents (<i>Fish, Children</i>)	48
3.5	Matrice de distances pour des objets différents (<i>Bone, Bottle</i>)	48
3.6	Matrice de distance pour des objets différents (<i>Key, Butterfly</i>)	48
3.7	La précision du processus de recherche pour des objets 2D et 3D.	49
5.1	La précision moyenne du processus de recherche pour chaque catégorie d'objets de la base de données utilisée.	80
5.2	La précision moyenne de la base de données en utilisant différents descripteurs.	80
5.3	Temps de calcul nécessaire pour chaque descripteur, pour chaque catégorie de la base de données TOSCA.	82
6.1	Exemples des résultats de précisions pour des modèles choisis aléatoirement dans les différentes classes de la base de données ESB	93
7.1	Intervalle de sélection.	108
7.2	La précision du meilleur clone à chaque itération de l'algorithme génétique : la fonction aptitude #1 est utilisée. On indique en caractère gras les meilleures précision dans chaque génération.	112
7.3	L'évolution de la fonction d'aptitude du meilleur clone Q^* au cours des générations de l'algorithme génétique : la fonction aptitude#1 est utilisée. . .	112
7.4	La précision du meilleur clone à chaque itération de l'algorithme génétique : la fonction aptitude#2 est utilisée.	113
7.5	L'évolution de la fonction d'aptitude du meilleur clone Q^* au cours des générations de l'algorithme génétique : la fonction aptitude#1 est utilisée. . . .	113
7.6	Temps des calculs.	121

Résumé :

Cette thèse porte sur l'appariement des formes, et la recherche par forme clef. Elle décrit quatre contributions à ce domaine. La première contribution est une amélioration de la méthode des nuées dynamiques pour partitionner au mieux les voxels à l'intérieur d'une forme donnée; les partitions obtenues permettent d'apparier les objets par un couplage optimal dans un graphe biparti. La seconde contribution est la fusion de deux descripteurs, l'un local, l'autre global, par la règle du produit. La troisième contribution considère le graphe complet, dont les sommets sont les formes de la base ou la requête, et les arêtes sont étiquetées par plusieurs distances, une par descripteur; ensuite cette méthode calcule par programmation linéaire la combinaison convexe des distances qui maximise soit la somme des longueurs des plus courts chemins entre la requête et les objets de la base de données, soit la longueur du plus court chemin entre la requête et l'objet comparé à la requête. La quatrième contribution consiste à perturber la requête avec un algorithme génétique pour la rapprocher des formes de la base de données, pour un ou des descripteur(s) donné(s); cette méthode est massivement parallèle, et une architecture multi-agent est proposée. Ces méthodes sont comparées aux méthodes classiques, et ont de meilleures performances, en terme de précision.

Mots-clés : Appariement de formes, descripteurs de formes, recherche par forme clef

Abstract:

This thesis concerns shape matching and shape retrieval. It describes four contributions to this domain. The first is an improvement of the k -means method, in order to find the best partition of voxels inside a given shape; these best partitions permit to match shapes using an optimal matching in a bipartite graph. The second contribution is the fusion of two descriptors, one local, the other global, with the product rule. The third contribution considers the complete graph, the vertices of which are the shapes in the database and the query. Edges are labelled with several distances, one per descriptor. Then the method computes, with linear programming, the convex combination of distances which maximizes either the sum of the lengths of all shortest paths from the query to all shapes of the database, or the length of the shortest path in the graph from query to the current shape compared to query. The fourth contribution consists in perturbing the shape query, to make it closer to shapes in the database, for any given descriptors. This method is massively parallel and a multi-agent architecture is proposed. These methods are compared to classical methods in the field, they achieve better retrieval performances.

Keywords: Shape matching, shape descriptors, dissimilarity measures, shape retrieval

The logo for the SPIM (École doctorale SPIM) features a stylized orange horizontal bar on the left, followed by the letters 'S', 'P', 'I', and 'M' in a large, white, sans-serif font.