Thèse présentée pour l'obtention du grade de Docteur de Télécom SudParis dans le cadre de l'école doctorale "Sciences Pour l'Ingénieur" pour laquelle l'Université Lille 1 est accréditée, et Télécom SudParis partenaire.

Thèse n° 2013TELE0009

Doctorat conjoint

Télécom SudParis et Université Lille 1

Spécialité: Informatique

par

Rachid El Khoury

Partial 3D-shape indexing and retrieval

Soutenue le 22 Mars 2013 devant le jury composé de :

M.	Florent Dupont	Professeur, Université Lyon 1	(Rapporteur)
M.	William Puech	Professeur, Université Montpellier 2	(Rapporteur)
M.	François Goulette	Professeur, Mines ParisTech	(Examinateur)
M.	Julien Tierny	Chargé de Recherches, CNRS	(Examinateur)
M.	Jean-Philippe Vandeborre	Maître de Conférences HDR, Télécom Lille1	(Directeur)
M.	Mohamed Daoudi	Professeur, Télécom Lille 1	(Co-Directeur)

This manuscript is dedicated
to all those who contributed, from near or far, to its elaboration.

ACKNOWLEDGMENTS

First of all I would like to thank my different supervisors, Pr. Mohamed Daoudi and Dr. Jean-Philippe Vandeborre, with whom I really appreciated to work this last three years and I discovered the scientific research world, in particular that of the 3D-multimedia field. I also thank them for their trust, daily encouragements, and their useful advises.

Special thank goes to the members of committee, in particular, M. Florent Dupont (professeur, Université Lyon 1) and M. William Puech (professeur, Université Montpellier 2) for accepting to review my thesis and for their interesting comments. I also thank the rest of the members; François Goulette (professeur, Mines ParisTech) and M. Julien Tierny (senior researcher, CNRS). All these people made me the honour of being present the day of my oral presentation despite their busy schedules.

I also thank the different members of the MIIRE and FOX teams who were more than colleagues! I thank them for the very good atmosphere of work, for their help on a technical level, and for all the special events that we organized together!

Special thanks goes to my lovely fiancée Carine who made me the happiest person and supported me during these three years.

I would like to express my gratitude to my parents and my brothers (Georges, Roger, Carlos) who always encouraged me and supported me.

I thank my dearest cousins and friends (Paul, Paula, Nohra,...) who tried to come to my Ph.D defence, without forgetting all my other best friends and lovely cousins.

Finally, I thank people at TELECOM Lille 1, especially the technical SIR service for their kindness.

Author's publications

Publications and awards:

- **El Khoury Rachid**, Mohamed Daoudi, Jean-Philippe Vandeborre "Indexed Heat Curves for 3D-model Retrieval" *Signal, Speech and Image/Video Processing, International Conference on Pattern Recognition (ICPR)*, November 2012, Tsukuba, JAPAN.

- **El Khoury Rachid**, Mohamed Daoudi, Jean-Philippe Vandeborre "Fonction de noyau de chaleur pour les graphes de Reeb de maillages 3D" *Compression et Représentation des Signaux Audiovisuels (CORESA'2012)*, May 2012, Lille, France.

- **El Khoury Rachid**, Mohamed Daoudi, Jean-Philippe Vandeborre "3D-Mesh Reeb Graph Computation Using Commute-time and Diffusion Distances" *Electronic Imaging Conference, 3D Image Processing (3DIP) and Applications*, January 2012 , San Francisco, USA.

- **Communication awards**
  Seminar " *des Doctoriales franco belges 2012* ", June 2012, Brussels.

CONTENTS

List of Figures

INTRODUCTION

<div style="text-align: right">1</div>

THREE dimensional (3D) object representations have become a major interactive graphical experience. A growing number of 3D graphic applications have an impact on today's society. These applications are being used in several domains ranging from digital entertainment, computer aided design, to medical applications (see figure 1.1).

The rapid evolution of this domain has created the need for 3D processing and analysing tools at the geometric, topological, and semantic levels. In the same vein, the evolution of 3D databases which grow in numbers and details precision, such as Google 3D-warehouse (a collection of 3D-data such as buildings, bridges, cars and so on). Google 3D-warehouse is getting bigger every day and it is completely accessible for every one to use. This evolution leads to look for a system that can automatically retrieve the 3D-models visually similar to a requested 3D-object. The problem of searching similar 3D objects arises in a number of fields. For example, in the area of medicine, the detection of similar organ deformations can be used for diagnostic purposes. Another example, in the industrial applications, a 3D model of the client's foot is generated using a 3D scanning tool. Then a similarity search is performed to retrieve the most likely fitting models according to the client's foot.

In this context, a 3D object search engine with a good performance in time consuming and results becomes a need for the society. A few 3D search engines that are available on-line such as *3D model Search Engine* proposed by the Princeton Shape and Analysis Group[1] presented in figure

---

[1] http://shape.cs.princeton.edu/search.html.
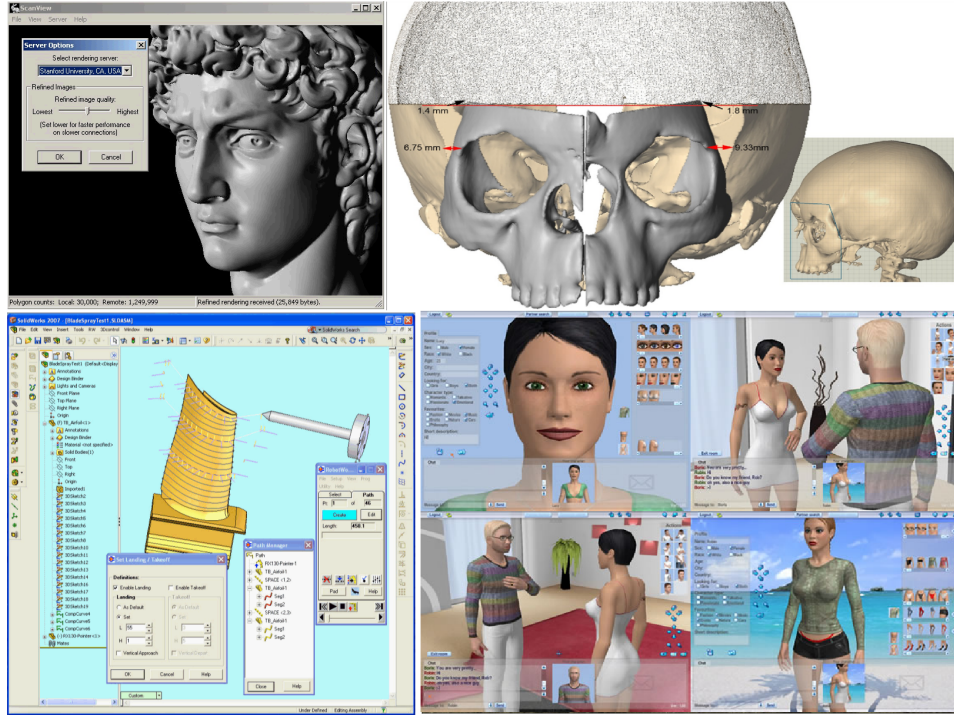
Figure 1.1 – *From left to right and top to bottom: Remote visualization of stored 3D models (©the Leland Stanford Junior University), life-changing facial reconstruction for young Child (©Sensable), solid modelling 3D-CAD design (©Compucraft, Ltd), AVChat3D 3D-video game (©Windows 8 App Store).*

1.2, the Semantic3D project search engine[2], the Taiwan University search engine[3] and Dejan Vranić's 3D Search Engine[4].

These 3D search engines are still limited in performance. Most proposed approaches for content-based retrieval are using statistical histograms that measure some geometric characteristics of 3D objects. The various measures of such characteristics are calculated from a 3D-triangulated mesh surfaces (see figure 1.3). The mesh representation is the simplest and the most frequently used.

In the literature, different kind of shape retrieval approaches exist. The oldest methods are inspired from 2D-methods like Fourier descriptors (30), invariant moments calculations (14), median line extraction, angular radial transform (89), etc. Shape description literature is very rich. Global 3D-shape description approaches generally characterize the shape of objects in a grossly way. They are often effective to discriminate simple

---

[2] http://www-rech.telecom-lille1.eu/3dretrieval/.

[3] http://3d.csie.ntu.edu.tw/.

[4] http://merkur01.inf.uni-konstanz.de/CCCC/.
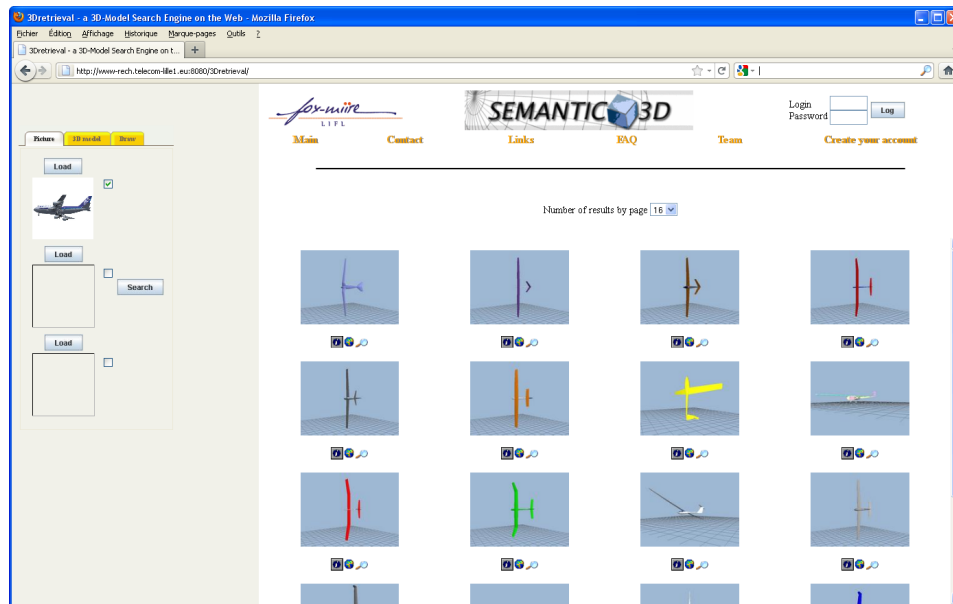
Figure 1.2 – *3D search engine of the MIIRE research team.*
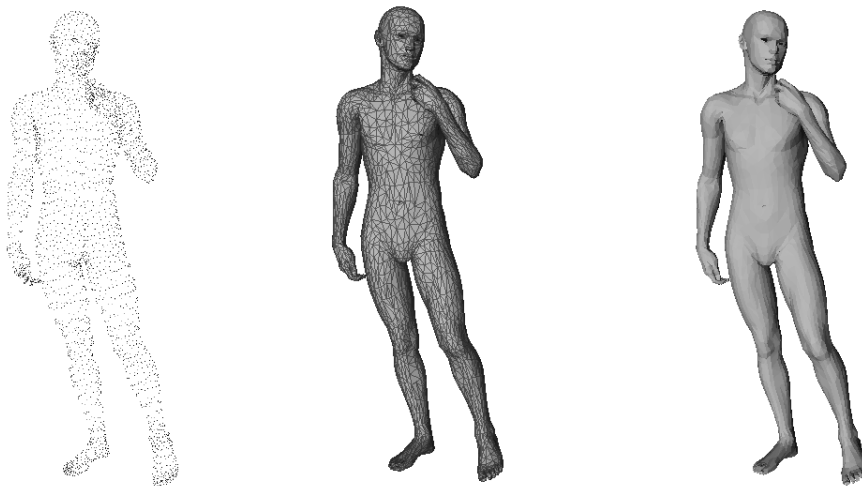
Figure 1.3 – *An example of 3D-object. From left to right: points, flat lines and flat rendering.*
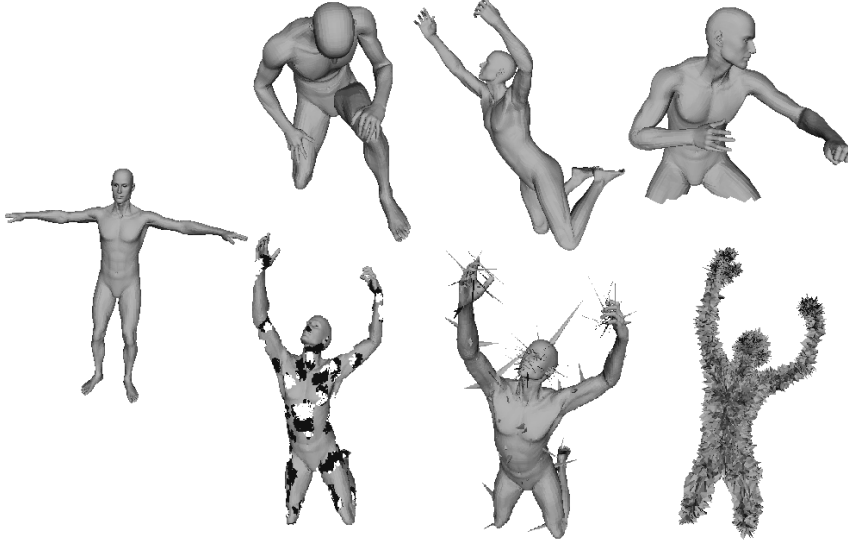
Figure 1.4 – *3D-objects under some deformations and transformations (affine, isometric, scaling, partial, noisy, etc.)*

shapes but insufficiently discriminative for more complex shapes. Conversely, 3D-local approaches characterize the local properties of the object's surface.

Existing solutions for 3D-shape retrieval are quite robust with respect to rigid transformation like translation, rotation or even scale change. However, figure 1.4 presents the acquired 3D shapes that suffer from high variability towards transformations (like isometric or affine transformation), representation degradations (like noise, sampling or scaling variation) or partial representation (like model with missing parts or combined with other). As a consequence, it leads to different descriptors for the same shape under these transformations that affect the results and the performance of the 3D search engine.

The aim of this thesis is to present an approach for 3D-model retrieval that can be robust under non-rigid transformations as well as rigid transformations and also handle 3D-partial retrieval.

1.1 CONTRIBUTIONS

In this thesis, as a start point, we believe that global descriptors are not very efficient for partial 3D-model retrieval. We need to define our signature based on a local descriptor. We have to decompose the 3D-models

into sub-parts, without forgetting that similar 3D-models should be decomposed similarly, then each part is affected to a local descriptor. The descriptor should describe the form of each part. We chose to describe each part by closed curves which encode the form of this part. In order to decompose similar 3D-model into sub part similarly, even if these parts do not have semantic meanings, we chose to define a scalar function on the surface and for each scale value we detect a region.

We start our investigation in scalar functions defined on surfaces, which in turn respects the main properties like the invariance to rigid and non-rigid transformations, the insensitivity to noise, the robustness to small topology changes, and the independence on parameters. The definition of such a function remains an open question. In this thesis, we respond to the question and we define an appropriate function on the surface. We prove the effectiveness of this function by transforming it to a Morse function then we computed the Reeb graph on a neutral pose 3d-model and against variations in 3D-model poses. Also, we tested it on different 3D-models including partial 3D-models and combined ones.

We continue by introducing a novel method for 3D-model retrieval based on indexed closed curves using our scalar function. We generate and analyse indexed closed curves raised from a source point (the source point is detected automatically by our scalar function) of the 3D-model using our scalar function. For each scale value of the function, we detect a region then we describe the form of this region by a closed curve. Finally, we analyse the 3D-model by analysing the shape of their corresponding level curves. We tested our approach on two datasets. Our experiments have good results but not very efficient for partial 3D-model retrieval.

Finally, we present a novel method, which in turn is an enhancement of our indexed closed curves method by using the bag of feature technique. Then we demonstrate the effectiveness of our approach on two sets of experiments. First set of experiments evaluate our approach for 3D-model retrieval and we compare our results to indexed closed curve approach. The second set of experiments validate our approach towards partial 3D-model retrieval.

1.2 Outline

The rest of this manuscript is illustrated in figure 1.5 and is laid out as follows.

In chapter 2, we introduce the assumption that the measure of similarity between two 3D-objects can be reduced to a distance computing between their two descriptors then we proceed by reviewing the existing solutions for 3D-model retrieval and for partial 3D-model retrieval.

In chapter 3, we present an approach to define our scalar function which in turn preserve invariance properties (see figure 1.5a).

In chapter 4, we apply our scalar function to compute Reeb graphs and to prove the invariance properties of our function (see figure 1.5b).

In chapter 5, we present a novel method based on indexed closed curves for 3D-model retrieval using our scalar functions (see figure 1.5c).

Chapter 6 is an enhancement of our indexed closed curves method by using the bag of feature technique for partial 3D-model retrieval (see figure 1.5d).

Finally, we conclude this manuscript by summarizing the contributions of this thesis, enumerate remaining open problems and propose directions for future research.

Introduction (en français)

La représentation tridimensionnelle d'objets (3D) est devenue une partie intégrante de différentes applications modernes, un nombre croissant de ces applications graphiques 3D ont un impact sur notre société. Elles sont utilisées dans plusieurs domaines allant des produits de divertissement numériques, de la conception assistée par ordinateur, aux applications médicales (voir figure 1.1).

L'évolution rapide de ce domaine a créé le besoin pour le traitement et l'analyse des outils 3D aux niveaux géométriques, topologiques et sémantiques. Dans le même esprit, les bases de données 3D se développent en nombre et en précision de détails. Par exemple, *Google 3D Warehouse* (une collection de données 3D telles que les bâtiments, les ponts, les

(a) *Chapter 3 is about definition of an invariant scalar function*
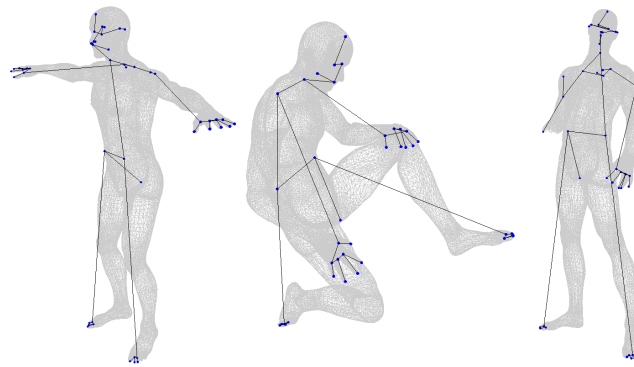
(b) *Chapter 4 is about Reeb graph computation*
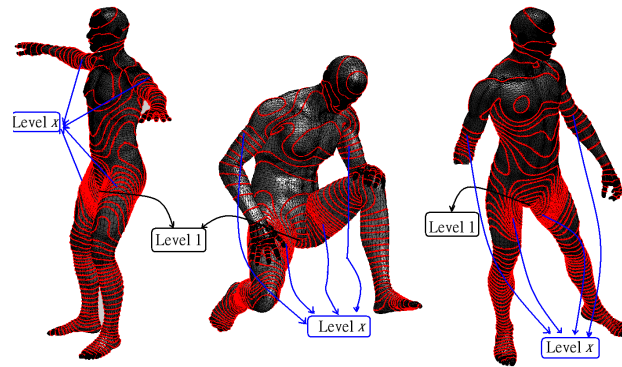
(c) *Chapter 5 is about indexed closed curves approach for 3D-model retrieval*
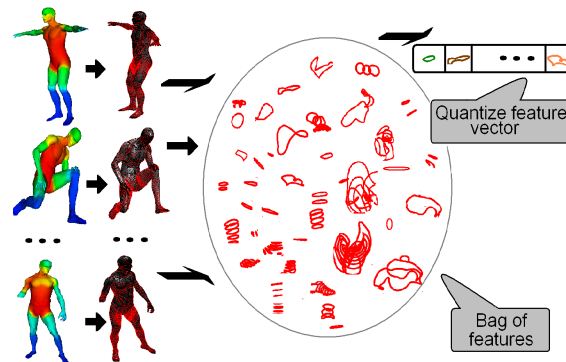
(d) *Chapter 6 is about bag of feature technique for 3D-model and partial 3D-model retrieval*

Figure 1.5 – *The outline of the rest of this manuscript.*

voitures, etc.) s'agrandit chaque jour et est accessible à tous. Cette évolu-
tion conduit à rechercher un système qui peut automatiquement récupérer
les modèles 3D visuellement similaires à un modèle 3D demandé. Le
problème de la recherche des objets 3D similaires se pose dans un cer-
tain nombre de domaines. Par exemple, dans le domaine de la médecine,
la détection des déformations des organes similaires permet des diagnos-
tics plus précis. Un autre exemple, dans les applications industrielles, un
modèle 3D du pied du client est généré en utilisant un outil de scan 3D.
Puis une recherche de similarité est effectuée pour extraire les modèles les
plus adéquates au pied du client pour produire une chaussure.

Dans ce contexte, un moteur de recherche d'objets 3D avec une bonne
performance en résultats et en temps d'exécution devient une nécessité.
Quelques moteurs de recherche 3D sont disponibles en ligne tels que *3D
model Search Engine* proposée par le groupe *the Princeton Shape and Anal-
ysis*[5] presenté dans la figure 1.2, *the Semantic3D project search engine*[6], *the
Taiwan University search engine*[7] et le moteur de recherche des modèles 3D
du Dejan Vranic[8].

Ces moteurs de recherche de modèles 3D sont encore limités en perfor-
mance. La plupart des approches proposées concernant leur contenu sont
basées sur les histogrammes statistiques qui mesurent certaines caractéris-
tiques géométriques des objets 3D. Les diverses mesures de ces caractéris-
tiques sont calculées à partir de maillages 3D des surfaces triangulaires
(voir figure 1.3). La représentation en maillage 3D est la représentation la
plus utilisée.

Dans la littérature, il existe différents types d'approches pour
l'indexation des modèles 3D. Les plus anciennes méthodes sont inspirées
des méthodes 2D comme les descripteurs de Fourier (30), les calculs des
moments (14), l'extraction de la ligne médiane, la transformation angu-
laire radiale (89), etc. La littérature concernant les descripteurs d'objets 3D
est très riche. Les méthodes basées sur une description globale d'objets
qui caractérise la forme des objets 3D d'une manière grossière sont sou-

---

[5] `http://shape.cs.princeton.edu/search.html`.

[6] `http://www-rech.telecom-lille1.eu/3dretrieval/`.

[7]`http://3d.csie.ntu.edu.tw/`.

[8]`http://merkur01.inf.uni-konstanz.de/CCCC/`.

vent efficaces pour discriminer des formes simples mais insuffisantes pour des formes plus complexes, contrairement aux méthodes basées sur des descriptions locales.

Les solutions existantes pour l'indexation d'objets 3D sont assez robustes à l'égard des transformations rigides comme la translation, la rotation ou le changement même de facteur d'échelle. Cependant, la figure 1.4 présente des formes 3D qui souffrent de la forte variabilité par rapport à des transformations (comme la transformation isométrique ou affine), la présence de dégradations (comme la variation de bruit, d'échantillonnage ou de mise à l'échelle) ou la représentation partielle (comme le modèle avec des pièces manquantes ou combiné avec d'autres). En conséquence, les méthodes existantes conduisent à des descripteurs différents pour la même forme sous ces transformations qui affectent les résultats et les performances du moteur de recherche 3D.

L'objectif de cette thèse est de présenter une approche pour l'indexation des modèles 3D qui est invariante aux transformations rigides et non rigides et également robuste à l'indexation partielle de modèles 3D.

Contributions

Dans cette thèse, comme point de départ, nous sommes persuadés que les descripteurs globaux ne sont pas très efficaces pour l'indexation partielle de modèles 3D. Nous proposons une signature basée sur un descripteur local. Nous décomposons les modèles 3D en sous-parties; chaque partie est affectée à un descripteur local. Le descripteur doit décrire la forme de chaque partie. Nous avons choisi de décrire chaque partie par des courbes fermées qui encodent la forme de cette partie. Pour décomposer les modèles 3D similaires en partie de la même façon, même si ces pièces n'ont pas de signification sémantique, nous avons choisi de définir une fonction scalaire sur la surface et pour chaque valeur de cette dernière nous détectons une région.

Nous commençons nos recherches dans les fonctions scalaires définies sur des surfaces, qui respectent les propriétés principales comme

l'invariance aux transformations rigides et non rigides, l'insensibilité au bruit, la robustesse aux changements de topologie, et l'indépendance des paramètres. La définition d'une telle fonction reste une question ouverte. Dans cette thèse, nous définissons une fonction appropriée sur la surface. Nous démontrons l'efficacité de cette fonction en la transformant à une fonction de Morse, pour calculer le graphe de Reeb d'un modèle 3D. En outre, nous l'avons testée sur différents modèles 3D, y compris des modèles 3D partiels et combinés.

Nous continuons par l'introduction d'une nouvelle méthode pour l'indexation des modèles 3D basée sur des courbes de niveaux fermées en utilisant notre fonction scalaire. Nous générons et analysons les courbes de niveaux fermées à partir d'un point source (le point source est détecté automatiquement par notre fonction scalaire) du modèle 3D en utilisant notre fonction scalaire. Pour chaque valeur de la fonction, nous détectons une région, puis nous décrivons la forme de cette région par une courbe fermée. Enfin, nous comparons les modèles 3D en analysant la forme de leurs courbes de niveau correspondant. Nous avons testé notre approche sur deux bases de données. Nous concluons que notre méthode à de bons résultats, mais n'est pas très efficace pour l'indexation partielle de modèles 3D.

Enfin, nous présentons une nouvelle méthode, qui à son tour est une amélioration de notre approche mentionnée ci-dessus à l'aide de la technique *sacs de mots*. Ensuite, nous démontrons l'efficacité de notre approche sur deux séries d'expériences. La première série d'expériences évalue notre approche sur l'indexation de modèles 3D et la deuxième série d'expériences valide notre approche sur l'indexation partielle de modèles 3D.

Plan

Le reste de ce manuscrit est illustré dans la figure 1.5 et est exposé comme suit.

Le chapitre 2 présente l'hypothèse que la mesure de similarité entre deux objets 3D est réduite à un calcul de distance entre leurs deux de-

scripteurs. Ensuite, nous procédons par l'introduction des solutions existantes pour l'indexation des modèles 3D et pour l'indexation partielle de modèles 3D.

Dans le chapitre 3, nous présentons une approche pour définir notre fonction d'application qui à son tour préserve les propriétés d'invariance (voir figure 1.5a).

Dans le chapitre 4, nous utilisons notre fonction d'application pour calculer les graphes de Reeb et prouvons les propriétés d'invariance de notre fonction (voir figure 1.5b).

Dans le chapitre 5, nous présentons une nouvelle approche pour l'indexation des modèles 3D basée sur des courbes de niveaux fermées dans $\Re^3$. (voir figure 1.5c).

Le chapitre 6 est une amélioration de notre approche du chapitre précédent en utilisant la technique *sacs de mots* pour l'indexation partielle de modèles 3D (voir figure 1.5d).

Enfin, nous concluons ce manuscrit en résumant les contributions de cette thèse, et en énumérant les problèmes qui restent ouverts et proposons des orientations pour de futures recherches.

State-of-the-art of 3D-models retrieval

2

Tʜɪs chapter introduces the assumption that the measure of similarity between two 3D-objects can be reduced to a distance computing between their two descriptors discussed in section 2.1. Then we proceed by introducing three sections. First section 2.2, is about reviewing the state-of-the-art of three dimensional object retrieval using the 3D based methods. The second section 2.4, is about the state-of-the-art of three dimensional object retrieval using the 2D based methods. The last section 2.5, is about the state-of-the-art of partial three dimensional object retrieval. Recently, researchers have investigated a lot of problems about 3D-object retrieval. Also, they have done an extensive amount of literature. We will give a brief discussion about the advantages and drawbacks of different techniques. Before concluding with a comparison table of all or almost all existing methods in the literature presented in section 2.6.

Iɴᴛʀᴏᴅᴜᴄᴛɪᴏɴ (ᴇɴ ғʀᴀɴçᴀɪs)

Ce chapitre présente l'hypothèse que la mesure de similarité entre deux objets 3D est réduite à un calcul de distance entre leurs deux descripteurs discutée dans la section 2.1. Ensuite, nous procédons par l'introduction de trois sections. La première section 2.2 présente l'état de l'art de la technologie récente en indexation des modèles 3D baseés sur les méthodes 3D. La deuxième section 2.4 présente l'état de l'art de la technologie récente en indexation des modèles 3D baseés sur les méthodes 2D. La dernière section 2.5 présente l'état de l'art de la technologie récente en indexation par similarité partielle de formes 3D.

Récemment, les chercheurs ont fait des nombreuses expériences concernant les problèmes liés à l'indexation des modèles 3D. En outre, ils ont introduit plusieurs solutions. Nous allons présenter une brève discussion sur les avantages et les inconvénients des différentes techniques existantes dans la littérature. Enfin, on conclut par un tableau de comparaison des méthodes existantes.

2.1 SHAPE MATCHING CONCEPTS

3D-object retrieval based on content, requires having an automatic method for measuring the similarity between two objects.

Generally, this is done based on the assumption that the measure of similarity between two 3D-objects can be reduced to a distance computed between them where small distance means small dissimilarity (large similarity) and large distance means large dissimilarity (small similarity). The definition of such distance (similarity distance) is given by:

$$d(s_i, s_j) : S \times S \to \Re^+ \qquad (2.1)$$

Where $s_i, s_j \in S$ and $S = \{s_1, s_2, ..., s_i, ..s_N\}$ is a set of shapes. This distance function $d(s_i, s_j)$ is called a *metric* if it respects the following properties:

- **Identity** means same shapes totally match if $\forall s_i \in S, d(s_i, s_i) = 0$.

- **Positivity** property ensures that two different shapes never match completely: if $\forall s_i, s_j \in S, d(s_i, s_j) > 0$.

- **Symmetry**: if $\forall s_i, s_j \in S, d(s_i, s_j) = d(s_j, s_i)$

- **Triangle Inequality**: if $\forall s_i, s_j, s_k \in S, d(s_i, s_j)(s_i, s_k) + d(s_k, s_j)$

The distance function $d(s_i, s_j)$ is called a **transformation invariant metric** if $d(s_i, s_j)$ is a metric and respect the **transformation invariance** (scaling, rotation or translation). Given a transformation group G, if $\forall s_i, s_j \in S, \forall g \in G, d(s_i, g(s_j)) = d(s_i, s_j)$.

The similarity distance of two shapes $D(s_i, s_j) = min(d(s_i, g(s_j)))$ is not an efficient similarity measure for 3D shapes since, the representation forms of 3D shapes are not well suited for matching. To define a suitable similarity distance we have to deal with two main problems the pose normalization (any combination of rigid transformation such as scaling, rotation and translation), and the feature invariance that satisfies several requirements for invariance. A typical set of requirements includes invariance to similarity transformations, invariance to shape representations, invariance to geometrical and topological noise, and invariance to articulation or global deformation.
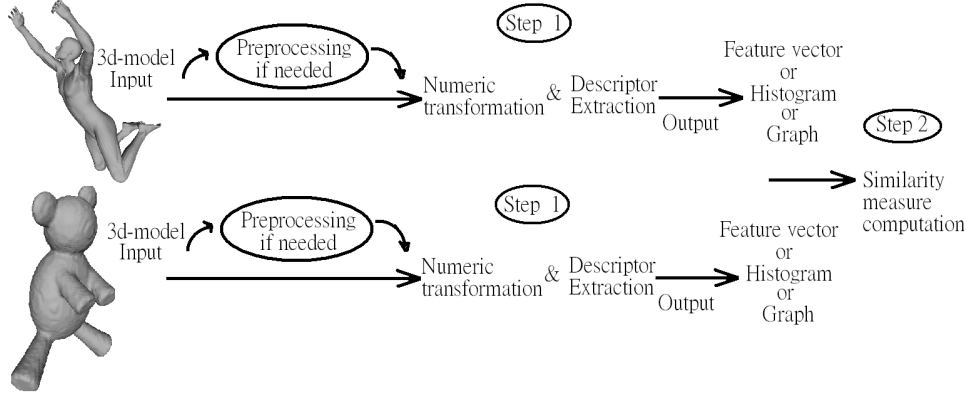
Figure 2.1 – *Two main steps for 3D-objects comparison and a preprocessing step if needed for rotation, translation and/or scaling invariance.*

- to deal with the pose normalization, let G be a transformation group, $\forall s_i, s_j \in S, \forall g_1, g_2 \in G, d(s_i, s_j) = d(g_1(s_i,)g_2(s_j))$

- to deal with the feature invariants, a shape is described by the invariants features under a function f that respects the $d(s_i, s_j) \simeq d(f(s_i), f(s_j))$ this function is called feature extraction function.

For 3D model matching, we need to define a shape descriptor or shape signature that captures the significant features of the shapes. Let q a 3D shape query and let $S = \{s_1, s_2, ..., s_i, ...s_N\}$ a 3D shape database. To retrieve 3D models that are similar to q we need to retrieve all the shapes from S where the similarity distance D is lower than a given threshold t: $D(q, s_i) < t$ or we need to retrieve the first k shapes where $D(q, s_i)$ are minimum.

As we noticed a comparison between two 3D-objects as presented in figure 2.1 involves two main steps the signature extraction (or object description) and similarity measure. Also most approaches involve a prepossessing step if needed by the descriptor, to handle the invariance to rotation, translation, and/or scaling.

2.2   3D-MODEL RETRIEVAL STATE OF THE ART

Several 3D-object retrieval approaches and shape descriptors have been introduced in the literature (82, 65, 105) and some of them are implemented

in 3D search engines that are available on-line (3D model Search Engine) proposed by the Princeton Shape and Analysis Group., the Taiwan university search engine, etc. These approaches lead to a number of different categorizations "Geometrical techniques, structural and topological technique, local methods, global methods, view based methods, etc".

I use a specific categorization since there are some approaches that use a combination of methods and can be classified into more than one category. I prefer to divide these methods into two main categories 3D based methods and 2D based methods.

2.3 3D based methods for 3D-model retrieval

3D based methods for 3D-model retrieval involve all methods that took into consideration the 3D model as itself to retrieve information and define the descriptor. We can distinguish five families of approaches that are used to describe 3D-objects: Global based methods, methods based on spectral embeddings, methods based on extended Gaussian images, Local based methods and graph based methods.

2.3.1 Global based methods

The global based methods present approaches where the descriptor characterizes the whole 3D-model. All these methods or most of them do not describe the details of the 3D-model that leads to affect their results. This section introduces the main existing global methods and their respective advantages and drawbacks.

The cord histogram

In 1997, Paquet and Rioux (78) released their first descriptor named the *cord descriptor*. The authors used a database that consists of a large number of objects digitized with 3D laser scanners which they have developed. Using their scanners, the authors can simultaneously acquire the shape and the colour of a 3D-object. The proposed indexing approach is based on some statistics on the cords of the object to be indexed. The cords are
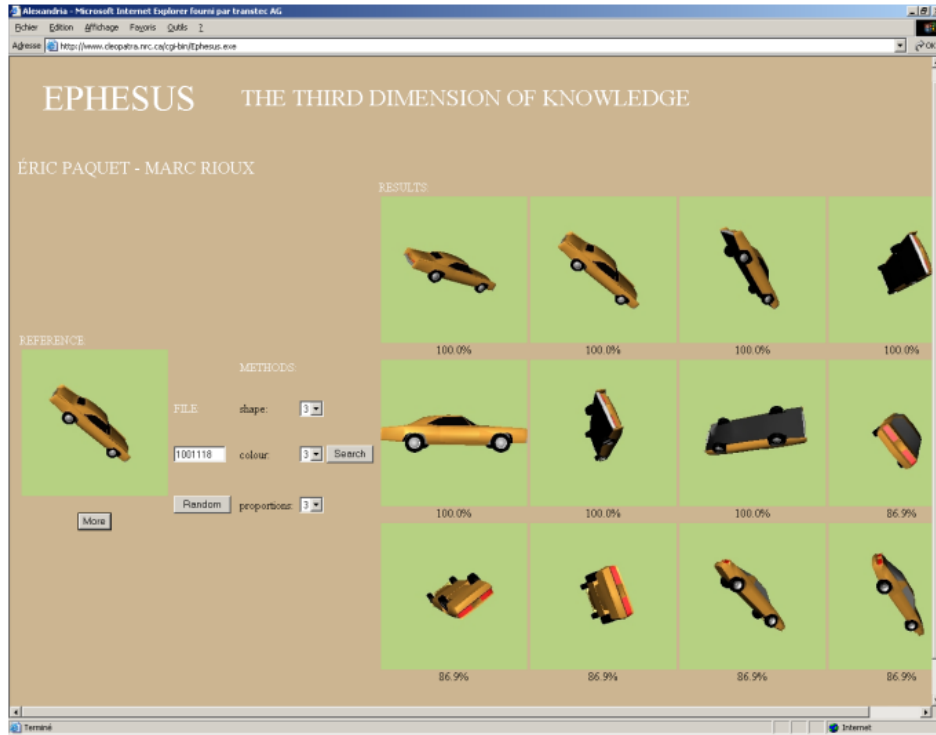
Figure 2.2 – *3D search engine of the CNRC Ottawa.*

defined here as the segments connecting the centre of gravity of the object
and the centres of each triangle of the mesh. The author proposed three
cord histograms that can be constructed:

- a histogram of the cord lengths;

- a histogram of the angles between the cords and the first principal
  axis;

- a histogram of the angles between the cords and the second principal
  axis.

This technique is presented in the Nefertiti project which is implemented
in the first 3D-model search engine. This search engine needs a prepro-
cessing algorithm for data normalization to handle isometric transforma-
tions and it is not good enough due to the fact that local shape features
are not well described neither robust toward perturbations. The time con-
suming process for execution is very good toward other techniques.

Area and volume method

Zhang and Chen(120) extracted feature vector based on the area and the volume of the mesh. The authors Compute the surface or the volume for each element of the shape (triangles for meshes and voxels for volumetric), then add up all the values for the mesh. This method is not categorized for precise searching and details detecting. In my point of view, this method can be used as a preprocessing stage for fast pruning data and accelerates the query due to the fast computation of an area and a volume of an object.

3D Hough descriptor

The optimized 3D Hough Transform Descriptor presented by Zaharia and Prêteux (119) is intrinsically invariant to connectivity representation issues, but not to geometric transformation. The descriptor of this method is based upon the principle of accumulating the representative parameters from triangulated faces of the object in spherical representation (figure 2.3). For the choice of the weights, the authors consider the relative area of the faces combined by their normal vectors.
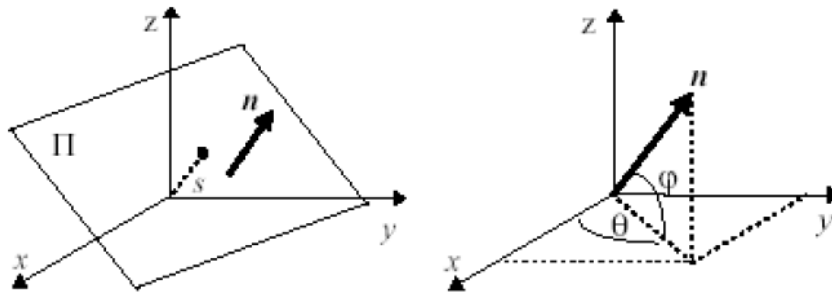
Figure 2.3 – *Switching to spherical coordinates for the Hough transform. The plan is described by 3 parameters.*

Shape distribution descriptor D2

The most famous descriptor of the "Shape Retrieval and Analysis" research group, of Princeton University, USA, is the shape distribution descriptor **D2** (76). The shape distribution represents a probability distribution of a

shape function measuring global geometric properties of the object. The shape descriptor of a 3D object is given by a probability distribution that counts the occurrence of Euclidean distances between pairs of points randomly chosen on the surface of the object. The authors use histograms
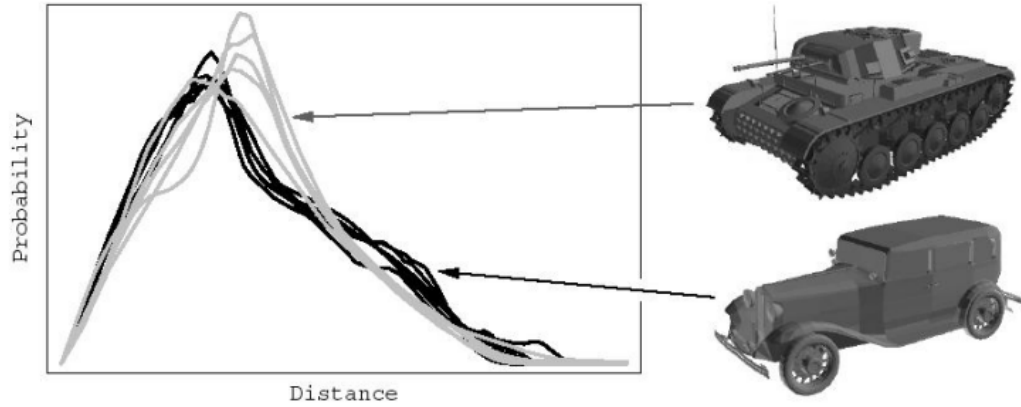
Figure 2.4 – *Shape distribution of different vehicles. Figure taken from Osada et al.(76).*

containing 64 bins to represent their descriptor (see figure 2.4), and the Minkowski norm to calculate the distances between histograms. The proposed approach is invariant to rigid transformations and is robust to deformations of the mesh connectivity.

The shape distribution proposed by Princeton University is a probabilistic method, whose main advantages are the ease of implementation, computing time, the invariance to geometric transformations and the robustness with respect to the mesh noise (connectivity, decimation). The descriptor characterizes the global shape of objects but not the details. This approach can be applied to classify models. Since it fails to capture the details of a shape, therefore it fails to discriminate among locally dissimilar shapes.

2.3.2   Methods based on spectral embeddings

In this paragraph, we describe recent works based on spectral embeddings which provide approaches for 3D-shape analysis like correspondence analysis (47, 15), for clustering (72, 121) and especially for non-rigid 3D-model retrieval and comparison (64, 87, 12).

These works study the differential properties of a linear operator defined on the mesh. Despite descriptors based on moments or multidimensional scaling using geodesic distance presented by Elad and Kimmel (29), we focused on 3D-model retrieval methods based on Laplace-Beltrami operator due to their good results for non-rigid shape retrieval.

The discretization of Laplace-Beltrami operator on the mesh leads to an eigensystem and the graph eigenvalues studies, that have interesting insight in many areas of mathematics, particularly the interaction between the spectral Reimannian geometry and spectral graph theory.

Sorkine (98) presents geometry processing that are related to the Laplacian processing. Levy (59) explains how to compute an approximation of the eigenfunctions of a differential operator. Belkin and Niyogi (4) present shape embedding in a high dimensional space.

Reuter *et al.* (85, 86) improve the discretization to a higher level of Laplace-Beltrami operator by adding extra nodes on each triangle in the mesh. They use the spectra given by the first n eigenvalues as isometric invariance descriptor since the eigenvalues properties are related almost to all issues of invariants and transformations.

Mahmoudi and Sapiro (64) compare histogram of pairwise using the diffusion distances between all vertices on the mesh. The descriptor ignores important local information since it is defined to describe the whole shape. The authors enhance their descriptor to local recognition by considering a graph connecting a number of patches. But this descriptor still depends on a time parameter to detect the level of details.
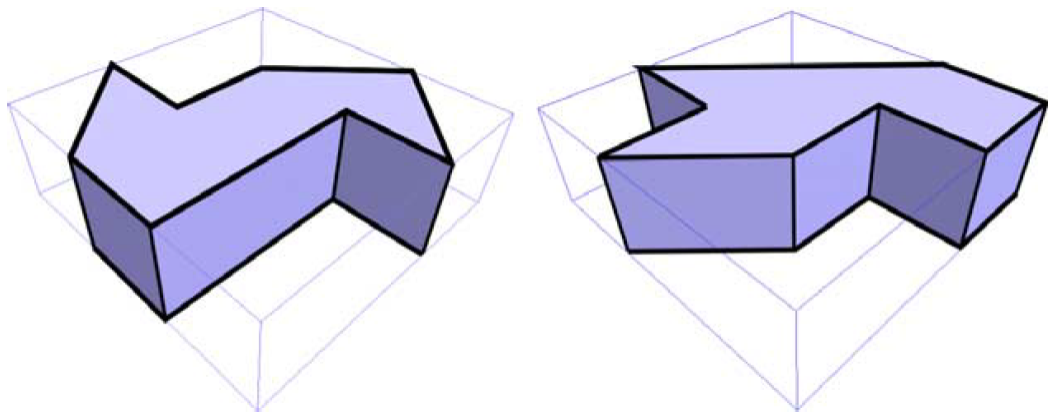
Figure 2.5 – *Isospectral shapes. Figure taken from Reuter et.al. (87).*

Reuter *et al.*(87) and Marini *et al.* (66) describe the shape by the spectra (eigenvalues) of the Laplace-Beltrami operator. Reuter *et al.*(87) got good results even though two different shapes may have the same spectrum then the same descriptor (see figure 2.5). The authors assume that this situation is rare and each shape has its own spectrum and similar shapes have similar spectrum. Then the authors prove that isospectral shapes but not isometric models can be discriminated by the spectrum of the boundary surface. Marini *et al.* (66) investigate if the selection of a particular spectrum sequence is the best choice or there exists other sequences of eigenvalues that provide better results. In my point of view, the computation of the spectrum depends on the mesh density. For both methods to be very accurate in the computation of the spectrum, they need to use dense meshes that affect the time consuming for their methods.

Rustamov (91) creates a descriptor vector from the evaluated eigenfunctions of Laplace-Beltrami operator. The author computes histograms that capture the distribution of distances between points based on the Green function (the dot product of two d-dimensional vectors defined from the eigenfunctions and the eigenvalues of Laplace-Beltrami operator) for uniformly sampled points on the mesh. Since the author uses the cotangent weight to discretize laplace-Beltrami operator, this method does not handle degenerated meshes also he didn't propose boundary condition for surfaces with boundary.

Bronstein *et al.* (11) and Bronstein and Kokkinos (12) compute the remaining heat on each vertex after a scale time t. For scale invariance, they improve the heat kernel signature (HKS) to scale-invariant heat kernel signature (SI_HKS) by scaling and shifting using a logarithmic scale-space based on Fourier transform shown in figure 2.6 where the curves remain almost unchanged after scaling the object. For the discrete computation of the heat kernels, they used the cotangent weight approximation of the Laplace-Beltrami operator with the 200 smallest eigenvalues. Marini *et al.* (66) investigate in the best particular spectrum sequence and using the first 200 eigenvalues with cotangent weight is not the best choice for original models.
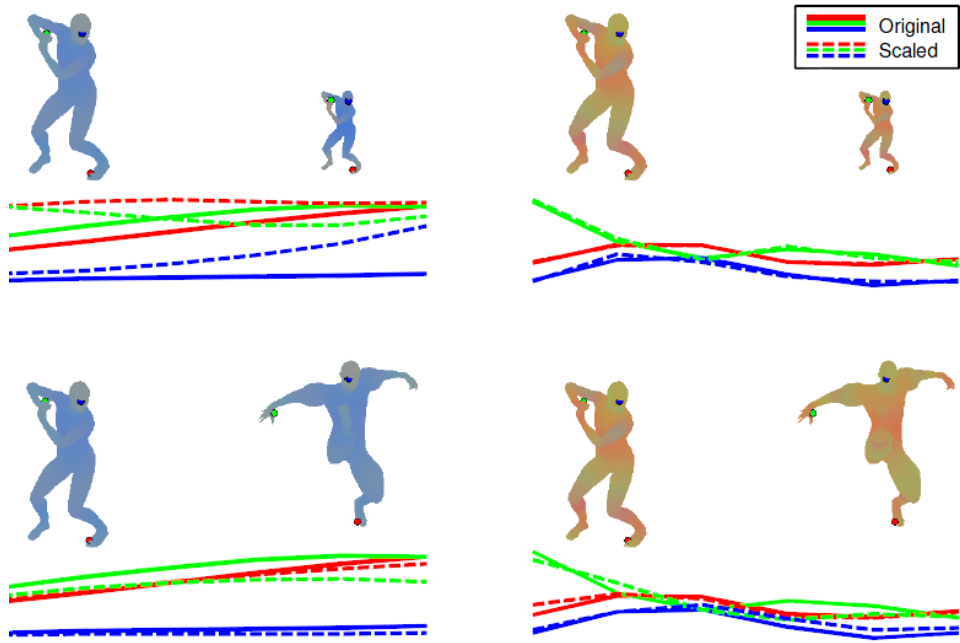
Figure 2.6 – *The improvement of the HKS (left) and the SI_HKS(right) where the curves remain almost unchanged after scaling the object. Figure taken from Bronstein and Kokkinos (12).*

2.3.3 Methods based on Extended Gaussian Images

The Extended Gaussian Images (EGI) descriptor consists to map a function that synthesizes some information concerning the 3D-mesh on a Gauss sphere partitioned into several facets. Each triangle contributes to its corresponding facet (the facet given by the direction of its normal) by a weight equal to the area of the triangle (5).

There exists a variant of EGI, the Complex Extended Gaussian Images (CEGI) gives as contributions to the facets, for each triangle, a complex number whose magnitude is equal to the area of the triangle and whose phase is equal to the distance from the center of the triangle to the center of the sphere (50).

This representation allows to discriminate primitive shapes. In addition, it also allows to obtain many useful information as the symmetry properties or the length of the cords. The disadvantages of the EGI are its dependency on the connectivity of the mesh, the over representation of information at the poles due to the discretization in spherical coordinates, the non-invariance to some geometric transformations, and also the fact

that the method is poorly suited to objects that are not homeomorphic to a sphere.

2.3.4   Local based methods

The local based methods present approaches where 3D-objects can be characterized by attributes computed on local surfaces or patches of the object associated with local descriptors. To select local features of a given object some authors use an object segmentation method. Other authors use a sampling method and select patches according to some geometric criterion, following successful approaches in 2D image recognition like SIFT (63). More recently, some authors propose the use of feature points extraction algorithm for detecting points of interest around which they extract patches.

A local descriptor reflects the local geometric characteristics of a 3D-object, unlike global methods which tend to describe globally the shape of the 3D-objects.

Local descriptor based on closed curves

Lmaati *et al.*(62) reconstruct 3D closed curves and extract feature vector as a descriptor. The feature vector is a combination of the area and the dot product descriptors that describe the reconstructed 3D closed curve in order to define the 3D curve analysis. This method needs to align the model into canonical position before the construction of the closed curves.

Tabia *et al.* (103) detect feature points located at the extremities of a 3D model based on the geodesic distance presented by Tierny *et al.* (111). For each feature point, they generate a collection of closed curves based on the geodesic distance. Each feature point and its collection of closed curves represent a part of the model (see figure 2.7). Finally, they use the belief functions to define the global distance between 3D-models. This method is very sensitive to topology and a small variation of the feature point leads to a large variation in curves.
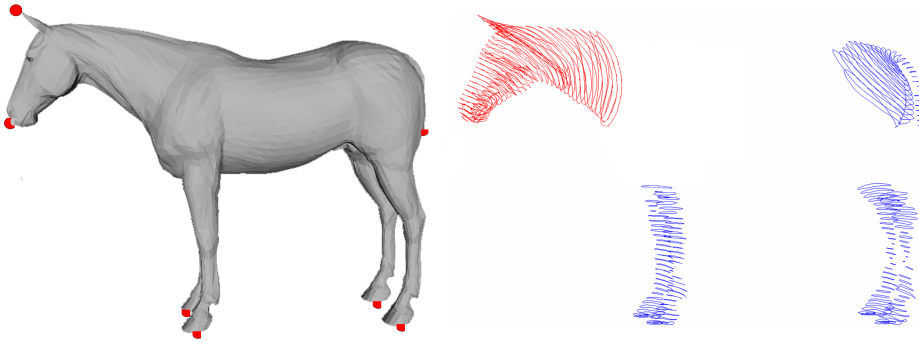
Figure 2.7 – Representation of patches by an indexed collection of curves.

Local descriptors based on spherical harmonics

Saupe and Vranic (25) proposed to apply a Fourier transformation on the sphere S_2 by applying the spherical harmonics formulas proposed by Healy *et al.* (43). Then, to overcome the problem of invariance to rotations, Kazhdan and Funkhouser (52) proposed to implement the decomposition in spherical harmonic functions defined by the intersection of the surface of the 3D object with a set of concentric spheres (see figure 2.8).
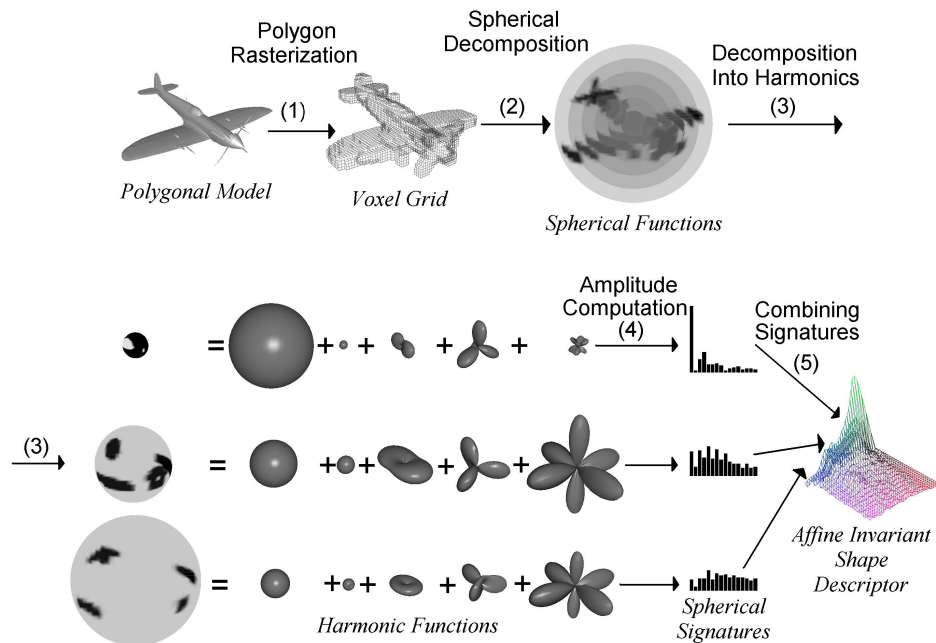
Figure 2.8 – Computing the Harmonic Shape Representation.(Figure taken from Kazhdan and Funkhouser 2002 ACM Press (52)).

The authors prove that the spherical harmonics method gives better

results than their previous descriptor (D2 shape distribution). However, it is based on a voxelization of 3D models and therefore depends on the level of resolution of the voxelization, resulting in a loss of details in the description of the object.

Vranic (115) proposed to apply the method directly on 3D-meshes with new 3D spherical functions. The results he obtained on his database with his method are superior to those based on voxel model, and less time consuming. However, these results also show that the encoded information does not really allow accurate querying on the shapes. The drawbacks of this method is the preprocessing step (like Principals Component Analysis) needed for data normalization and finding the canonical position and orientation of a model to insure the invariance requirement for the 3D-shape descriptors. Also the main limitation is the number of concentric spheres and the number of harmonic coefficients which remain, may be too low. The author chooses in practice 32 and 16 concentric spheres by harmonic spheres, or a descriptor of 32 * 16 = 512 coefficients.

Local methods based on spectral descriptors

Sun *et al.* (100) present a multi-scale local descriptor, the Heat Kernel Signature (HKS) using the eigendecomposition of the Laplace Beltrami operator. They compute the amount of heat that remains at point x after time t. For each point x on the shape, its Heat Kernel Signature is an n-dimensional descriptor vector of the form

$$p(x) = c(x)(K_{t_1}(x,x), ..., K_{t_n}(x,x)),$$

where $c(x)$ is chosen in such a way that $\|p(x)\|_2 = 1$. The HKS is informative, multi-scale that captures both local features and global shape structure and thus isometry-invariant (two isometric shapes have equal HKS) (see figure 2.9).

Recently another method also based on an eigendecomposition of the Laplace-Beltrami operator presented by Ruggeri *et al.* (90). The authors detect critical points of the eigenfunctions related to the smaller eigenvalues of the Laplace-Beltrami operator. Then, these points are origin of
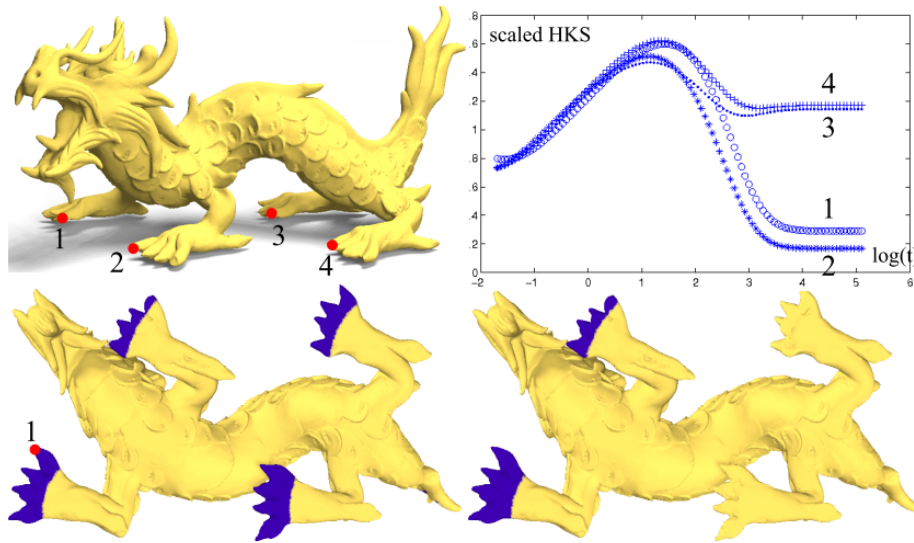
Figure 2.9 – *Top left: dragon model; top right: scaled HKS at points 1, 2, 3 and 4. All four signatures are close at small t's while big t's separate the points on the front claws from those on back; bottom left: the points (blue), whose signature is close to the signature of point 1 based on the smaller half of the t's; bottom right: based on the entire range of t's.(Figure taken from Sun et al.(100)).*

a smartly sampling technique using statistical criteria for controlling the density and number of reference points. For each reference point they associate a local descriptor consisting of the geodesic shape distribution around the point. This method depends on a threshold to control the density of the reference points and also depends on the feature point detected at first by the eigendecomposition of the Laplace-Beltrami operator which, to our experience with this method, generates undesirable feature points or misplaces them.

Another method proposed by Lavoué (57) based on the spectral coefficients amplitude that are computed using the eigenvectors of the Laplace-Beltrami operator. The author selects a set of random point uniformly sampled on the surface. Then, each point is associated with a local patch by considering the connected set of facets belonging to a given sphere. His descriptor is computed by projecting the geometry of each patch onto the eigenvectors of the Laplace-Beltrami operator.

2.3.5   Graph based methods

The graph based methods describe the topology structure of the shape by
linking the meaningful components of a 3D-object. In contrast to methods
discussed in previous paragraphs, which consider mostly the geometry of
the shape. Despite approaches based on model graph that are applicable
for 3D solid models produced by CAD system, we discussed two families
of approaches that are used in the literature: methods based on Reeb
graph and methods based on skeleton.

Methods based on Reeb graph

Reeb graphs are symbolic representation of a certain subset of Morse func-
tions. Their constructions are based on Morse theory (95) that character-
izes the topology of closed surfaces, we find as many cycles in graphs
as holes in objects (see figure 2.10). Reeb graphs have been introduced
in 1964 by Georges Reeb (84). Recently it has been in use for 3D-shape
applications. Resulting an expressive topology description and forming
a high level skeletal representation of the surface. The Reeb graph is de-
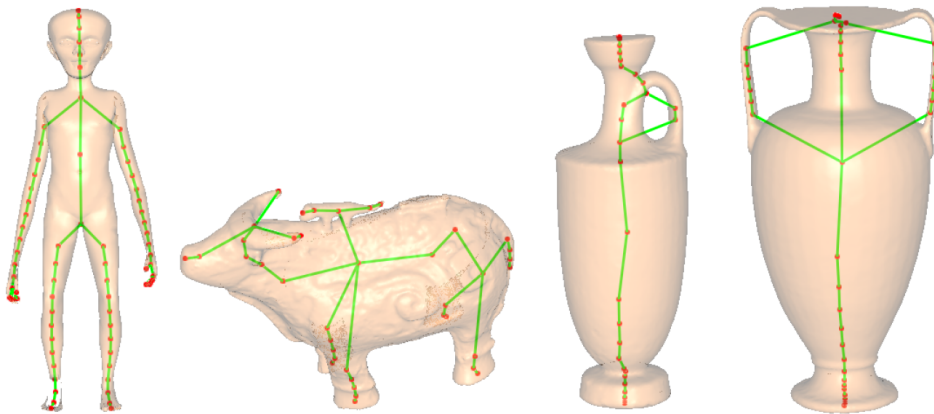
Figure 2.10 – *There are as many cycles in the graphs as holes in the objects. (Figure
taken from Tung (113)).*

fined as the quotient space of a shape and a quotient function. Tierny *et
al.* (111) define their quotient function to generate a set of critical points
based on a geodesic distance. This function computes for each vertex the
geodesic distance to the closest feature point (points located on the ex-
tremities of the mesh). Hilaga *et al.* (67) define quotient function by an

integral geodesic distance and introduced the multiresolution Reeb graph which allowed two kind of comparison of 3D-object, with low levels of resolutions to get quick results, or using a maximum resolution to obtain finer results. The multiresolution graph was expanded by Tung (113).

Biasotti *et al.* (7) compare Reeb graphs obtained by using different quotient functions and highlight how the choice of these functions determines the final matching result. For instance, the integral geodesic distance as quotient function is especially suited for articulated objects, while the distance to the barycenter should be preferred if the aim is to distinguish different poses of an articulated object.

To summarize, quotient functions defined using the geodesic distance are suited for matching articulated 3D objects, but they are sensitive to topological changes. Also, they cannot be applied to arbitrary meshes.

Methods based on skeleton graph

The skeleton graph is a modern variant of shape skeletons inspired by the medial axis defined by Blum (9) in 1967 and has shown a great potentiality in 3D-model. The corresponding medial axis for 3D shapes is the medial surface (40), because in addition to curves, it can also contain surface patches. However, 1-dimensional representations are reported to be more convenient from an applicative point of view (1). Consequently, most authors consider a pruned version of the medial surface, as shown in figure 2.11.

We can found different techniques for skeleton graph extraction. Wang *et al.*(116) present an approach using volume thinning. The curve skeleton extracted can be used to describe the geometry and the topology of the 3D-object. Another technique proposed by Oscar *et al.*(1) who use Laplacian contraction. For a comprehensive survey on skeleton graph extraction we refer the reader to Nicu *et al.* (21).

In 3D-model retrieval domain, Sundar *et al.* (101) use a skeletal graph as a shape descriptor. The authors obtain a hierarchical graph structure using different thinness parameters of a thinning algorithm developed by Gavgani and Deborah (36). The skeletal points are connected in an undi-
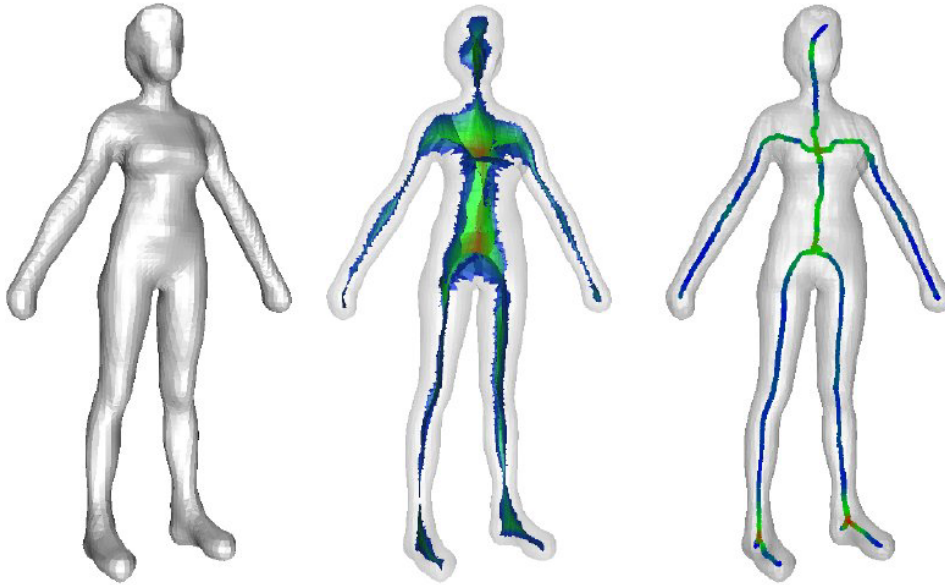
Figure 2.11 – *A surface mesh, its medial surface and its graph skeleton. (Figure taken from Tamal et al.(23)).*

rected acyclic shape graph by applying the Minimum Spanning Tree algorithm. For each nodes of the graph, the authors assign two vectors encoding the radial distribution about the segment (geometrical information), and the topology of the subtrees rooted at the node (topological information). Generally, these methods using thinning algorithm are sensitive to topology change and the process is quite time consuming.

2.4   2D based methods for 3D-model retrieval

The idea of using 2D-views to index 3D-models is based on the assumption that two 3D models are similar, if they look similar from the same viewpoints (angles of view), therefore a number of views (2D projections) of objects could be used to represent the shape of the objects. Many authors proposed 3D-model indexing methods based on 2D views. In this section we will discuss some of these techniques.

2.4.1   Static 3D-model retrieval from 2D views

The fact that all 2D views representing a 3D-model do not contains the same importance as information as shown in figure 2.13. For this issue,

Chen *et al.*(16) proposed to describe the 3D-model by one hundred orthogonal projection excluding the symmetry projection to define different 2D views. Then, each view is encoded as feature for 3D-model retrieval by Zernike moments and Fourier descriptor. Similar to 2D-views, the silhouettes are composed of the shape boundaries from one view point. In order to represent a 3D shape, a set of silhouettes is extracted, from which, a set of descriptors are computed and stored. Silhouettes can be seen as a more economical representation compared to model based representations. This representation is commonly used in object classification task where matching is done between one silhouette of a 3D shape and a database of objects represented as set of silhouettes of models (see figure 2.12).

Heczko *et al.* (44) characterize 3D-objects in terms of their silhouettes. Silhouettes are obtained by parallel projections onto three planes. The descriptor is obtained by concatenating Fourier descriptors of the three resulting contours. Another descriptor based on image with the same context as the silhouette called *depth buffer* but instead of three silhouettes, it renders using parallel projection six grey-scale image. The experiments of Bustos *et al.* (13) conclude that this method is able to outperform other descriptors on Benchmark database. However, the problem with this kind of descriptors is that, in theory, different 3D shapes might have the same set of silhouette images.
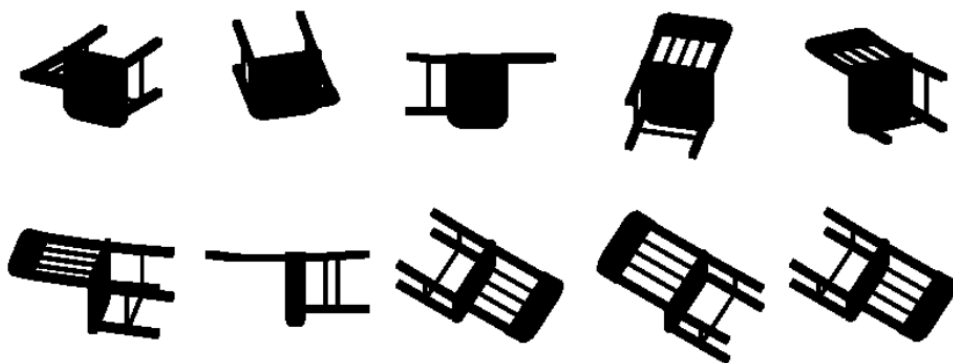
Figure 2.12 – *A typical example of the 10 silhouettes for a 3D model. (Figure taken from Ding-Yun(16)).*
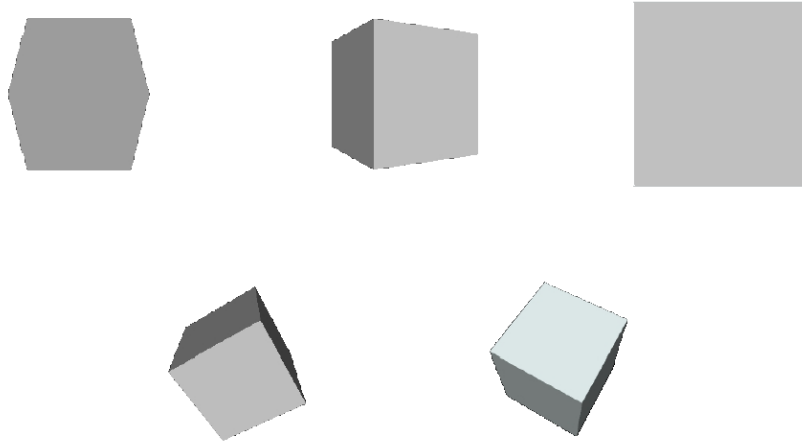
Figure 2.13 – *Characteristic views of a cube. (Figure taken from Filali et al.(31)).*

2.4.2 Dynamic 3D-model retrieval from 2D views

Another method proposed by Filali *et al.*(31) which provides an optimal selection of 2D views from a 3D-model. Initials views are taken from a camera placed on each face of the 320 faceted polyhedron, after scaling and translating the 3D-model to its barycenter coincides with the center of the polyhedron. Each view is represented by 49 coefficients of Zernike moments. The selection of the characteristic views that best characterize the 3D-model is an approach based on a method derived from K-means. Then the Bayesian information decides if new characteristic views better fits the data or not. Figure 2.14 presents a snapshot of the 3D-search engine provided by the authors. Results show that their approach have better results except for Light Field Descriptor (94) which consume much more time for execution either a lot of memory since the Light Field Descriptor is based on 100 characteristic view and two descriptors (Zernike moments and Fourier transform). The main drawback of these methods is a loss of topology information.

2.5 PARTIAL 3D-MODEL RETRIEVAL STATE OF THE ART

Certain applications such as registration (45) or application where you can create new shapes by cutting sub parts and pasting on other shapes, need to index objects with regard to partial similarity. The partial 3D-
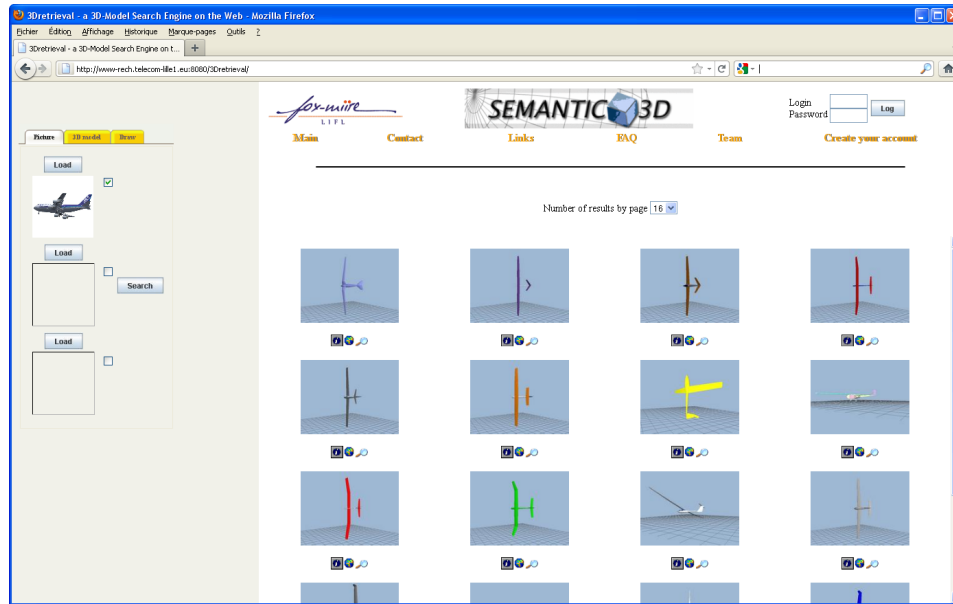
Figure 2.14 – *3D search engine of the MIIRE research team.*

model retrieval systems are expected to retrieve objects that have similar sub-shapes even if they visually differ globally and is therefore robust with respect to deformation. The partial 3D-model retrieval techniques are facing two main difficulties that still exist for 3D model retrieval systems: matching incomplete or partial models and matching combined models (see figure 2.15).

Descriptors used to globally describe a 3D-models cannot be efficient to describe partial or combined 3D-models. In the literature, we distinguish two main categories: local based methods and structural based methods. Also we distinguish an efficient technique for partial 3D-model retrieval based on the bag of feature.

2.5.1 Local based methods for partial 3D-model retrieval

Local based techniques aim to characterize the local properties by local descriptors on a large number of feature points or patches selected from the 3D-model. Then, similarity of partial 3D-model is estimated by feature point-to-point or patch to patch correspondence matching. Local based methods for 3D-model retrieval that are explored in section 2.3.4 can be applied for partial 3D-model retrieval but they are not evaluated on an existing partial 3D models database.

(a)   (b)
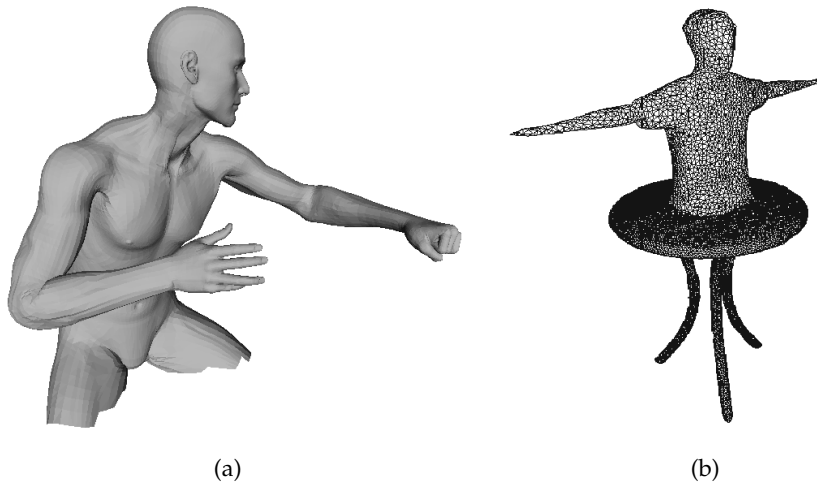
Figure 2.15 – *Representation of a partial 3D-model and a combined 3D-model.*

Despite local based methods for 3D-model retrieval discussed in section 2.3.4, Liu *et al.* (61) propose to use a Monte-Carlo sampling on the surface mesh and capture the local aspect of the shape with spin image signatures (48), their descriptor is not invariant under non-rigid transformations. Gal *et al.*(37) present an interesting geometrical hashing mechanism associated with a local surface description based on curvature analysis.

Funkhouser and Shilane (33) present a more sophisticated sampling strategy and then describe local geometry with three different descriptors based on Spherical Harmonics. Inspired by text document analysis, the authors use complex data structures for the matching derived from RANSAC. The drawbacks of this method is a preprocessing step needed for data normalization to find the canonical position and orientation of a model and to insure the invariance requirement for the 3D-shape descriptors. Also the main limitation is the number of concentric spheres and the number of harmonic coefficients which remain, may be too low.

Gal *et al.*(38) extend the shape context to the non-rigid setting, by using geodesic distances. However signatures based on geodesic distances are very sensitive to topological noise, since addition or removal of a small connection can change geodesic distances dramatically over a large portion of the shape.

Figure 2.16 – *Geometric parts extracted from a database of 3D models can be used to create new objects. The large brown chair was built from the circled parts of the others. (Figure taken from Funkhouser et al.(34)).*

Other methods based on local descriptors using spectral embedding like Sun *et al.* (100) introduced a multi-scale local descriptor, the authors restrict his studies to the temporal domain and compute their signature by observing the evolution of the heat diffusion over time. Recently, Dey *et al.*(24) detect feature points based on the heat kernel signature from spectral theory that shows this multi-scale property. Then, the authors filter these feature points by considering only the ones where the maxima of the heat kernel persist beyond a given threshold. This selection of t give a more robust set of feature points then they are integrated into a region matching algorithm.

Lavoué (57) considers a set of feature points uniformly sampled on the

mesh and associates local Fourier descriptors for each feature point. A visual dictionary is built by clustering a large set of feature descriptors, then each 3D-model is described by an histogram of occurrences of these visual words.

However, most of local descriptor methods base their partial similarity estimation on point-to-point matching only. This is particularly detrimental in term of re-usability in applicative contexts such as modeling-by-example (34) (see figure2.16) where the similar sub-parts have to be explicitly identified and extracted.

2.5.2 Structural based methods for partial 3D-model retrieval

Structural based techniques aim to segment the 3D-model to sub-parts in order to retrieve objects that have similar sub-shapes even if they visually differ globally. Then the 3D-model will be presented by a graph linking its sub-parts depicting the structural relations between them. These methods present the advantage to explicitly identify the surface patches that have been matched. Then, partial shape similarity is estimated using graph matching techniques.

Cornea *et al.* (20) propose to extract the skeletal representations then match the query skeleton against all other skeletons in the database. The skeleton representation is the curve-skeleton extracted using generalized potential field based on the method presented by Chuang *et al.* (17). Then the matching process is based on the Earth Mover's distance (53) to evaluate the partial similarity of the skeletons. The sub-parts descriptors are based on the Euclidean distance between the surface and its curve-skeleton which makes the method quite sensitive to isometric transformations.

Tierny *et al.* (109) present a structural approach for partial 3D shape retrieval, based on Reeb graphs. The authors enhanced the Reeb graph to compute their topological skeleton by analyzing the Gaussian curvature in each vertex of the mesh. The authors segment the topological skeleton obtained into patches and propose sub-part geometry signature based on parameterization techniques (Figure2.17). The Reeb graph is computed

using the geodesic distance which make the overall method very sensitive to topology change.
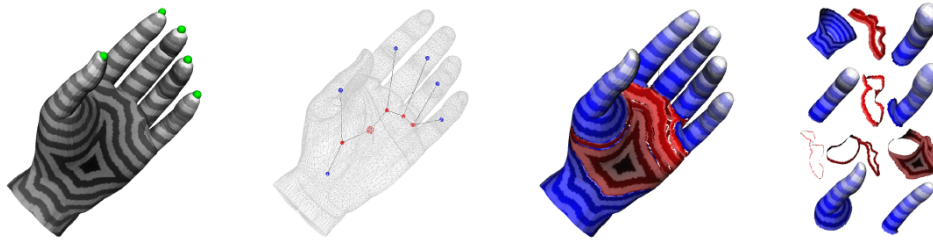
Figure 2.17 – *Segmentation of a hand triangulated surface model into its Reeb charts. (Figure taken from Tierny et al.(109)).*

Biasotti *et al.* (8) proposed an efficient method based on a derived version of multi-resolution Reeb graphs and Spherical Harmonics sub-parts signatures (spherical harmonics). Even if the Reeb graph computation is robust to isometric transformations, sub-parts signatures (spherical harmonics) are not, which is slightly detrimental to the overall robustness of the approach.

2.5.3 Bag-of-features technique

In 3D-model retrieval, the bag-of-features or bag-of-words technique is inspired by the approach used in the text retrieval. It is defined by treating the 3D-model features as words. All features of the 3D-model define the document. Then, a classification task is performed. Finally, a 3D-model is a vector of occurrence counts of words (see figure 6.1). The bag-of-feature technique is very useful for partial 3D-model retrieval, since this technique consider the max number of local features describing the model to construct a valuable document.

Liu *et al.*(61) presented a 3D-shape descriptor named "Shape Topics" and applied it to 3D partial shape retrieval. In their method, a 3D-object is considered as a word histogram obtained by vector quantifying Spin images of the object.

Ohbuchi *et al.*(74) introduced a view-based method using salient local features. They represented 3D-objects as word histograms derived from
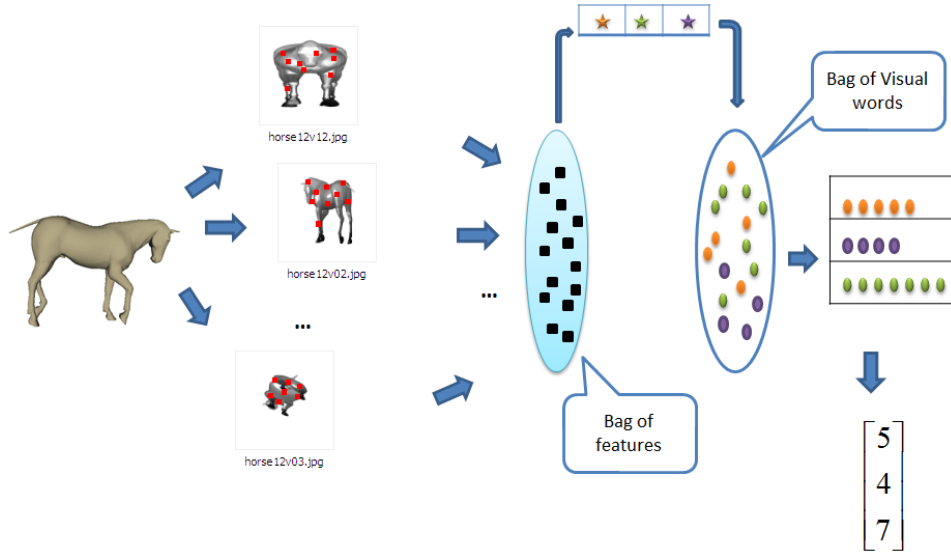
Figure 2.18 – *Bag-of-feature technique.*

the vector quantifying of salient local descriptors extracted on the depth-buffer views captured uniformly around the objects.

Ovsjanikov *et al.* (77) presented an approach to non-rigid shape retrieval similar in its spirit to text retrieval methods used in search engines. They used the heat kernel signatures to construct shape descriptors that are invariant to non-rigid transformations.

Toldo *et al.*(112) has used the bag-of-words for 3D-object categorization. Toldo's categorization framework is based on semantic segmentation. In general, the problem of segmenting a 3D object into meaningful parts is not a trivial issue. Their framework is quite sensitive to the identification of the boundaries of the meaningful part.

The main drawback of this technique that it loses the spatial relation between parts, since it ignores the position of each feature.

2.6 Conclusion and comparison of 3D-models retrieval methods.

We presented in this chapter the concept of shape matching that the measure of similarity between two 3D-objects can be reduced to a distance computing between their two descriptors. Then, we review existing methods in the literature for 3D-model retrieval, and for partial 3D-model re-

trieval. We discussed their drawback and their advantages. Generally speaking, a method for 3D-model retrieval or partial 3D-model retrieval which gives the ideal results do not exist yet.

To conclude, in this section we present a comparison table of the existing methods. As we noticed very few existing methods are able to perform well on 3D-model retrieval and none of them have very good results concerning the partial 3D-model retrieval. In this thesis, we present an approach for 3D-model retrieval. Then we enhanced our approach by using the bag-of-features technique to handle partial 3D-model retrieval.

	Category of methods	Methods	Non-rigid transformation	Noise and perturbation	Topology change	Depend on parameter	Shape model	Partial matching support	Comments
3D based methods for 3D-model retrieval	Global 3d based methods	Cord histogram	No	No	No	Yes	Mesh models	No	Local shape features are not well described neither robust toward perturbation
		Area and volume	No	No	No	Yes	Mesh models	No	More efficient for pruning data
		3d Hough descriptor	No	Yes	?	Yes (Preprocessing step needed)	Mesh models	No	Rotation invariance only
		Shape distribution	No	Yes-no	?	Yes (density of the sampled point)	Mesh models	No	Rotation and translation invariance
	Spectral embeddings	Pair-wise diffusion distance	Yes	Yes	Yes	Yes (time parameter and the density of sampled point)	Point clouds	No (Extended to support partiality)	Defined to describe the whole shape and miss the details
		Descriptors based on the spectra	Yes	Yes	Yes	No	Mesh and volume models	No	Selection of a particularly spectrum is not always the best choice
		Descriptors based on the eigenfunctions	Yes	Yes	Yes	No	Mesh models	No	Do not work on degenerated meshes
		Heat kernel signature (HKS)	Yes	Yes	Yes	Yes	Mesh models	No	To our experiments HKS misplace feature points
	Extended Gaussian images	Extended Gaussian images	No	Yes	No	Yes	Mesh models	No	Confusion in non convex objects
		Complex extended Gaussian images	No	Yes	No	Yes	Mesh models	No	Sensitive to some parameter
	Local based methods	Closed curves	Yes	Yes	No	Yes	Mesh models	Yes	Small variation of the feature point leads to a large variation in curves
		Spherical harmonics	No	Yes	No	Yes (Preprocessing step needed)	Mesh and volume models	No	The main limitation being the number of concentric spheres to detected the details
		Spectral descriptors	Yes	Yes	Yes	Yes	Mesh models	Yes	Method generates undesirable feature points or misplaces them
	Graph based methods	Reeb graphs	Yes	Yes	No	No	Mesh models	No (Extended to support partiality)	Complexity of graph matching, The enhancement of the graph is sensitive to local deformation
		Skeletal graphs	Yes	Yes	No	Yes	Volume models	Yes	Complex in calculation, Rotation invariance only
		Multi-resolution Reeb graphs	Yes	Yes	No	Yes	Mesh models	No	Quotient function affects the results
2D based methods for 3D-model retrieval		Descriptor with silhouettes (2D/3D)	No	No	Yes	Yes	Volume and mesh models	No	Different 3D shapes might have the same set of silhouette images
		Characteristic 2D views	No	No	Yes	Yes	Volume and mesh models	No	Good results for solid models only

Scalar Function On Surfaces 3

Contents

I N this chapter, we present an approach to define our invariant mapping function. Later, in chapter 4, we will construct the Reeb graph to show the stability and the invariance properties preserved by our mapping function. In chapter 5, we will define a novel method for 3D-model retrieval based on this function, and in chapter 6, we will enhance our method to handle partial 3D-model retrieval.

Our mapping function computes a real value for each vertex of the mesh which provides interesting insights to describe the topology structure of the 3D-model, and respects some important properties. It is invariant to rigid and non rigid transformations, it is insensitive to noise, it is robust to small topology changes, it does not depend on any parameters, and it is practical to compute on a discrete mesh. However, current methods that are presented in the state-of-the-art, do not have these basic properties. The definition of our mapping function consists of two steps:

First, we extract feature points located at the extremities of prominent components of the 3D-model. We combine local and global properties of the diffusion distance to detect stable feature points.

Second, these feature points are used as origin to define our mapping function based on the commute-time distance. The commute-time distance detects the local and global properties, since it takes into consideration all paths connecting two nodes on the graph and do not depend on any parameters.

This chapter is organized as follows: In section 3.1, we introduce topological structure and function on surfaces related work. Section 3.2 presents mathematical background and definitions of the heat kernel, the diffusion distance and the commute-time distance also we discussed their properties. In section 3.3 we present an approach for stable feature point extraction. We discuss the construction and the properties of our mapping function in section 3.4. Before concluding, we define the parameter settings and the implementation of our mapping function in section 3.5.

Introduction (en français)

Dans ce chapitre, nous présentons une approche pour définir notre fonction d'application. Plus tard, dans le chapitre 4, nous allons construire le graphe de Reeb pour montrer la stabilité et les propriétés d'invariance conservées par notre fonction. Dans le chapitre 5, nous allons définir une nouvelle méthode pour l'indexation des modèles 3D basée sur cette fonction, et dans le chapitre 6 nous allons améliorer notre méthode pour l'indexation partielle de modèles 3D.

Notre fonction d'application calcule une valeur réelle pour chaque sommet du maillage décrivant la structure topologique du modèle 3D. Elle possède des propriétés importantes: elle est invariante à des transformations rigides et non rigides, elle est insensible au bruit, elle est robuste à de petits changements topologiques, elle ne dépend pas de paramètres et elle est facilement calculable sur un maillage discret.

Cependant, les méthodes actuelles qui sont présentées dans l'état de l'art, ne possèdent pas ces propriétés de base. La définition de notre fonction d'application se compose en deux étapes.

Tout d'abord, on extrait des points caractéristiques situés aux extrémités des éléments significatifs du modèle 3D. Nous combinons les propriétés locales et globales de la distance de diffusion pour détecter les points caractéristiques bien localisés.

Ensuite, ces points caractéristiques sont utilisés comme origine pour définir notre fonction d'application basée sur la distance de migration pendulaire (*commute-time distance*). La distance de migration pendulaire détecte les propriétés locales et globales, car elle prend en compte tous les chemins reliant deux sommets sur le graphe et ne dépend pas des paramètres.

Ce chapitre est organisé comme suit. Notre méthode propose une solution à la définition d'une fonction scalaire appropriée qui respecte la stabilité et les propriétés d'invariance. Dans la section 3.1, nous nous sommes concentrés sur les méthodes existantes dans l'état de l'art qui traitent ce problème. La section 3.2 présente les définitions mathématiques du noyau de la chaleur, de la distance de diffusion et de la distance

de migration pendulaire dont nous discutons aussi des propriétés. Dans
la section 3.3 nous présentons une approche pour l'extraction des points
caractéristiques. Nous discutons la construction et les propriétés de notre
fonction d'application dans la section 3.4. Avant de conclure, nous définis-
sons les paramètres et l'implémentation de notre fonction d'application
dans la section 3.5.

3.1 SCALAR FUNCTION ON SURFACES RELATED WORK

The scalar function $f : S \mapsto \Re$ is defined by a real value for each vertex on a polygonal mesh. Critical points are located with the help of this function. The type and the number of these critical points are related to the topology of the mesh. In the literature, the use of such functions computed over triangulated surfaces is an important tool for a 3D-model analysis in different tasks: topological structure of a 3D-model (73), surface coding and modeling (95), molecular analysis (10), Reeb graph construction (39, 111) and so on.

Our method proposes a solution to the definition of an appropriate scalar function. Currently, such functions defined on surfaces existing in the state-of-the-art do not respect the stability and the invariance properties listed in the introduction of this chapter.

The height function was firstly applied by Shinagawa and Kunii (96) and is a simple example to understand the Reeb graph. Most of the methods define the scalar function using many techniques based on the geodesic distance (the length of the shortest path between vertices along the surface) (111, 93, 67). These functions are very sensitive to small topology changes.

Mortara and Patané (71) extracted feature points where Gaussian curvatures are dependant on a given threshold. However, the drawback of this method appears when using constant curvature surfaces.

Gebal *et al.* (39) define an autodiffusion function depending on the time variable that leads to very different levels of details. This function computes the remaining heat on each vertex after a scale time t. For scale invariance, the authors normalize the spectrum (eigenvalues) of the graph. The autodiffusion is a function based on the heat kernel, which in turn depends on eigenvalues and eigenvectors of Laplace-Beltrami operator. In the discrete space for his application, the authors use the cotangent weight (11, 80) that minimizes the Dirichlet energy. This may creates unexpected local extrema due to the negative values of geometry weights.

Lazarus and Verroust (58) proposed to map a vertex to its geodesic distance after the user choice of the source vertex. Tierny *et al.*(111) detect

the farthest two points on the mesh, then the authors compute two sets of feature points based on local maximum (a vertex such that all its direct neighbours have a lower values) from the farthest two points. The intersection of the two sets computed eliminates the unwanted feature points as it shown in Figure 3.1. The authors define the scalar function as the geodesic distance to the nearest feature point. The method's disadvantage that is its sensitivity to a small topology change.

Figure 3.1 – *The intersection of the two sets computed eliminates the unwanted feature points on a triangulated surface. (Figure taken from Tierny et al.(109))*

Ni *et al.* (73) solve the Laplace equation adding selected vertices as constraints elements, the resulting function will be harmonic. This method yields to a smooth function to eliminate local maximal point except at its boundary.

The advantage of our scalar function is shown in table 3.1. Also, this table presents an overview of these functions listed above compared to our function which in turn will be defined later in this chapter.

3.2   MATHEMATICAL BACKGROUND AND DEFINITION

In this section, we present the mathematical background needed to compute our mapping function. We firstly present the general heat kernel definition which will be used to define the diffusion distance and the commute-time distance.

	Rigid transformation	Non rigid transformation	Noise and perturbation	Topology changes	Do not depend on parameters
Height Function Shinagawa and Kunii (96)	−	−	−	−	+
Gaussian curvature Motara and Patané (71)	+	+	+	−	−
Laplace equation Ni *et al.* (73)	+	+	+	not clearly discussed	−
Geodesic distances Lazarus and Verroust (58)	+	+	−	−	−
Geodesic distances Tierny *et al.*(111)	+	+	+	−	−
Auto diffusion function Gebal *et al.* (39)	+	+	+	not clearly discussed	−
Our Function	+	+	+	+	+

Table 3.1 – *Scalar functions defined on surfaces existing in the state-of-the-art compared to our function.*

3.2.1   Definition of the heat kernel

Let S be a connected closed Reimannian manifold. Given an initial heat distribution $U : S \times [0, \infty[\to \Re$ for all time t the $U(0, x) = f(x)$, $f(x)$ is a given function on the mesh. The heat diffusion process on S is governed by heat equation:

$$\Delta_S U(x, t) = -\frac{\delta U(x, t)}{\delta t} \tag{3.1}$$

for $x \in S$ and $0 < t < \infty$, and the distribution of the heat U is a function defined on $S \times [0, \infty[\to \Re$, the solution $U(x, t)$ is the amount of heat on the surface S at point x in time t. Δ_S is a generalization of the Laplacian to non-euclidean domains (from flat spaces to manifolds) called Laplace-Beltrami operator. We consider that the operator is separable, initial and boundary conditions are on constant-coordinate surfaces. Using the separation of variables method, we get the spectral problem:

$$\Delta_S U(x, t) = \lambda^2 U(x, t) \tag{3.2}$$

S is a compact manifold, Δ_S is compact self-adjoint operator in $L^2(S)$. By the finite-dimensional spectral theorem such operators have an orthonormal basis in which the operator can be represented as a diagonal matrix with entries are real numbers. The basis is built on the surface from the eigenfuctions ψ_n and all the eigenvalues λ_n are positives. The operator $-\Delta_S$ is represented by the diagonal matrix λ_i ($0 \leq i \leq n$). By the spectral theorem the operator $e^{t\Delta_S}$ has the diagonal matrix entries $e^{-t\lambda_i}$.

$$-\Delta_S \psi_n = \lambda_n \psi_n \tag{3.3}$$

Since the equation 3.1 is a homogeneous linear equation the initial heat distribution $U(0, x)$ can be written in the basis as the following expansion:

$$U(0, x) = \sum_{n=0}^{\infty} < U(0, x), \psi_n(x) > \psi_n(x) \tag{3.4}$$

where the inner product is defined as the surface integral

$$< U(0, x), \psi_n(x) > = \int_S (U(0, y)\psi_n(y)) dy \tag{3.5}$$

The solution of the problem can be written as:

$$U(t, x) = \sum_{n=0}^{\infty} < U(0, x), \psi_n(x) > \psi_n(x) e^{\lambda_n t} \qquad (3.6)$$

Finally

$$U(t, x) = \int_S \underbrace{\sum_{n=0}^{\infty} e^{\lambda_n t} \psi_n(y) \psi_n(x)}_{K(t, x, y)} U(0, y) dy \qquad (3.7)$$

$$K(t, x, y) = \sum_{n=0}^{\infty} e^{\lambda_n t} \psi_n(y) \psi_n(x) \qquad (3.8)$$

$K(t, x, y)$: $C^{\infty}((0, \infty) \times S \times S) \to \Re$ is the fundamental solution of the heat equation corresponding to the initial condition of an initial point source of heat at a known position called the *heat kernel*. The initial point source of heat is $k(t, x, .) = H(t, \delta(x))$ where H denote the heat distribution at time t and $\delta(x)$ is Dirac delta function at x: $\delta(x, z) = 0$ for any $z \neq x$, and $\int_S \delta(x, z) dz = 1$. For more details see Grigorian (41).

Properties of the heat kernel

The heat kernel is computed using the eigendecomposition of Laplace-beltrami operator. The first eigenvectors (corresponding to small eigenvalues) are smooth, slowly varying functions on the mesh, and the last eigenvectors have high frequency (rapid oscillations). For example, the first eigenvector is the constant vector, that is, the smoothest mesh function that does not vary at all. In fact, the Laplacian eigenbasis is an extension of the discrete Fourier basis to irregular domains (107). The eigenvalues can be interpreted as frequency and $K(t, x, y)$ can be seen as as a low-pass filter function (robust to perturbation and noise).

The Laplace-Beltrami operator is intrinsically linked to the geometry of the mesh and involves intrinsic properties and invariance to isometric deformations, which implies the properties to the heat kernel as well. The heat kernel can be used to match articulated shapes such as animals in different poses.

The heat kernels depend on the parameter t that reflects the multi-scale property which means for small time parameter t the heat kernel is determined by small neighborhoods and reflects local properties of the shape. These neighborhoods grow bigger as t increases to reflect the global

properties. The heat kernel's properties was explored in different domains for construction of global shape descriptor (87, 11), construction of multi-scale shape descriptor (100), and parametrization of surfaces (59). We will apply the heat kernel in order to compute the diffusion distance for feature points extraction and also to compute the commute-time distance to define our mapping function.

3.2.2 Diffusion distance

Coifman *et al.*(18) introduced diffusion maps and diffusion distances as a method for data parametrization and dimensionality reduction. In general, the diffusion kernel $k(x,y)$ reflects the degree of proximity or measures the similarity between two points x and y it means that $k(x,y) \approx connectivity(x,y)$. It can be used to define a metric on S.

$$d^2(x,y) = \|k(x,.) - k(y,.)\|^2_{L^2(S)} \tag{3.9}$$

Let $k(x,y)$ be a positive symmetric diffusion kernel constructed over all pairs of points on the surfaces and n is the number of points of the shape S. That leads to $n \times n$ symmetric, positive, semi-definite matrix.

$$p(x,y) = \frac{k(x,y)}{\Sigma_y k(x,y)} \tag{3.10}$$

where $\Sigma_y k(x,y)$ is the sum of the elements of each row. Since $p(x,y) \geq 0$; $\Sigma_y p(x,y) = 1$; then $p(x,y)$ can be interpreted as the probability for a random walker on the shape to jump from x to y in a single time step. The corresponding matrix $P = p(x,y)$ is the transition matrix of this Markov chain in a single time step. We multiply the entry of the matrix P by $\sqrt{\frac{\Sigma_y k(x,y)}{\Sigma_x k(x,y)}}$ to transform the matrix P to symmetric version we called it ρ.

$$\rho(x,y) = p(x,y)\sqrt{\frac{\Sigma_y k(x,y)}{\Sigma_x k(x,y)}} \tag{3.11}$$

Using the random walk formulation the t-th power of the matrix ρ determine the transition probability from x to y in t steps. Then the diffusion distance between two points x, y is defined as

$$d_t^2(x,y) = \sum |\rho^t(x,.) - \rho^t(y,.)| \tag{3.12}$$

In spectral expression using the eigendecomposition of ρ, the diffusion distance can be written as

$$d_t^2(x,y) = \sum_{i=0}^{\infty} k^{2t}(\lambda_i)\psi_i(x)\psi_i(y) \tag{3.13}$$

Where λ_i and $\psi_i(.)$ are the *i-th* eigenvalue and eigenfunction respectively of Laplace-Beltrami operator.
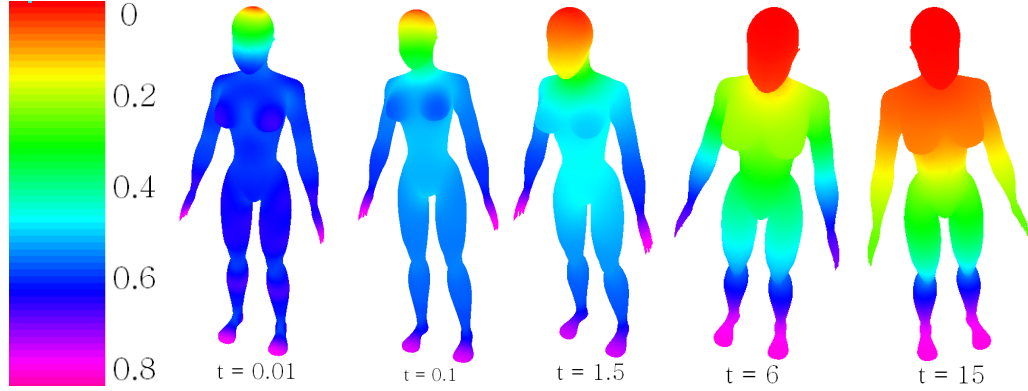
Figure 3.2 – *Diffusion distance diffused from a vertex located on the top of the head in different scaled time t.*

In our methods, we compute the diffusion distance based on the heat kernel due to nice properties listed in the previous section, consequently $k^{2t}(\lambda_i) = e^{-2\lambda_i t}$. Back to equation 3.13 the diffusion distance can be represented as:

$$d_S(t,x,y)^2 = \sum_{i=1}^{\infty} e^{-2\lambda_i t}(\psi_i(x) - \psi_i(y))^2 \tag{3.14}$$

3.2.3 Commute-time distance

Qiu and Hancock (83) presents a scale invariant kernel, the *commute-time kernel*. The commute-time distance is defined similarly to the diffusion distance. But the commute-time distance takes into consideration all paths connecting a pair of nodes in the graph, not only paths with length t. This is done by summing the diffusion distances over the possible discrete time-steps on the graph. Let x, y two points on the graph S, and $d_S(t,x,y)^2$ the diffusion distance between x and y. The commute-time distance $dc_S(x,y)^2$ is represented as:

$$dc_S(x,y)^2 = \sum_{t=1}^{\infty} d_S(t,x,y)^2 \tag{3.15}$$

Then

$$dc_S(x,y)^2 = \sum_{t=0}^{\infty}\sum_{i=1}^{\infty} e^{-2\lambda_i t}(\psi_i(x) - \psi_i(y))^2 \tag{3.16}$$

By the property of the power series, we got $\sum_{t=0}^{\infty} e^{-2\lambda_i t} = 1/1 - e^{-2\lambda_i}$ finally the commute-time distance is represented as:

$$dc_S(x,y)^2 = \sum_{i=1}^{\infty} 1/\alpha_i(\psi_i(x) - \psi_i(y))^2 \tag{3.17}$$

Discussion

The diffusion and commute-time distances are an intrinsic metrics and respect the invariance to isometric deformation. Since the diffusion distance and commute-time distances are a direct consequence of the invariance of Laplace-Beltrami operator.

Also, the diffusion distance can be understood using random walks, and the number of steps of a random walk is limited by the parameter t. The parameter t plays the role of scale. That is to say, when t is small, x and y are linked by short paths, and diffusion takes place only over a very local neighborhood. As t becomes larger, the diffusion occurs over a larger area. In other words, as t goes from zero to infinity, the diffusion map measures the connectivity of x and y with a specific path length.

The diffusion distance describes locally and globally the shape depending on the parameter t. We combine these properties in a technique to extract feature points. On the other hand, the commute-time distance is the sum of the diffusion distance over all possible paths connecting x and y which makes it time independent. We use feature points extracted as origin to define our mapping function based on the commute-time distance.

3.3 FEATURE POINT EXTRACTION

Feature points is a concept introduced by several authors (71, 51), for which it is hard however to find a formal definition. We define vertices
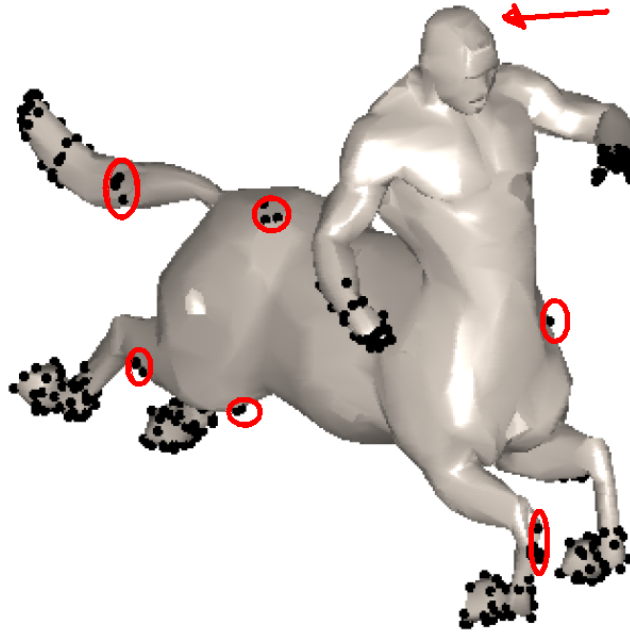
Figure 3.3 – *Feature point detected based on the heat kernel. To our concept, feature point detected other than those located on the extremities are undesirable feature points. Also, the method introduced by Bronstein et al.(11)) missed to detect feature point located on the top of the head of the centaur. (Figure taken from Bronstein et al.(11))*

located on the extremities of the important elements of the 3D-model as feature points. These feature points will be used as origins of our mapping function.

In the literature, many techniques were proposed to detect feature points (51, 111, 39). Recently, several methods (11, 100) are using the heat kernel to detect feature points. These methods compute the amount of heat remaining at each vertex after a large time. These functions detect feature points automatically since the quantity of heat that remains, will be bigger on features or close to them. But to our experiments, this function generates undesirable feature points or misplaces them. Which means to our concept taht feature points should be located only on the extremities. Methods based on the heat kernel generate feature points on the extremities and on other parts. Also these methods miss to detect extremities of an important part, like the missing feature point that should be located on the top of the head of the centaur as it is shown in figure 3.3.

Tierny *et al.* (111) proposed a crossed analysis method using two

geodesic function. The author's algorithm produces well-localized feature points but very sensitive to topology changes.

Our Method   is inspired from Tierny *et al.* (111) using the same technique with diffusion distance. The method starts by feature points extraction using the diffusion distance. In a large time variable t, global properties are detected and the farthest two feature points (where the diffusion distance is minimum) are computed (see figure 3.4a). Let V_1 and V_2 be the farthest feature points. These two feature points are used as origins for two scalar function defined as

$$f_{V_1}(v, t) = d_S(t, v, V_1)^2 = \sum_{i=1}^{\infty} e^{-2\lambda_i t} (\psi_i(v) - \psi_i(V_1))^2 \qquad (3.18)$$

$$f_{V_2}(v, t) = d_S(t, v, V_2)^2 = \sum_{i=1}^{\infty} e^{-2\lambda_i t} (\psi_i(v) - \psi_i(V_2))^2 \qquad (3.19)$$

In a small variable time t, we compute local minimum diffusion distance (vertex that all its level-one neighbours have a higher value) to detect local properties. Let S_1 and S_2 be the set of local minima of f_{V_1} and f_{V_2} respectively (see figure 3.4b and figure 3.4c). The set of feature points is the union of S_1 and S_2 presented in figure 3.4d.

In the discrete space, feature points will appear in the same neighbourhood of the vertex not exactly on the same ones this is why we put a limit dependent on a two level neighbourhood (see figure 3.5). Then we extract the nearest feature point to the barycenter of the all points constructing the region. This combination of local and global properties to extract feature points shows good results on different 3D-models under different rigid and non-rigid transformations (see figure 3.6).

3.4   Definition of the mapping function

The concept of the mapping function is to reveal the most meaningful parts of the model. Our mapping function computes a real value for each vertex which provides interesting insights to describe topology structure of the 3D-model, and respects some important properties. It is invariant to rigid and non rigid transformations, it is insensitive to noise, it is robust
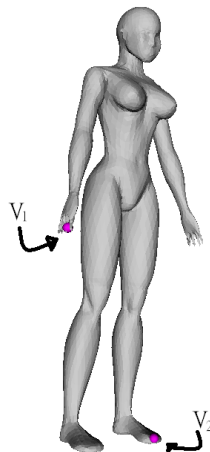
(a) *V_1 and V_2 are the farthest two feature points*

(b) *The set of local minima of f_{V_1}*

(c) *The set of local minima of f_{V_2}*
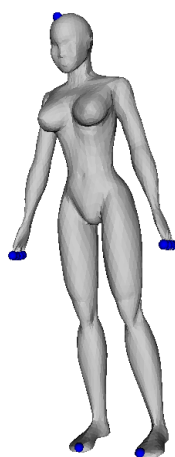
(d) *Final Feature points set $= V_1 \cup V_2$*

Figure 3.4 – *Approach for feature points detection. Figure 3.4a shows the farthest two feature points detected. Figure 3.4b and figure 3.4c show the set of local minima of f_{V_1} and f_{V_2} respectively and figure 3.4d shows the final set of feature points.*
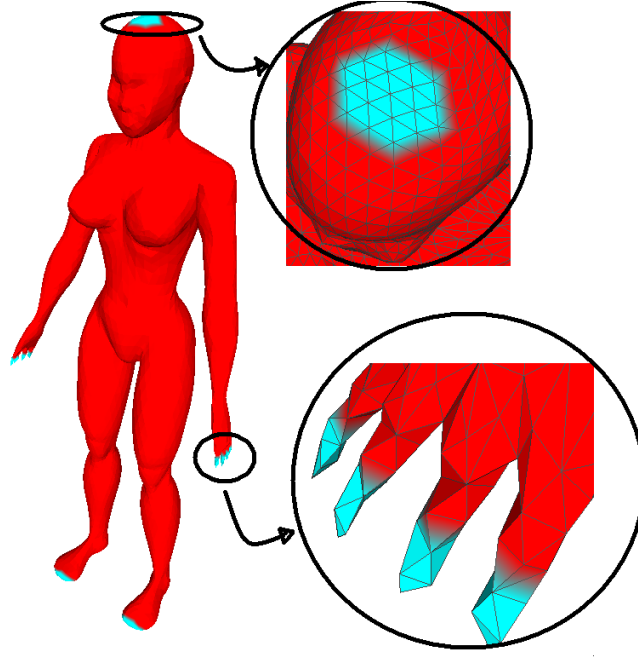
Figure 3.5 – *Region detector for feature point. limit dependent on a two level neighbourhood*

to small topology changes, and it does not depend on parameters, and it is practical to compute on a discrete mesh. These properties are discussed in section 3.4.2.

Later, in chapter 4 we will use this function and we will add a perturbation strategy to construct Reeb graph. In chapters 5 and 6 this function will be used to construct closed curves as a descriptor for 3D-model retrieval and partial 3D-model retrieval respectively.

3.4.1   Construction of the mapping function

Tierny *et al.* (111) define a mapping function that computes for each vertex v the geodesic distance to the nearest feature point. This function does not handle small topology changes and is very sensitive to noise. Based on the same technique using the commute-time distance, we define our mapping function. The commute-time distance takes into consideration all paths connecting a pair of nodes in the graph, a small topology change does not affect enormously the results. We use this distance to define our mapping function F_m that computes for each vertex v the commute-time distance to

(a) *Null 3D-model*

(b) *Partiallity*

(c) *Sampling changed*

(d) *Scale change*

(e) *Affine transformation*

(f) *Isometric transformation*
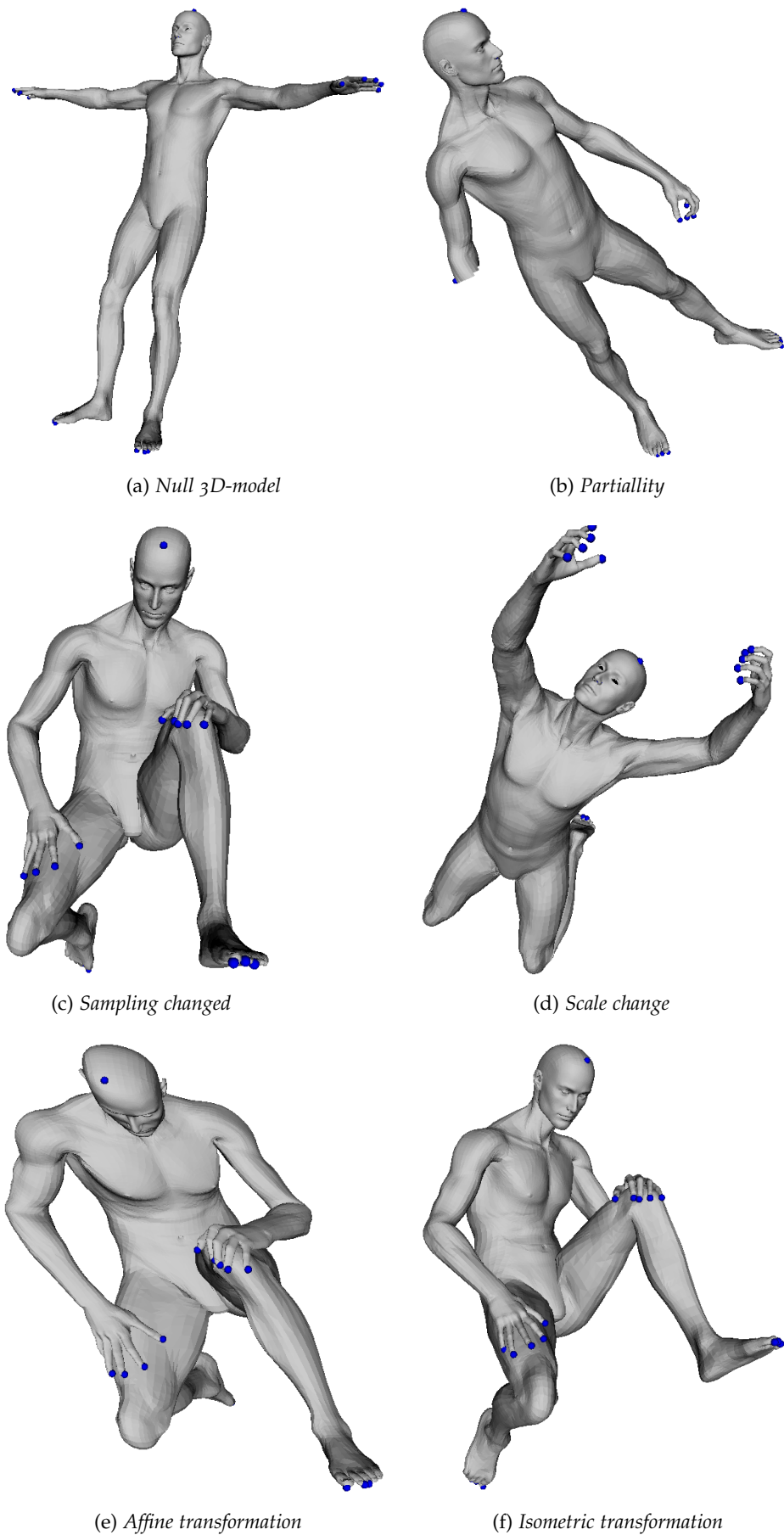
Figure 3.6 – *Feature points extracted from different poses of a 3D-model taken from SHREC 2011 database.*

the nearest feature point.

$$F_m(v) = max(dc_S(v, V_i), i = 1..nbV_i)) \tag{3.20}$$

where V_i is the ith feature point and nbV_i is the number of feature points.

(a) *Null 3D-model*   (b) *Partiallity*

(c) *Sampling changed*   (d) *Scale change*

(e) *Affine transformation*   (f) *Isometric transformation*
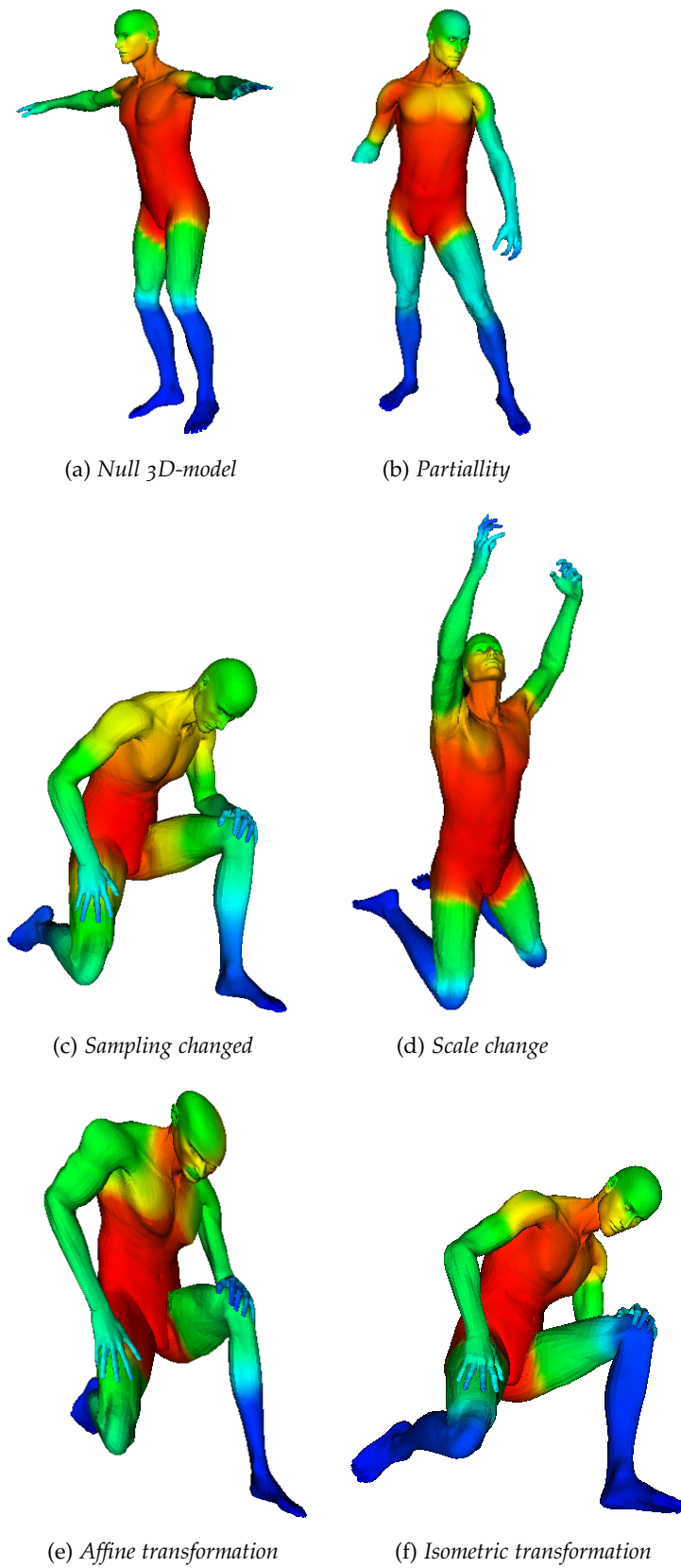
Figure 3.7 – *Figure(a) to figure(f) show the robustness of the mapping function against variations in 3D-model pose, red to blue colors express the increasing values of the mapping function.*

3.4.2   Properties of the mapping function

Our mapping function respects important properties and reveals the most meaningful parts of the model that can be useful to describe a 3D-model as it shown in figure 3.7. Since the commute-time distance is a direct consequence of the invariance of Laplace-Beltrami operator presented in section 3.2. In the following we discuss these properties.

- **Topology information:**

  Our function computes for each vertex the commute time distance to the nearest feature point. In other words, each vertex is assigned with the commute time distance to a feature point, all vertices assigned to the same feature point define a region. Vertices having the same distance to two or more feature points are also grouped to define another region. In this way, our mapping function segments the 3D-model into sub-parts and the evolution of these regions can be used to describe the topology of the mesh. Later, in chapter 4 we analyze the evolution of the region to construct the Reeb graph.

- **Geometric information:**

  The mapping function is defined to be dependent only on the structure of the mesh. (for more details see section 3.2).

- **Invariance to isometric deformation:**

  commute-time distances are an intrinsic metric and respect the invariance to isometric deformations. Since the commute-time distances are a direct consequence of the invariance of Laplace-Beltrami operator which depends only on the structure of the mesh.

- **Robustness to noise and perturbation:**

  The mapping function is defined by the commute-time distance. It is computed using the eigenfunction and eigenvalue of the Laplace-Beltrami operator. The first eigenvectors (corresponding to small eigenvalues) are smooth, slowly varying functions on the mesh, and the last eigenvectors have high frequency (rapid oscillations).

  For example, the first eigenvector is the constant vector, that is, the "smoothest" mesh function that does not vary at all. In fact, the

Laplacian eigenbasis is an extension of the discrete Fourier basis to irregular domains (107). The eigenvalues can be interpreted as frequencies and we defined $K(t, x, y)$ using the first eigenvalues. Consequently it can be seen as a low-pass filter function (robust to perturbation and noise).

- **Invariance to uniform scaling:**
  After the eigendecomposition of Laplace-beltrami, we normalized the spectrum (eigenvalues) of the mesh. In this way, all models with different scales are computed almost in the same base defined by their eigenvalues.

- **Handle small topology change:**
  The commute-time distance takes into consideration all paths connecting a pair of nodes in the graph, a small topology change does not affect enormously the results.

- **Does not depend on parameter:**
  Our function is based on the commute-time distance that considers all paths connecting two nodes on the graph, unlike Gebal *et al.* (39) who define the autodiffusion function depending on the time variable that leads to very different levels of details. All other parameters such as the number of the first eigenvalues are fixed for all 3D-models.

3.5 Parameter settings and implementation

In the discrete space, to formulate the diffusion distance and the commute-time distance on surface meshes, we need to discretize the Laplace-Beltrami operator and compute the eigenfunctions and the eigenvalues.

In the literature several discretizations of Laplace-Beltrami operator have been proposed. Pinkall and Polthier (81) were the firsts who introduce the geometric approach called the cotangent weights. Generally, the cotangent weight discritization can be written in the following form:

$$\Delta_S f(p_i) = \sum_{j \in N(i)} w_{ij}(f(p_i) - f(p_j)) \qquad (3.21)$$

Where p_i and p_j are the vertices of the surface mesh S, $N(i)$ is the set of one ring neighbours of the vertex p_i, f is a function defined on the triangulated surface S, and w_{ij} are the weights defined on the geometric of the mesh.

Desbrun *et al.* (22) and Meyer *et al.* (68) took almost the same discretization as Pinkall and Polthier (81) and added a normalization factor c_i. Then the discretization is written in the following form:

$$\Delta_S f(p_i) = c_i \sum_{j \in N(i)} w_{ij}(f(p_i) - f(p_j)) \qquad (3.22)$$

where Desbrun *et al.* (22) define $c_i = \frac{3}{Area(p_i)}$ where $Area(p_i)$ is the area of the triangles around p_i.

Meyer *et al.* (68) define $c_i = \frac{1}{Area_M(p_i)}$ where $Area_M(p_i)$ is the area of the region shown in figure 3.8.
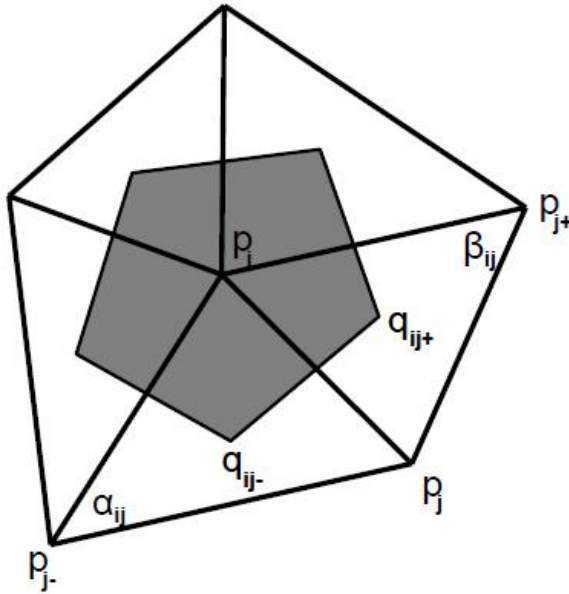
Figure 3.8 – *Definition of the angles and the area around the vertex, q_{ij} is circumcenter*

In practice, we numerically computed the eigenfunctions and the eigenvalues using the discretization proposed by Meyer *et al.* (68). The problem can be transformed in matrices notation where the eigensystem

defined in the equation 3.3 is simplified to generalized symmetric eigen-value problem:

$$M\psi = \lambda S\psi \qquad (3.23)$$

Where M is a symmetric matrix whose entries $m_{ij} = \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2}$ are the cotangent weights. (α and β) are the angles shown in figure 3.8. The matrix S whose entries are S_i defined on the area around the vertex p_i.

We solve the generalized eigenvalue problem using the Implicitly Restarted Arnoldi Method implemented in MatLab.

3D-models with boundaries are solved by imposing the Newman boundary condition (114). We force the outward normal derivative to be zero on the boundary vertex.

We define the basis by 50 eigenfunctions related to the 50 smallest eigenvalues. The first 50 eigenvalues are fairly enough to detect the most important details and smooth enough to eliminate the noise, as we noticed the eigenvalues can be interpreted as frequency and K(t,x,y) can be seen as a low-pass filter function. To extract feature points described in section 3.3, the diffusion distance is estimated in a small and in a large variable time t. For small t ranging in [1..2], the diffusion is propagated significantly to detect local properties. For a large $t > 15$, the diffusion distance remains almost unchanged; so we fix it experimentally to 20.

3.6 Conclusion

To conclude, we presented in this chapter a robust invariant mapping function which reveals the most meaningful parts of the 3D-model.

First, we start by introducing the mathematical background and the properties of the heat kernel, diffusion distance and commute time distance. Then, we extract feature points defined on the extremities of a 3D-model. Our technique combines the local and global properties of the diffusion distance to detect stable feature points. We present feature point extracted on different 3D-models with different poses and transformations.

Second, these feature points are used as origin to define our mapping function based on the commute time distance. This mapping function

respects good properties as the invariance to rigid and non rigid transformations, the insensitivity to noise, the robustness to small topology changes, and do not depend on parameter.

The main drawback of our function is presented in 3D-models like cups where feature points are not well defined due to the shapes itself of these 3D-models where the extremities are difficult to define.

In the next chapter, we will use this function to construct Reeb graph which encodes the topological information of the 3D-model and we will prove the robustness of the mapping function.

In chapters 5 and 6 we will present two methods for 3D-model retrieval and partial 3D-model retrieval respectively based on this function.

REEB GRAPH COMPUTATION AND TOPOLOGICAL ANALYSIS OF THE MESH

4

Traditional skeleton extraction and mesh segmentation approaches are not clearly defined to preserve topological properties of a 3D-model faced to the Reeb graph theory (84). We see the benefits of topology graphs, such as Reeb graph, in many applications in diverse fields. It has shown to be interesting for shape description (6), surface parametrization (99), mesh segmentation (93), 3D-model retrieval (67), and so on.

The mapping function defined in the previous chapter section 3.4 can be seen as a piecewise linear function but not a Morse function.

When dealing with topological structure based on Reeb graph, obviously, we have to define an appropriate piecewise linear Morse function. This function should respect the invariance to rigid and non rigid transformations, the insensitivity to noise, the robustness to small topology changes, and the independence on parameters. The definition of such a Morse function remains an open question (6).

To answer this question, we propose in this chapter a solution based on our function defined in the previous chapter. Also we present the stability and the invariance properties of our piecewise linear Morse function for Reeb graphs computation.

We organize this chapter as follows: The first section 4.1 introduces the Morse theory in continuous and discrete space. Followed by section 4.2 which presents the computation of Reeb graph also in the continuous and discrete space.

In section 4.3 we transform our piecewise linear function defined in the previous chapter to a piecewise linear Morse function by adding a perturbation strategy. Then, section 4.4 is about Reeb graph computation and topological analysis of the mesh. Before the conclusion, the experiments that prove the efficiency of our approach are explored in section 4.5. Section 4.1 to section 4.3 are taken from Tierny's Ph.D thesis (110).

Introduction (en français)

Les approches traditionnelles de segmentation d'un modèle 3D et l'extraction de squelettes ne sont pas efficaces pour préserver les propriétés topologiques d'un modèle 3D par rapport à la théorie de graphe

de Reeb dans de nombreuses applications. Les graphes de Reeb sont utilisés pour l'analyse de forme de modèles 3D (6), pour la paramétrisation des surfaces (99), pour la segmentation de maillages (93), pour l'indexation de modèles 3D (67), etc.

La fonction d'application définie dans le chapitre précédent est vue comme une fonction linéaire par morceaux, mais pas une fonction de Morse. Pour analyser la structure topologique d'un modèle 3D par les graphes de Reeb, il faut definir une fonction de Morse simple linéaire par morceaux. Cette dernière doit respecter les propriétés d'invariance citées dans le chapitre précédent en décrivant le modèle 3D. La définition d'une telle fonction de Morse reste une question ouverte (6).

Pour répondre à cette question, nous proposons dans ce chapitre une solution basée sur notre fonction définie dans le chapitre précédent. En outre, nous présentons la stabilité et les propriétés d'invariance de notre fonction de Morse linéaire par morceaux en l'appliquant sur le calcul de graphes de Reeb.

Nous organisons ce chapitre comme suit: la première section introduit la théorie de Morse dans l'espace continu et discret. La section suivante présente le calcul du graphe de Reeb aussi dans l'espace continu et discret.

Dans la section 4.3 nous transformons notre fonction linéaire par morceaux, définie dans le chapitre précédent, en une fonction de Morse linéaire par morceaux en ajoutant une stratégie de perturbation. Ensuite, la section 4.4 concerne le calcul de graphe de Reeb et l'analyse topologique du maillage. Avant la conclusion, les expériences qui prouvent l'efficacité de notre approche sont discutées dans la section 4.5. Les sections 4.1 à 4.3 sont tirées de la thèse de Tierny (110).

4.1 MORSE THEORY

In the beginning of the twentieth century, Marston Morse (70) presents the Morse theory (32, 27) in differential topology to characterize the topology of a manifold. The Morse theory analyzes the topology of the domain (in our case the 3D-mesh) from the study of a scalar function defined on its domain. these scalar functions are called *Morse functions*.

John Hart (42) illustrates the Morse theory by a simple example. The observation of a doughnut progressively immersed in a coffee shown in figure 4.1. He observed the evolution of the curve resulting from the intersection of the coffee surface and that of the doughnut. This intersection has been emphasized by showing the immersed portion of the doughnut surface on the left of the pictures in figure 4.1.

Along the immersion, one can notice that the topology of the intersection curve evolves at very precise configurations of the doughnut. In particular, the start point is the point where the doughnut touches the coffee surface. Then, it progressively shifts to a connected closed curve, as shown in figure 4.1b until this connected closed curve bifurcate into two disjoint components and the intersection curveś topology varies at another critical point called bifurcation point (figure 4.1d). These two connected components individually evolve until they reconnect in another symmetric critical point (figure 4.1e) called junction point. Finally, the intersection curve evolves as a single connected component and fall to a point when the doughnut is nearly completely immersed (figure 4.1f).

An interesting observation of this particular experiment is that the intersection curve disconnected as many times as it reconnected. This observation is a topological invariant of the doughnut surface.

This experiment showed that by focusing on the topology of some function level lines and specifically on the configurations where this topology varies, one can infer some topological invariant of the whole surface.
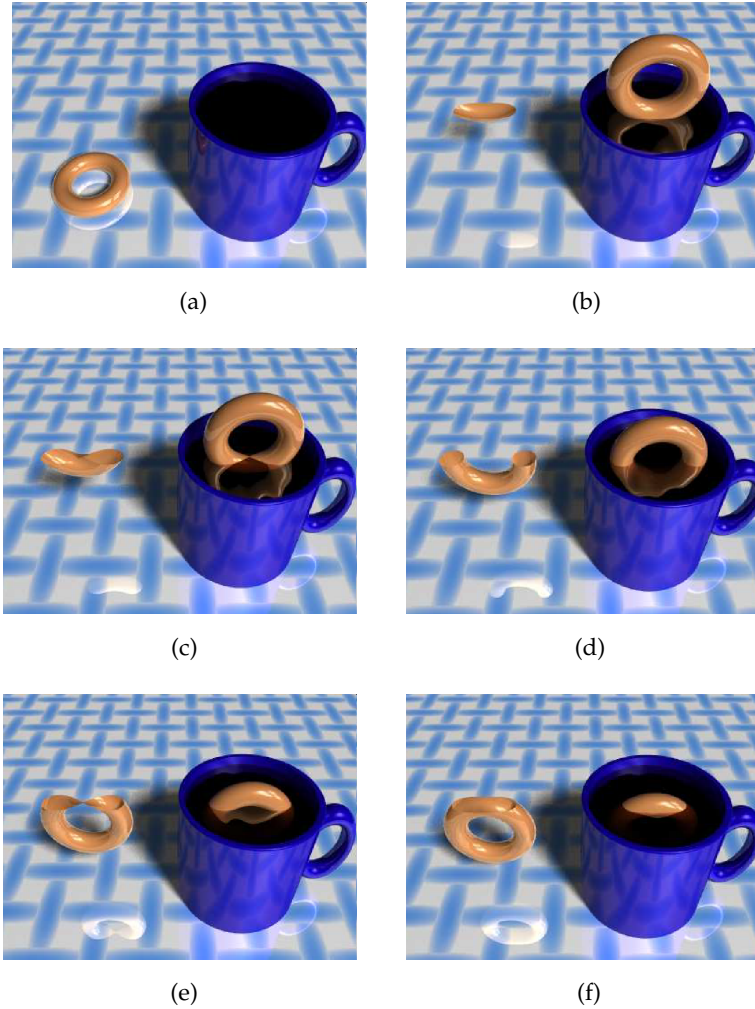
Figure 4.1 – *A shiny doughnut and a cup of coffee 4.1a set the stage to demonstrate Morse theory. As the doughnut first touches the coffee 4.1b, the portion of the doughnut's surface in the coffee changes from the empty set to a shape homeomorphic to a disk. At the instant the coffee reaches the doughnut hole 4.1c, the topology of the dunked portion changes from a disk to a truncated cylinder 4.1d. At the instant the doughnut hole is completely immersed 4.1e, the topology of the dunked portion changes from a cylinder to a truncated torus 4.1f. Releasing the doughnut causes the coffee to completely engulf the doughnut, completing the surface of the torus. Figure taken from Hart (42).*

Definitions:

Let f be a real valued function defined on a compact manifold M, $f : M \rightarrow \Re$ and a point $p \in M$.

A point p is a **critical point** of f if the *gradient* (partial derivatives of all orders in p) of the function at the point p equal zero else if p is not a critical point is called a **regular point**.

Using the *Hessian* matrix $H(p)$ (the matrix of second order partial derivatives of f at point p)

$$H(p) = \begin{pmatrix} \frac{\delta^2 f}{\delta x_1^2}(p) & \cdots & \frac{\delta^2 f}{\delta x_1 \delta x_n}(p) \\ \vdots & \ddots & \vdots \\ \frac{\delta^2 f}{\delta x_n \delta x_1}(p) & \cdots & \frac{\delta^2 f}{\delta x_n^2}(p) \end{pmatrix}$$

we can identify if the critical point p is *non-degenerate*.

a critical point p of a real valued function f is *non-degenerate* if the Hessian matrix of f at p is non singular (a matrix is singular if its determinant is equal to 0).

A real valued function f defined on a compact manifold M is called a **Morse function** if all its critical points are *non-degenerate*.

4.1.1 Morse theory in the discrete space

Banchoff (3) investigated to transfer the Morse theory from the continuous space to the discrete space (in our case triangulated surfaces). A piecewise linear function on a triangulated surface can be seen as a sampling of a function defined on a manifold. It is clear that a piecewise linear functions are definitely not Morse functions. The author proved that it is possible to define piecewise linear functions, called piecewise linear Morse functions, whose properties resemble those of Morse functions in the continuous setting. The author extends the Morse theory to triangulated surfaces.

We defined in the previous section that critical points are the points of the manifold where the gradient of the function vanishes.

Several authors (106, 28) approximated these critical points to vertices on the manifold, and they defined regular vertices and critical vertices. Figure 4.2 presents a regular vertices and some critical vertices. We can
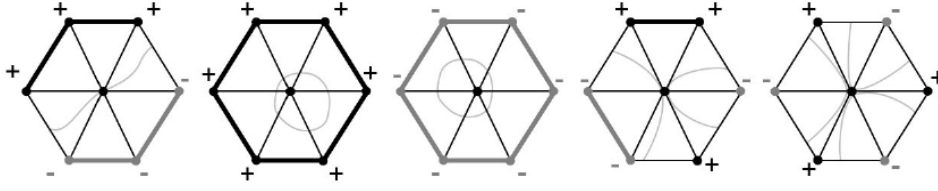
Figure 4.2 – *A regular vertex and some critical vertices. From left to right: regular vertex, a minimum non-degenerate critical vertex, a maximum non-degenerate critical vertex , a non-degenerate critical vertex and a degenerate critical vertex.(figure taken from (110))*

notice that the topology of f where the level lines (bright grey) splitting in more than two connected components as it is shown in the last right image in figure 4.2, then this vertex is a degenerate vertex and the piecewise linear function is not a Morse function.

If all the critical vertices of the piecewise linear function are non-degenerate this function is called a piecewise linear Morse function. In section 4.3 we will introduce a perturbation strategy to transform a piece-wise linear function to piecewise linear Morse function in order to construct the Reeb graph.

4.2 REEB GRAPH

The Reeb graphs have been introduced by Georges Reeb (84). We see the benefits of topology graphs such as Reeb graph in many applications in diverse fields. It has shown to be interesting for shape description (6), surface parametrization (99), mesh segmentation (93), 3D-model retrieval (67), and so on.

The Reeb graph is an interesting graph to describe topology structure that encodes the connectivity of its level sets based on the critical points of a Morse function. In continuous space, Reeb graph is defined as follow:

Let S be a compact manifold and $f : S \rightarrow \Re$ be a Morse function on S. The Reeb graph of f is the quotient space of f in $S \times \Re$ by the equivalence relation "$\sim$" defined as

$$(v_i, f(v_i) \sim (v_j, f(v_j)) \tag{4.1}$$

if and only if $f(v_i) = f(v_j)$ and v_i, v_j belong to the same connected component of $f^{-1}(f(v_i))$.

In an other way, all nodes having the same value under f are represented by one element in the Reeb graph. Figure 4.3 presents the height function f, few level lines, concentric circles which correspond to the index of the critical points, and the Reeb graph on the right of the figure, such that each equivalence class is contracted to a point on the bold lines.
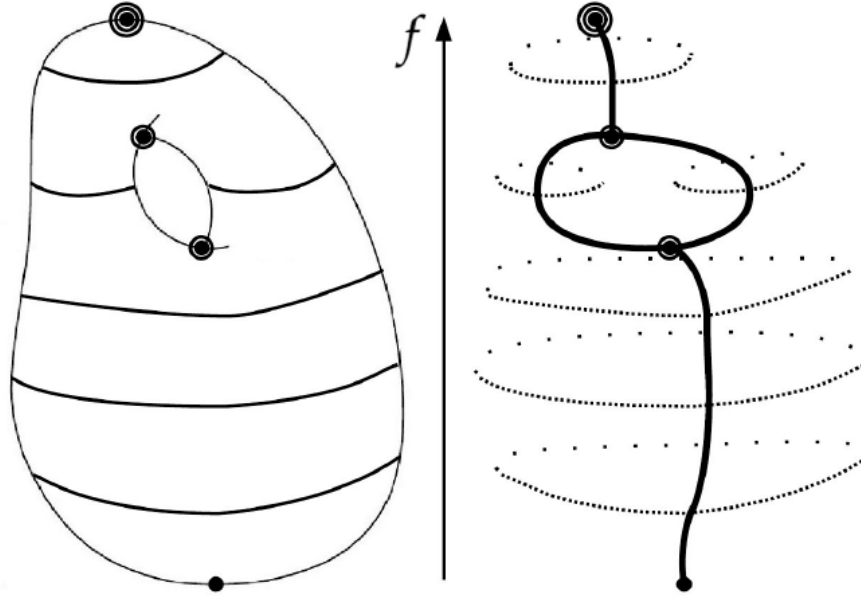
Figure 4.3 – *A smooth compact 2-manifold M and the Reeb graph of its height function f, both embedded in $\Re^3$.(Figure taken from (110))*

The Reeb graph inherits from the Morse theory the topological descriptive, also the Reeb graphs provide an additional description from the connectivity of the critical points through the tracking of level set topology evolution, resulting in an expressive topology representation.

4.2.1 Rebb graph in the discrete space

In the discrete space, the computation of Reeb graphs on triangulated surfaces requires piecewise linear Morse function where all the critical vertices of the piecewise linear function are non-degenerate.

A simple local perturbation strategy discussed in section 4.3 can be added to the input function to ensure that each vertex has a distinct value

on the triangulated surface and transform the degenerated points to non-degenerated ones, in order to transform the input function to a Morse function.

Shinigawa *et al.* (95) introduce the first algorithm for Reeb graph computation in the discrete space that runs in $O(n^2)$ steps with n the number of edges of the triangulated surfaces. Cole-McLaughlin *et al.* (19) proposed a Reeb graph computation algorithm for triangulated surfaces of arbitrary genus running in $O(nlog(n))$ steps (with n the number of edges). The main drawback of their approaches is that the equivalence classes of the output Reeb graphs do not explicitly encode contours, they just contract them to points in the graph. Consequently, it is possible to map a vertex to its equivalence class, but not its inverse since contours are not explicitly stored in data-structures. In the context of shape matching, some authors (67, 8), propose to compute Reeb graph approximations by hierarchies of dichotomies of the function base domain, called multi-resolution Reeb graphs. Such representations have shown useful in the context of shape comparison but they might not capture the topological descriptive properties of piecewise linear Morse functions since they are not based on critical vertex analysis.

Despite this amount of literature, it is hard to find a formal definition of the transposition of the Reeb graph concept to the discrete setting. Recently, Pascucci *et al.* (79) proposed a robust technique for the computation of Reeb graphs for functions defined on the surfaces. The algorithm is robust in handling non-manifold meshes and general in its applicability to input models of any dimension. However, their method is very applicable and simple to implement but not very clear how to deal with degenerated critical points. For this issue and to ensure that all critical points are non-degenerated points, we applied a perturbation strategy. For more information on Reeb graph and Morse theory see Tierny thesis (110).

4.3   Perturbation strategy of the mapping function

In the discrete space, Reeb graph needs to be defined relatively to a piecewise linear Morse function. One of the interest of Reeb graphs is that their

invariance properties and the topological characteristics of the surface can be formulated through the definition of the piecewise linear Morse function. Figure 4.4 illustrates an example of Reeb graph based on the piecewise linear Morse function $f : M \to \Re$ such that $f(p) = x$, $f(p) = y$ and $f(p) = z$ respectively in 4.4a, 4.4b and 4.4c.

As we noticed, the Reeb graph of the height function is not intrinsic to the surface since it will vary under isometric transformations, and also vary to rigid and non rigid transformations, not robust to topology change, and is very sensitive to noise and perturbations.

The important properties of our mapping function defined in previous chapter in section 3.4 and discussed in section 3.4.2 are that it is invariant to rigid and non rigid transformation, it describes the 3D model, it is insensitive to noise, it is robust to topology changes, it does not depend on parameters, and it is practical to compute on a discrete mesh. However, currently methods that are presented in the state of the art, also discussed in the previous chapter section 3.1, do not have all these basic properties.

In the discrete setting our mapping function is a piecewise linear function but not a Morse function. In order to compute the Reeb graph, we have to transform our mapping function to a piecewise linear Morse function. We used a perturbation strategy as defined by Tierny in (110) such that the mapping function (piecewise linear function) takes distinct values in every vertex of the mesh.

Let $V_s = v_0, v_1, ...v_i, ...v_n$ the set of vertices, where v_i is the ith vertex. We sorted V_s with regard to our mapping function, and added a constraint by considering the neighbours of a vertex (when two vertices or more have the same value of the mapping function I took the one in the neighbors of the previous selected). Then, we introduced the perturbation strategy induced by $f : S \to \Re$ as follows:

$$f(v_i) = \frac{i}{n} \tag{4.2}$$

with n number of vertices in V_s.

The perturbation strategy is kind of sorting the vertices and ensure that each vertex has a unique value. That will guarantee distinct values
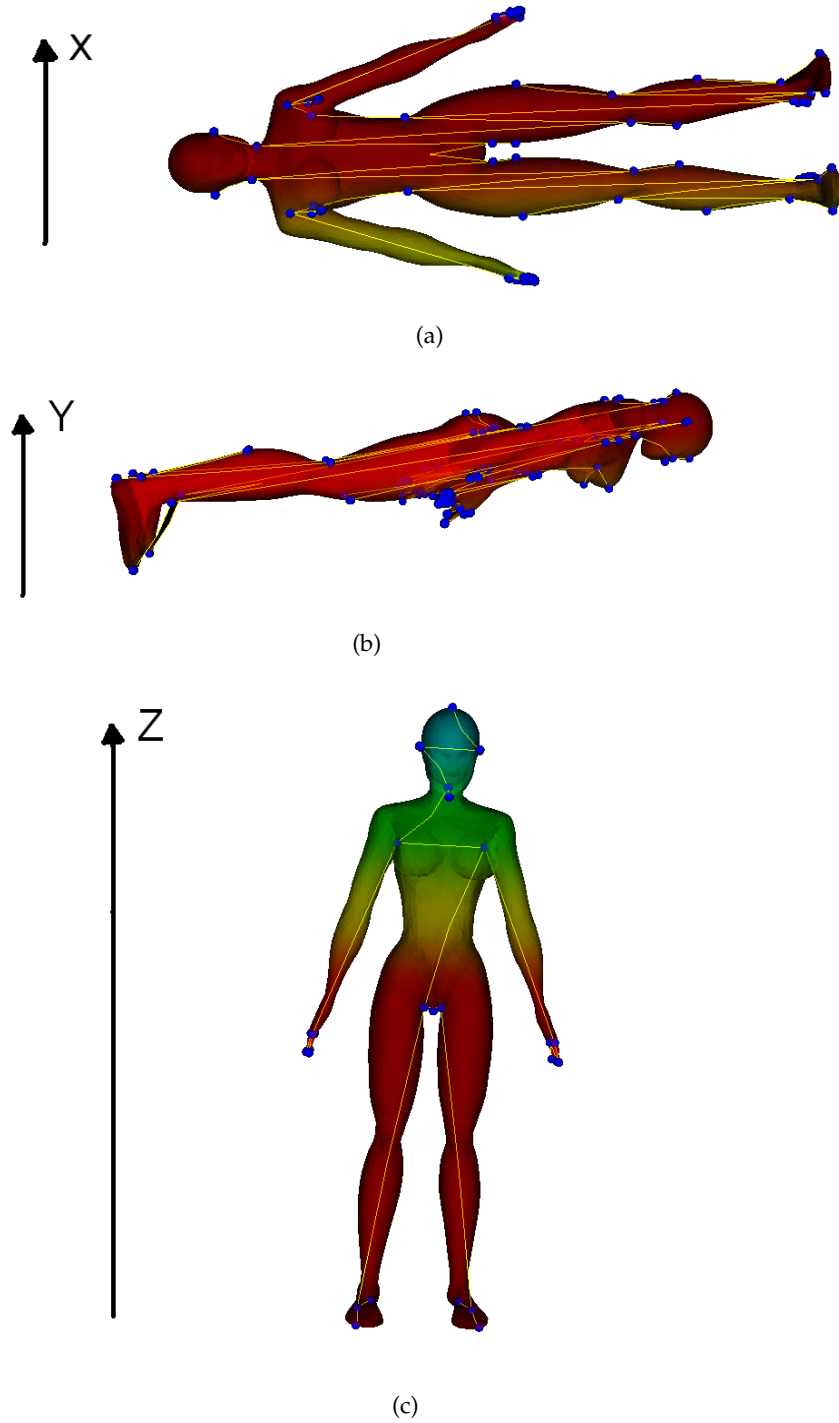
(a)

(b)

(c)

Figure 4.4 – *The reeb graph of the 3D human model based on the piecewise linear Morse function $f : M \to \Re$ such that $f(p) = x$ in 4.4a, $f(p) = y$ in 4.4b and $f(p) = z$ in 4.4c*

of the function on critical points and transform degenerate critical points into non degenerate ones (see figure 4.5).

The red node in figure 4.5a is a degenerated vertex because the level lines are splitted into more than two connected components and creates additional critical nodes. Indeed, the piecewise linear approximation of non-differentiable functions (non Morse function) leads to the appearance of additional critical vertices. Applying the peturbation strategy by sorting the vertices and starting from the minimum values of f at the red node the additional critical points will have bigger values thus the level lines at the red nodes do not split in more than two connected components. That will ensure the transformation of degenerated vertices to non-degenerated ones. The perturbation strategy transform the red node to a non-degenerated vertex, and thus the piecewise linear function to Morse function.

4.4 Reeb graph computation and topological analysis of the mesh

After the computation of our piecewise linear Morse function in the previous section 4.3 on the mesh, we use the defined function to generate level sets (79).

Lazarus and Verroust (58) consider the piecewise linear Morse function which maps a vertex to its geodesic distance to a user-selected source vertex. To get rid of the source vertex selection problem. Our mapping function detects the source vertex automatically. It computes for each vertex the commute-time distance to the nearest feature point. Consequently, the farthest vertex (where the Morse function is minimum) of all feature points will be detected and most of the time will be located in the center of the 3D model. This vertex will be the source vertex to generate the level sets.

These level sets are defined by the equivalence relation in section 4.2. We analyse them to define 4 types of topological changes:
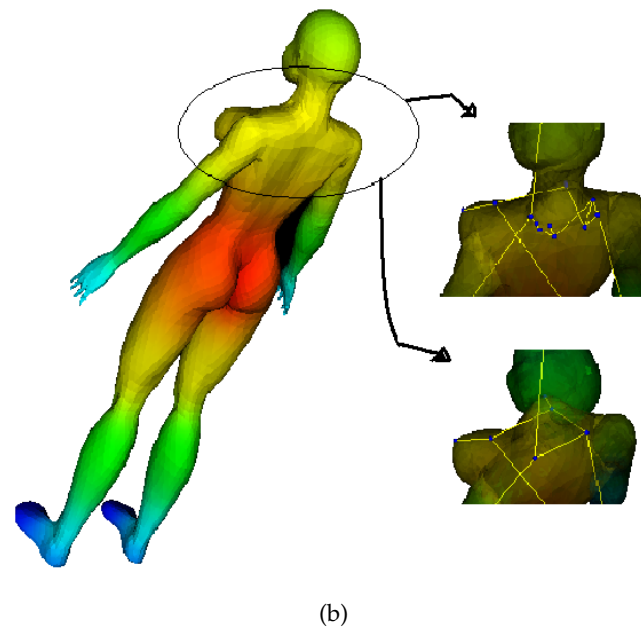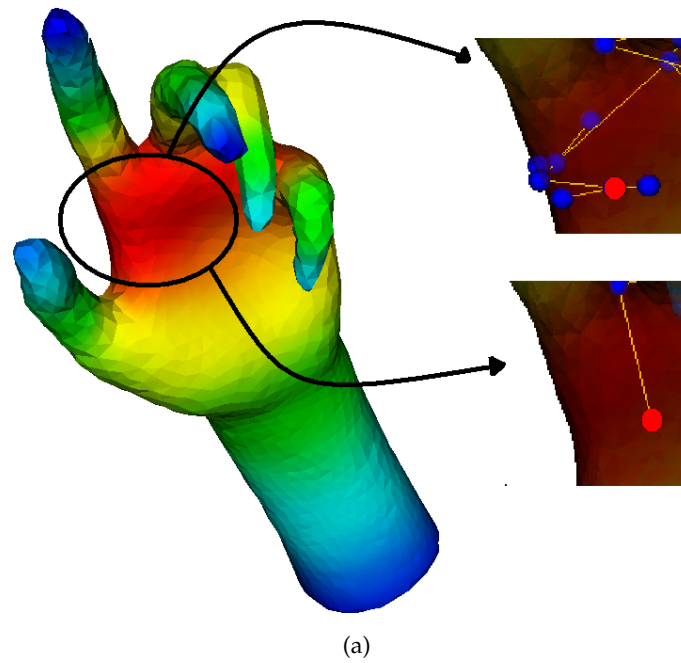
(a)

(b)

Figure 4.5 – *In each image in the figure the top right is the mapping function and bottom right is the Morse function after the perturbation strategy of the mapping function to transform critical vertices (the red node in figure 4.5a) into non degenerate ones.*

- The **start point** is located where the value of the Morse function is the minimum (center of the 3D-model).

- **Bifurcation** are detected where the level sets split.

- **Junction** are detected where the level sets merge.

- The **termination point**.

The critical points are created in each iteration where we detect a topological changes. These critical points and the connection between them correspond to the nodes and the edges of the Reeb graph.

Experiments and results of Reeb graphs on various 3D-models are generated using the Visualization Toolkit (VTK)[1]. The Visualization Toolkit implements the most robust Reeb graph computation algorithms developped by Pascucci *et al.* (79). Their method is very applicable and simple to implement. We add a perturbation strategy to ensure that all critical points are non degenerated.

4.5 Experiments and results

In this section, we evaluate our method on 3D-models presented as connected triangulated surfaces. In our experiments we focused on three points: First the robustness of our piecewise linear Morse function. Then, we extended our experiments and we computed the Reeb graphs on different types of 3D-models including partial and combined ones. Finally, we prove robustness of our approach to topology changes.

4.5.1 Experiments on our mapping function

Our piecewise linear Morse function expresses the robustness against variations in 3D-model pose. In Fig 4.6, the same colors have been affected to the same corresponding regions. Variation from red to blue colors expresses the increasing values of the Morse function. Our Morse function does not depend on a parameter to handle the level of details. It describes the local details and describes the global as well (see figure 4.7j). As an

---

[1]The Visualization Toolkit (VTK) `http://www.vtk.org/`.

(a) *Null 3D-model*   (b) *Partiallity*   (c) *sampling changed*

(d) *Null 3D-model*   (e) *Partiallity*   (f) *sampling changed*

(g) *scale*   (h) *Affine*   (i) *Isometric transformation*
    change   *transformation*

(j) *scale change*   (k) *Affine*   (l) *Isometric*
                           transformation   *transformation*
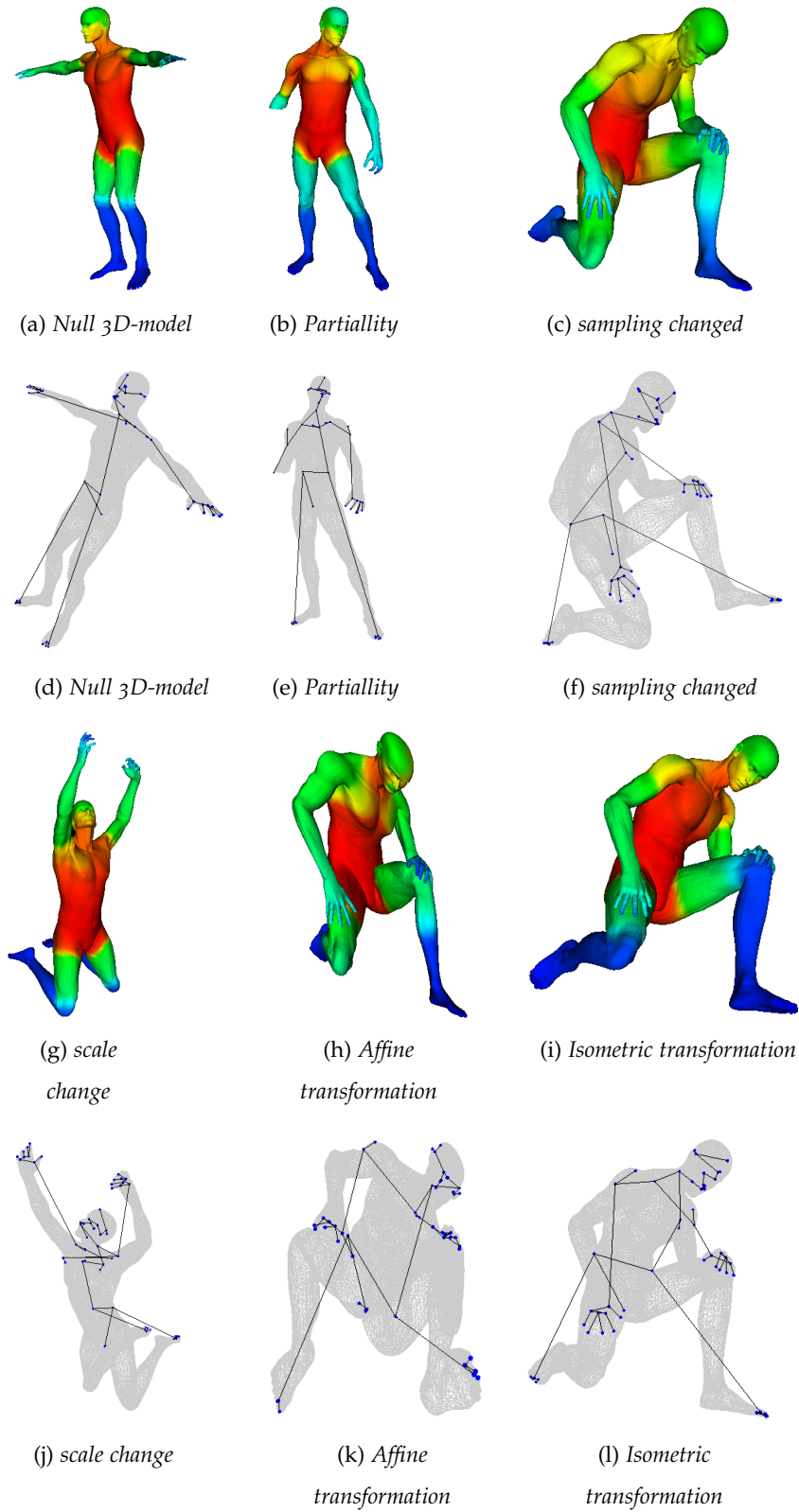
Figure 4.6 – *Images in the first and in the third rows show the robustness of the piecewise linear Morse function against variations in 3D-model pose, red to blue colors express the increasing values of the Morse function. Images in the second and the fourth rows show the corresponding Reeb graph. The number of vertices in all 3D-models are around 60000 vertices except 5.4a the number of vertices is reduced to 30000. 3D-models are taken from SHREC 2011 database.*

example, the small fingers and the back of the camel are detected. The autodiffusion function defined by Gebal *et al.* (39) depends on a parameter to handle the level of details. For a large variable time t, the function describes the global details of the 3D-model and misses the small ones. For a small variable time t, the autodiffusion function detects all details and moreover, it manages an important number considered as details that will lead to create unwanted branches in the Reeb graph.

Two reasons, as drawbacks of our method, lead to perturb the Reeb graph. First, the scalar value based on the mapping function on each vertex and its neighbours are very close or almost similar. These values depend on a variation around 10^{-20} and affect a bifurcation to change direction. In figure 4.8d the connection from the right breast to the right shoulder changed to the left shoulder. After the perturbation strategy, the Morse function takes into consideration the number of vertices to sort. This leads to detect bifurcation or junction in the construction of the Reeb graph, such as the chest of the human model in figure 4.6j.

4.5.2 Examples on different types of 3D-models

We tested our method on different types of 3D-models including 3D-models with missing parts or combined with others. Figure 4.6a and figure 4.6b show the robustness of our Morse function that leads to a stable Reeb graph. The Reeb graph of a partial 3D-model (figure 4.6e) includes the Reeb graph of its null 3D-model (figure 4.6d). Figure 4.7 shows examples of our method applied on 3D-models with rigid and non rigid transformations.

4.5.3 Robustness to topology changes

Figure 4.8 shows the robustness of the Morse function toward a small topology change. The colors remain the same after adding a small topology change linking the hump and the hand in figure 4.8c. This proves the invariance of our mapping function toward a topology change, due to the commute-time distance which took into consideration all paths connecting the vertices. If we apply a function based on the geodesic distance

(a)   (b)   (c)

(d)   (e)   (f)

(g)   (h)

(i)   (j)
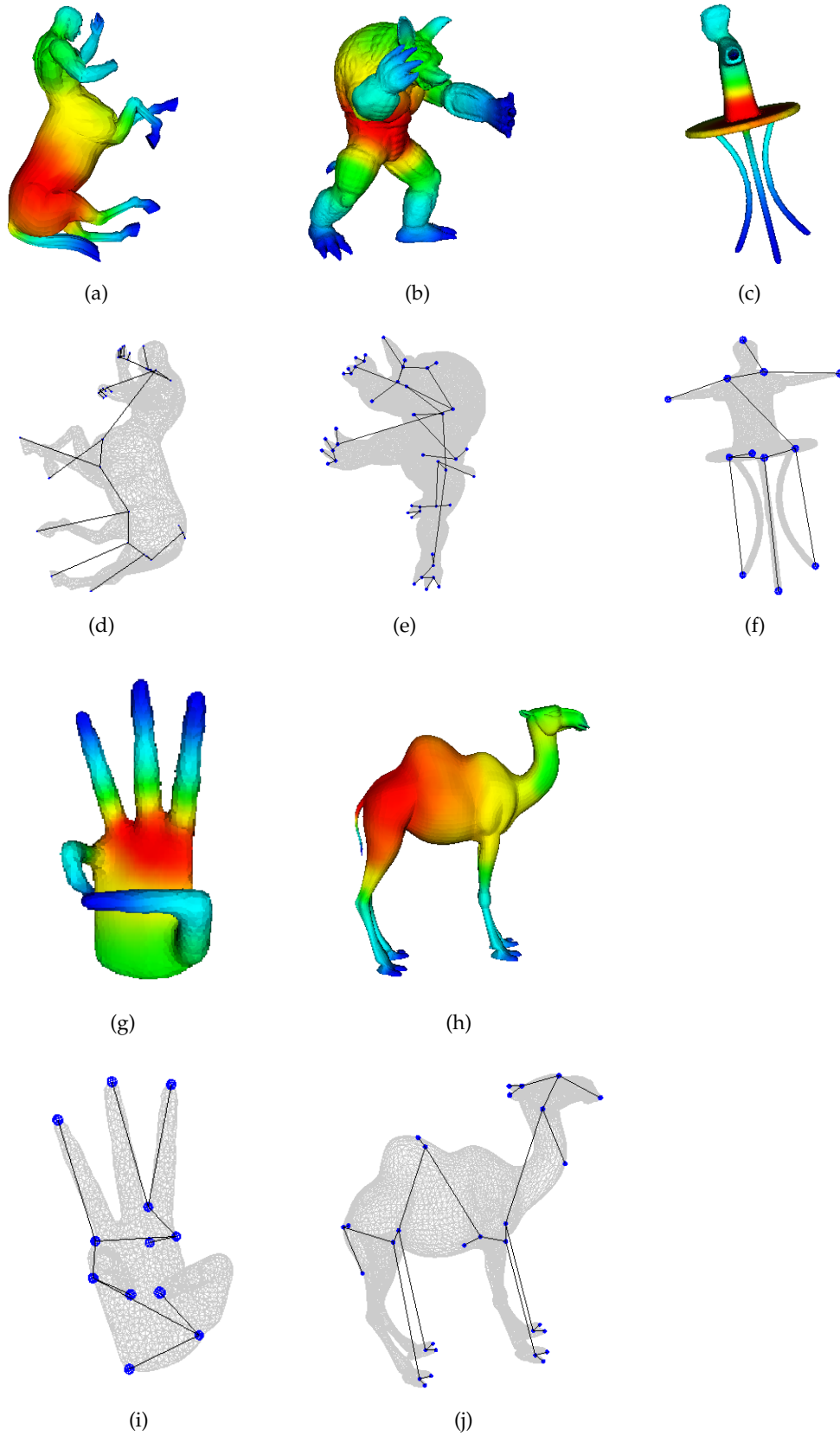
Figure 4.7 – *Reeb graph for different types of 3D-models.*

(a) *Null model*

(b) *Reeb graph*

(c) *Topology change*
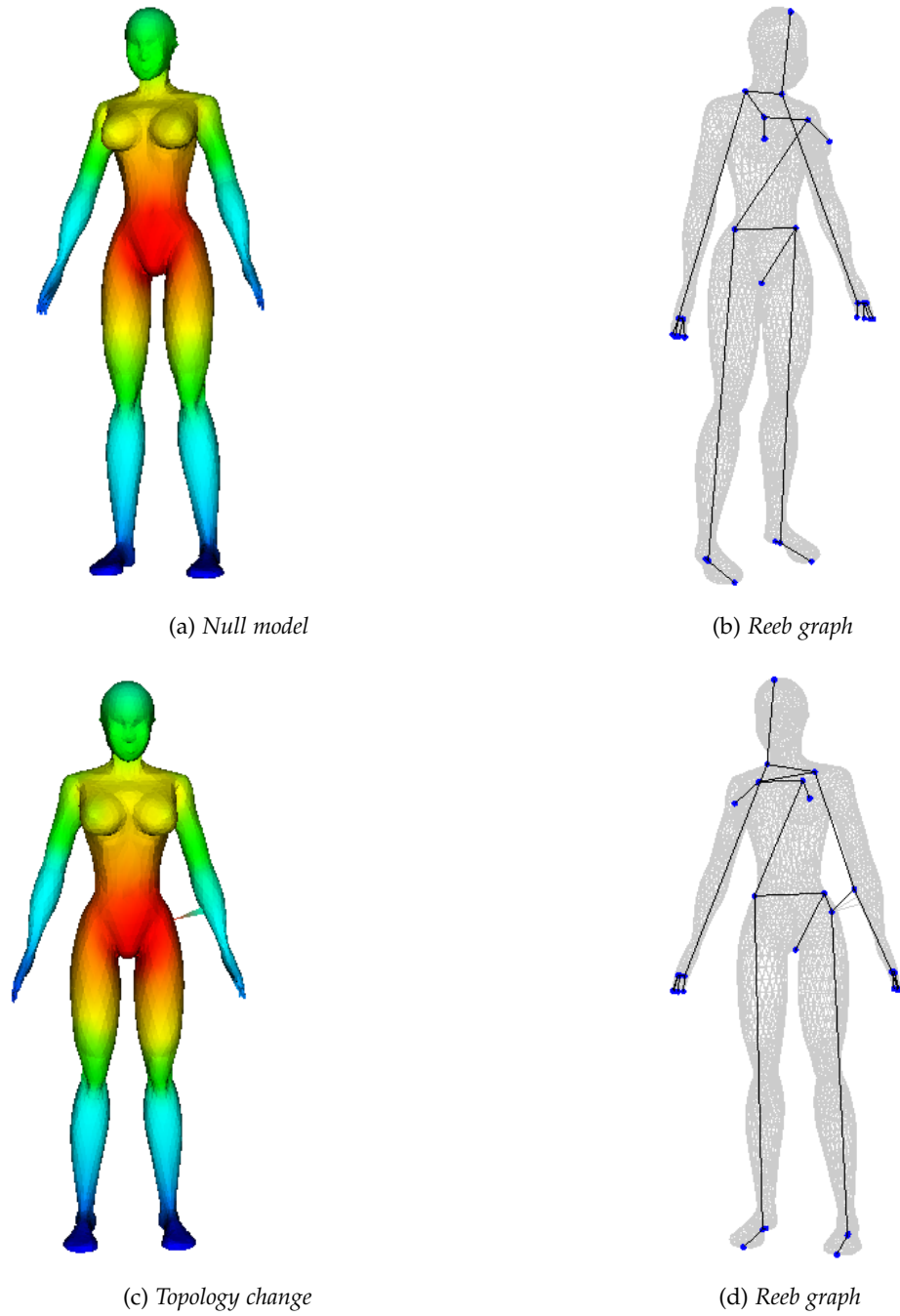
(d) *Reeb graph*

Figure 4.8 – *Robustness against topology changes.*

like the one defined by Tierny *et al.* (108), the Reeb graph will globally be changed. But using our Morse function, the Reeb graph undergoes the local change to describe the genus created toward the topology change.

4.6 Conclusion

To conclude, we start this chapter by introducing the Morse theory and the Reeb graph. Then, we presented a robust invariant piecewise linear Morse function for Reeb graph computation. Our Morse function is computed based on our mapping function defined in the previous chapter section3.4 by adding a perturbation strategy to transform it to a piecewise linear Morse function. This function inherits the properties from our mapping function such as the invariance to rigid and non rigid transformations, the insensitivity to noise and the robustness to small topology changes.

To compute the Reeb graph we need a Morse function and source vertex which will be used as a start point to generates level sets. Unlike the method defined by Lazarus and Verroust (58) where the source vertex is selected by the user, our approach detects the source vertex automatically, located at the centre of the 3D-model to generates level sets thanks to the definition of our mapping function. These level sets are analysed to describe the topology of the 3D-model.

We applied our method to a null 3D-model and a set of its variations in 3D-model pose. Also, we applied our method to different types of 3D-models including partial and combined ones and we showed the robustness toward a topology change. Our results show the effectiveness of our approach, consequently the effectiveness of our mapping function defined in the previous chapter. In the following chapter, we will introduce a novel method for 3D-model retrieval using indexed closed curves based on our mapping function.

INDEXED CLOSED CURVES FOR 3D MODEL RETREIVAL

<div style="text-align: right;">5</div>

CONTENTS

Tʜɪs chapter presents an approach for 3D-model retrieval by creating index of closed curves in $\Re^3$. Methods based on curves are more popular in face analysis topic (92, 46, 2) than 3D Shape analysis of generic 3D surfaces. A few works based on curves for 3D-model retrieval are presented in the literature, and none of them is very efficient.

Lmaati *et al.*(62) reconstruct 3D closed curves and extract feature vectors as a descriptor. This method needs to align the 3D-model into canonical position before the construction of the closed curves.

Tabia *et al.*(103) detect feature points located at the extremities of a 3D model. For each feature point, they generate a collection of closed curves based on the geodesic distance. Each feature point and its collection of closed curves represent a part of the 3D-model. Finally, they use the belief functions to define the global distance between 3D-models. This method is very sensitive to topology and a small variation of the feature point leads to a large variation in curves.

Our curves are generated from a source point detected automatically located at the center of a 3D-model, using our mapping function defined in chapter 3 in section 3.4. This function respects important properties in order to compute robust closed curves. Each curve describes a small region of the 3D-model. To describe all the mesh, we compute a set of indexed closed curves. These curves lead to creates a descriptor which is invariant to different transformations. Then we compute the distance between models by comparing the indexed curves.

In order to evaluate our method, we used shapes from SHREC 2012 database. The results show the robustness of our method on various classes of 3D-models with different positions. Also we compared our approach to existing methods in the state-of-the-art using a dataset for SHREC 2010 - Shape Retrieval Contest of Non-rigid 3D Models.

This chapter is organized as follows. In section 5.1 an overview of our method is given. Section 5.2 is about indexed closed curves generations, and the definition of our descriptor. In section 5.3 we analyze the extracted curves. In section 5.4 we impose the parameter setting of our approach. In section 5.5 we define the performance measures for 3D-model

retrieval. Before the conclusion in section 5.7, the experiments that prove the efficiency of our approach are explored in section 5.6.

Introduction (en français)

Ce chapitre présente une approche pour l'indexation des modèles 3D par la création de courbes de niveaux fermées dans $\Re^3$. Les méthodes basées sur les courbes sont plus populaires dans le domaine d'analyse faciale (92, 46, 2) que l'analyse de formes des modèles 3D. Quelques travaux basés sur les courbes pour l'indexation de modèles 3D sont présentés dans la littérature, et aucun d'entre eux est très efficace.

Lmaati *et al.*(62) reconstruisent des courbes fermées et extraient des vecteurs caractéristiques comme descripteur. Cette méthode doit aligner le modèle 3D en position canonique avant la construction des courbes fermées.

Tabia *et al.*(103) détectent les points caractéristiques situés aux extrémités d'un modèle 3D. Pour chaque point caractéristique, ils génèrent un ensemble de courbes fermées en fonction de la distance géodésique. Chaque point caractéristique et sa collection de courbes fermées représentent une partie du modèle 3D. Enfin, ils utilisent les fonctions de croyance pour définir la distance globale entre les modèles 3D. Cette méthode est très sensible aux changements topologiques et une petite variation du point caractéristique conduit à une grande variation dans les courbes.

Nos courbes sont générées à partir d'un point source détecté automatiquement au centre d'un modèle 3D, en utilisant notre fonction d'application définie dans le chapitre 3. Cette fonction respecte des propriétés importantes pour extraire des courbes de niveaux fermées et robustes. Chaque courbe décrit une petite région du modèle 3D. Pour décrire tout le modèle, on calcule un ensemble de courbes de niveaux fermées. Ces courbes conduisent à créer un descripteur invariant sous différentes transformations. Ensuite, nous calculons la distance entre les modèles en comparant les courbes de niveaux.

Afin d'évaluer notre méthode, nous avons utilisé des formes de la base de données SHREC 2012. Les résultats montrent la robustesse de notre

méthode aux différentes classes de modèles 3D avec des positions dif-
férentes. Aussi, nous avons comparé notre approche aux méthodes exis-
tantes dans l'état de l'art en utilisant la base de données de *SHREC* 2010
-Shape Retrieval Contest of Non-rigid 3D Models.

Ce chapitre est organisé comme suit. Dans la section 5.1 nous présen-
tons un simple résumé décrivant les étapes de notre approche. Dans la
section 5.2, nous extrayons les courbes de niveaux et nous définissons
notre descripteur. Puis, nous analysons les courbes extraites à la section
5.3. La section 5.4 présente l'implémentation de notre approche. Nous
continuons par la section 5.5 pour définir les mesures de performance de
l'indexation de modèles 3D. Avant la conclusion dans la section 5.7, les ex-
périences qui prouvent l'efficacité de notre approche sont explorées dans
la section 5.6.

5.1 METHOD OVERVIEW

As it is shown in figure 5.1, our method starts by detecting the feature points (figure 5.1a) which are used as origin to define an appropriate scalar function based on the commute time distance presented on figure 5.1b. The feature points extraction and the definition of the mapping function are discussed in chapter 3.

Then we generate and analyze indexed closed curves raised from a source point of the 3D-model using our mapping function function (figure 5.1c). These curves are defined as level curves. A set of all curves of each 3D-model are indexed to define our descriptor. We used Joshi *et al*'s method (49) to analyze and compute the elastic metric between curves. Finally, we analyze the 3D-model by analyzing the shape of their corresponding level curves.

## 5.2 DEFINITION OF THE DESCRIPTOR

The source vertex is detected automatically to generate indexed closed curves. Indeed, the farthest vertex of all feature points is detected by the minimum of the mapping function (figure 5.1b), the black arrow points to source vertex located in the center of the 3D model. From this vertex, we generate indexed closed curves under a scale value of the mapping function.

Due to the properties of the invariant mapping function defined on the mesh, this function describes 3D-models with different transformations similarly that leads to detect small region described by closed curves similarly. Each curve describes a small region as it is shown on figure 5.2. Consequently, the set of closed curves describes the 3D model entirely. Also, the indexing of the curves saves the spatial relation between small regions. Thus, figure 5.1c presents that same levels describe the same regions or almost the same regions as an example level 1 in the figure.

We analyze the 3D-model by analyzing the shape of their corresponding level curves. Finally, our descriptor is defined as a set of indexed closed curves. In order to compare two descriptors, we need to define
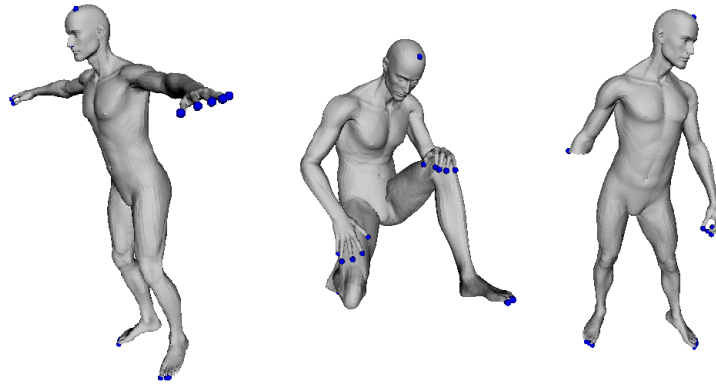
(a) *Feature points extraction.*
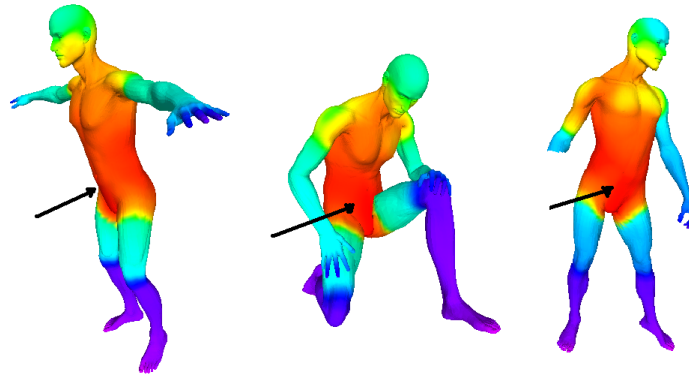
(b) *Mapping function, central point(black arrow).*
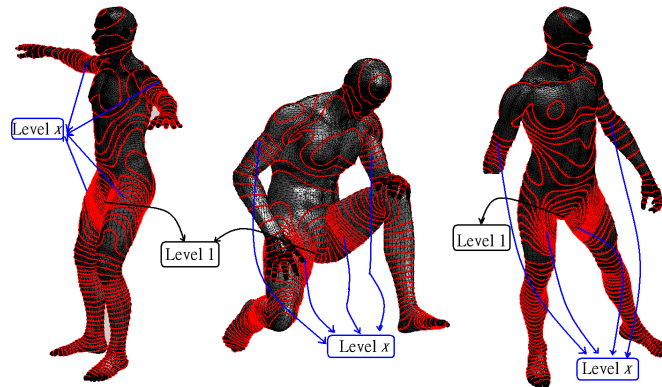
(c) *Indexed closed curves.*

(d) *Descriptor.*

Figure 5.1 – *The different steps of our approach applied to a neutral pose model and its isometric transformation, topology change and partiality.*
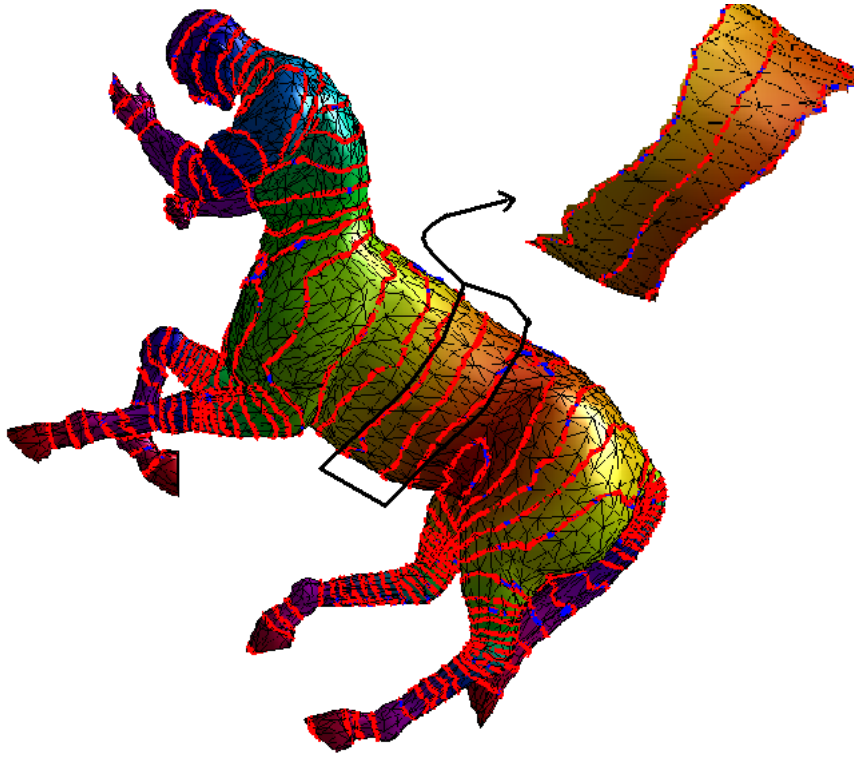
Figure 5.2 – *Each curve describe a small region.*

a similarity measure between two curves that analyzes the shape of the curve and respects the invariance to isometric transformation and to elastic deformation. The similarity measure between two curves is discussed in the next section.

5.3 ANALYSIS OF CURVES

In our approach, we treat curves as closed, parametrized in $\mathbb{R}^3$ with fixed origins for parametrization and we rescale them to have the same length, say 2π. This allows us to use one of many methods already available for elastic analysis of closed curves. The key idea in elastic analysis is that the points which are matched together are at unequal distances from their origins. Such matching can be considered as an elastic matching, which analyze the shape of the curves such as one curve has to (locally) stretch, compress and bend to match the other.

Several authors, starting with Younes (56), followed by Michor and Mumford (69) and others, have studied curves for planar shapes. More

recently Joshi *et al.* (49) have extended it to curves in $\mathbb{R}^n$ using an efficient representation of curves. Other authors, including Yezzi and Mennucci (118), have also used Riemannian metrics on curve spaces. Their main purpose was to study curves evolution rather than shape analysis. Here, we adopt the Joshi *et al.*'s approach (49) because it simplifies the elastic shape analysis. We summarize the whole method in the next 3 sections. For more details we refer the reader to Joshi *et al.*'s paper:

5.3.1 Define a space of closed curves of interest

The three following sections are taken from Joshi *et al.* (49). We start by considering a closed curve β in $\mathbb{R}^3$. Since it is a closed curve, it is parametrizable using $\beta : \mathbb{S}^1 \to \mathbb{R}^3$. We will assume that the parametrization is non-singular, i.e. $\|\dot{\beta}(t)\| \neq 0$ for all t. The norm used here is the Euclidean norm in $\mathbb{R}^3$. Note that the parametrization is not assumed to be arc-length; we allow a larger class of parametrization for improved analysis. To analyse the shape of β, we shall represent it mathematically using a *square-root velocity function* (SRVF), denoted by $q(t)$, according to:

$$q(t) \doteq \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|}} \tag{5.1}$$

$q(t)$ is a special function that captures the shape of β and is particularly convenient for shape analysis, as we describe next. Firstly, the squared $\mathbb{L}^2$-norm of q, given by:

$$\|q\|^2 = \int_{\mathbb{S}^1} \langle q(t), q(t) \rangle \, dt = \int_{\mathbb{S}^1} \|\dot{\beta}(t)\| dt \tag{5.2}$$

which is the length of β. Therefore, the $\mathbb{L}^2$-norm is convenient to analyze curves of specific lengths.

5.3.2 Impose a Riemannian structure on this space using the elastic metric

As shown in (49), the classical elastic metric for comparing shapes of curves becomes the $\mathbb{L}^2$-metric under the SRVF representation. This point is very important as it simplifies the calculus of elastic metric to the well-known calculus of functional analysis under the $\mathbb{L}^2$-metric. In order to

restrict our shape analysis to closed curves, we define the set:

$$\mathcal{C} = \{q : S^1 \to \mathbb{R}^3 | \int_{S^1} q(t)\|q(t)\|dt = 0\} \subset \mathbb{L}^2(S^1, \mathbb{R}^3) \qquad (5.3)$$

Here $\mathbb{L}^2(S^1, \mathbb{R}^3)$ denotes the set of all functions from S^1 to $\mathbb{R}^3$ that are square integrable. The quantity $\int_{S^1} q(t)\|q(t)\|dt$ denotes the total displacement in $\mathbb{R}^3$ as one traverses along the curve from start to end. Setting it equal to zero is equivalent to having a closed curve. Therefore, $\mathcal{C}$ is the set of all closed curves in $\mathbb{R}^3$, each represented by its SRVF. Notice that the elements of $\mathcal{C}$ are allowed to have different lengths. Due to a nonlinear (closure) constraint on its elements, $\mathcal{C}$ is a nonlinear manifold. We can make it a Riemannian manifold by using the metric: for any $u, v \in T_q(\mathcal{C})$, we define:

$$\langle u, v \rangle = \int_{S^1} \langle u(t), v(t) \rangle \, dt . \qquad (5.4)$$

We have used the same notation for the Riemannian metric on $\mathcal{C}$ and the Euclidean metric in $\mathbb{R}^3$ hoping that the difference is made clear by the context. For instance, the metric on the left side is in $\mathcal{C}$ while the metric inside the integral on the right side is in $\mathbb{R}^3$. For any $q \in \mathcal{C}$, the tangent space:

$$T_q(\mathcal{C}) = \{v : S^1 \to \mathbb{R}^3 | \langle v, w \rangle = 0, \ w \in N_q(\mathcal{C})\} , \qquad (5.5)$$

where $N_q(\mathcal{C})$, the space of normals at q is given by:

$$N_q(\mathcal{C}) = \text{span}\{\frac{q^1(t)}{\|q(t)\|}q(t) + \|q(t)\|\mathbf{e}^1, \frac{q^2(t)}{\|q(t)\|}q(t) + \|q(t)\|\mathbf{e}^2, \frac{q^3(t)}{\|q(t)\|}q(t) + \|q(t)\|\mathbf{e}^3\}$$
$$(5.6)$$

where $\{\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3\}$ form an orthonormal basis of $\mathbb{R}^3$.

It is easy to see that several elements of $\mathcal{C}$ can represent curves with the same shape. For example, if we rotate a curve in $\mathbb{R}^3$, we get a different SRVF but its shape remains unchanged. Another similar situation arises when a curve is re-parametrized; a re-parametrization changes the SRVF of curve but not its shape. In order to handle this variability, we define orbits of the rotation group $SO(3)$ and the re-parametrization group Γ as the equivalence classes in $\mathcal{C}$. Here, Γ is the set of all orientation-preserving

diffeomorphisms of S^1 (to itself) and the elements of Γ are viewed as re
parametrization functions. For example, for a curve $\beta : S^1 \to \mathbb{R}^3$ and a
function $\gamma : S^1 \to S^1$, $\gamma \in \Gamma$, the curve $\beta(\gamma)$ is a re-parametrization of β.
The corresponding SRVF changes according to $q(t) \mapsto \sqrt{\dot{\gamma}(t)}q(\gamma(t))$. We
set the elements of the set:

$$[q] = \{\sqrt{\dot{\gamma}(t)}Oq(\gamma(t))|O \in SO(3), \ \ \gamma \in \Gamma\} \tag{5.7}$$

to be equivalent from the perspective of shape analysis. The set of
such equivalence classes, denoted by $\mathcal{S} \doteq \mathcal{C}/(SO(3) \times \Gamma)$ is called the
shape space of closed curves in $\mathbb{R}^3$. $\mathcal{S}$ inherits a Riemannian metric from
the larger space $\mathcal{C}$ and is thus a Riemannian manifold itself. The main
ingredient in comparing and analyzing shapes of curves is the construc-
tion of a geodesic between any two elements of $\mathcal{S}$, under the Riemannian
metric given in Eqn. 5.4.

5.3.3   Compute geodesic paths under this metric

Given any two curves β_1 and β_2, represented by their SVRFs q_1 and q_2,
we want to compute a geodesic path between the orbits $[q_1]$ and $[q_2]$ in
the shape space $\mathcal{S}$. This task is accomplished using a *path straightening
approach* which was introduced in (54). The basic idea here is to connect
the two points $[q_1]$ and $[q_2]$ by an arbitrary initial path α and to iteratively
update this path using the negative gradient of an energy function

$$E[\alpha] = \frac{1}{2}\int_s \langle \dot{\alpha}(s), \dot{\alpha}(s) \rangle \, ds \tag{5.8}$$

The interesting part is that the gradient of E has been derived ana-
lytically and can be used directly for updating α. As shown in (54), the
critical points of E are actually geodesic paths in $\mathcal{S}$. Thus, this gradient-
based update leads to a feature point of E which, in turn, is a geodesic
path between the given points. We will use the notation $d(\beta_1, \beta_2)$ to de-
note the geodesic distance, or the length of the geodesic in $\mathcal{S}$, between
the two curves β_1 and β_2. Figure 5.3 is an example of geodesic path be-
tween two 3D-curves extracted from two different 3D surfaces; Figure 5.3a
illustrates the cow-head curve (left side) and the horse-head curve (right
side).

(a) *Geodesic path between cow-head curve and horse-head curve. (Figure taken from Tabia et al.(102)).*
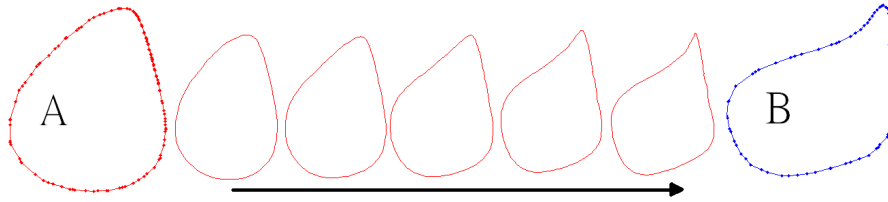
(b) *Geodesic paths between two curves (A,B).*

Figure 5.3 – *Geodesic paths between two curves.*

5.4 PARAMETER SETTING OF OUR APPROACH

The computation and the parameter settings of our mapping function are discussed in chapter 3. The farthest vertex of all feature points is detected by the minimum of the mapping function and it is defined as a source vertex. We generate closed curves under a scale value of the mapping function in an ascending order which are raised from the source vertex detected. We fixed the different scaled values of the mapping function to 50 levels. All levels are indexed starting from the first level where the mapping function is minimum and the region detected is very close to the source vertex.

In our experiments, while testing our approach two difficulties occurred:

1. Each model is described by 50 levels of closed curves in the database. A level could contain more than one curve as it shown on figure 5.1c,

where level x contained 4 curves. Matching two 3D-models, we need to match the corresponding curves in each level.

2. The mapping function have a slight variation between two similar models that leads to different levels of curves. Indeed, on figure 5.4 at level 15 the curve describes a small region at the knee of the 3D-model, while on figure 5.4a at the same level, the curve describes a small region near to the knee.
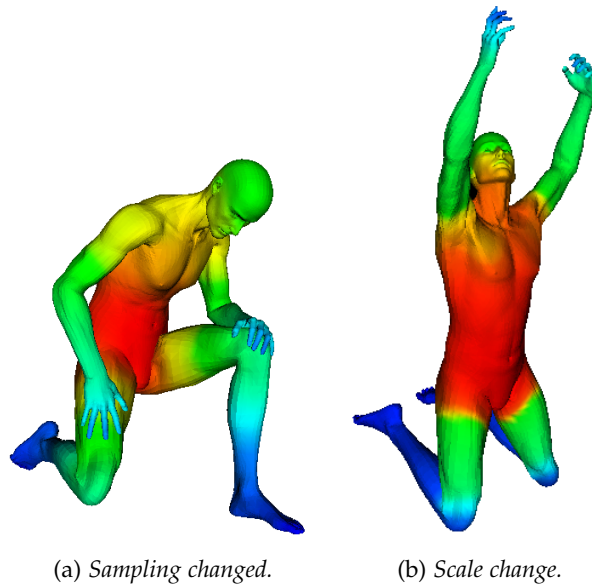

(a) *Sampling changed.*   (b) *Scale change.*

Figure 5.4 – *Red to blue colors express the increasing values of the mapping function. A slight variation between two similar models that leads to different levels of curves.*

To handle these two difficulties and match the corresponding curves in each model, we describe the query by 25 levels of closed curves. Each level of the query is compared to three levels of a model in the database. As an example, for a level l, this level is compared to $l - 1, l, l + 1$ levels with a model from the database (figure 5.5). Then, we take the one with the smallest length of their geodesic path (the most similar). Finally, to compute the similarity measure between two models, we compute the length of the geodesic path between their corresponding level curves.
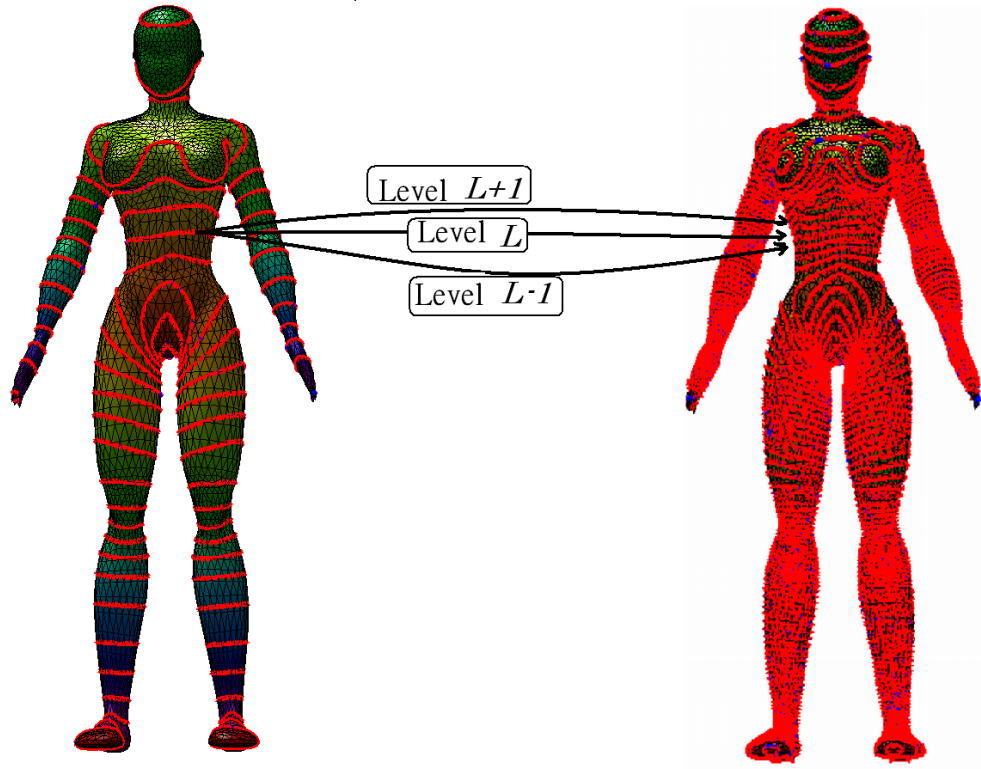
Figure 5.5 – *A level L in the query is compared to l − 1, l, l + 1 levels with a model from the database.*

5.5 EVALUATION CRITERION

There are several different performance measures to evaluate retrieval methods. In general, evaluation over the data set is performed by removing one model to act as the query, and ranking the remaining models from most similar to least similar. This ranked list can be evaluated in a different ways listed below. Performance for a particular method or set of parameters is given by averaging the performance over all query models. The evaluation measures are:

- Nearest Neighbour (NN), First Tier (FT) and Second Tier (ST). These evaluation measures share the similar idea, that is, to check the ratio of models in the query's class that also appear within the top K matches, where K can be 1, the size of the query's class, or the double size of the query's class. Specifically, for a class with $\|C\|$ members, $K = 1$ for Nearest Neighbor, $K = \|C\| − 1$ for the first tier, and

$K = 2(\|C\| - 1)$ for the second tier. The final score is an average over all the objects in database.

- Precision vs Recall plots. Precision is the ratio of retrieved objects that are relevant to all retrieved objects in the ranked list. Recall is the ratio of relevant objects retrieved in the ranked list to all relevant objects. They are well known in the literature of content-based retrieval. The Precision vs Recall are defined as follow:

$$Precision = \frac{N}{A} and Recall = \frac{N}{R} \tag{5.9}$$

where N is the number of relevant models retrieved in the top A retrievals. R is the number of relevant models in the collection, which is the number of models to which the query belongs to.

- E-measure (E) and F-measures (F) are a combined measure of the precision and recall for a fixed number of results (as an example the evaluation neighborhood size is 32, the candidates is more interested in the first page of query results than in later pages). The F-measure is the weighted harmonic mean of precision and recall and it is defined as:

$$F = \frac{(1 + \alpha) * P * R}{\alpha * P + R} \tag{5.10}$$

where precision (P), recall (R) and α is the weight. The weight of precision and recall in all 3D-retrieval methods evaluation is the same so $\alpha = 1$.

The E-measure (E) is defined as $E = 1 - F$

$$E = 1 - \frac{2 * P * R}{P + R} \tag{5.11}$$

The fact is that a user of a search engine is more interested in the first page of query results than in later pages.

## 5.6 EXPERIMENTS AND RESULTS

The experiments was to retrieve 3D-models from a large database containing similar models with different types of transformations. The proposed approach has been tested on two databases, the first one is to show the

effectiveness of our method, and the second database is to compare our approach to existing methods in the state-of-the-art.

5.6.1 Experiments to show the effectiveness of our method

(a) *One typical example for each category of the database.*
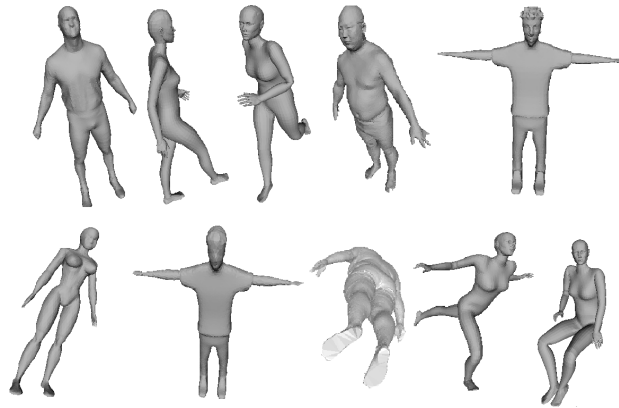
(b) *10 3D-models from human category.*

Figure 5.6 – .

First database is a part of SHREC 2012 - Sketch-Based 3D Shape Retrieval Dataset[1]. The collection we used consists of 130 3D-models classified into 13 categories (Tables, Sharks, Pliers, Planes, Octopus, Human, Hands, Glasses, Cups, Chairs, Bunnies, Birds and Ants). Each category contains 10 3D-models. Figure 5.6 shows the collection of the dataset we used. To evaluate our approach, we used the evaluation tool: the Precision vs Recall plot discussed in the previous section.

We plot the Precision vs Recall graph for the whole dataset (figure 5.7a) and also for each category (see figure 5.7). Our method shows very good results due to the invariant mapping function defined on the mesh. This function describes 3D-models with different transformations similarly that leads to detect small region described by closed curves similarly. We also

---

[1]http://www.itl.nist.gov/iad/vug/sharp/contest/2012/SBR/data.html

(a) *PR curve All Data*

(b) *PR curve Humans*

(c) *PR curve Cups*

(d) *PR curve Glasses*

(e) *PR curve Planes*

(f) *PR curve Ants*

(g) *PR curve Chairs*

(h) *PR curve Octopus*

(i) *PR curve Tables*

(j) *PR curve Teddies*

(k) *PR curve Hands*

(l) *PR curve Pliers*

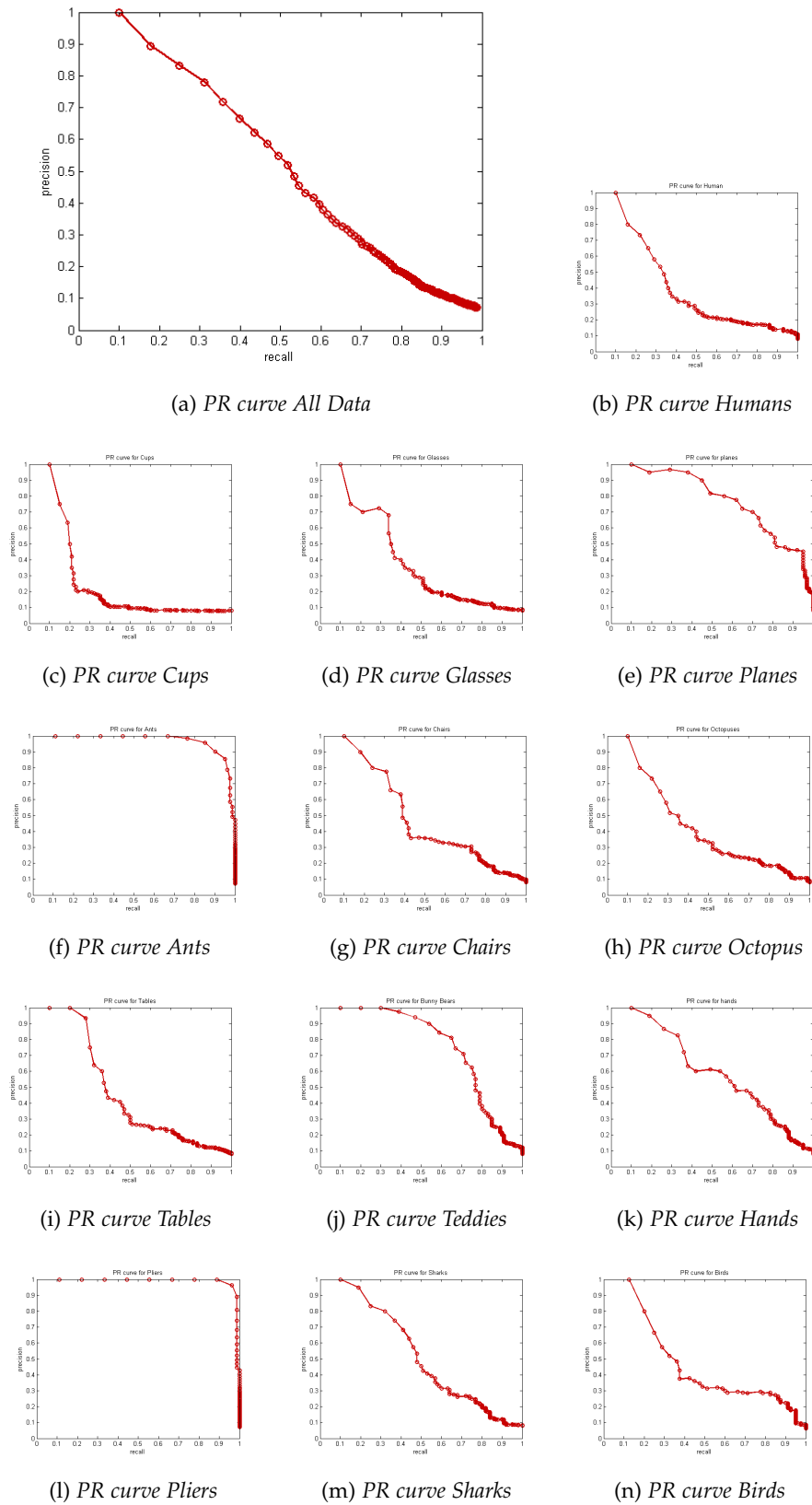(m) *PR curve Sharks*

(n) *PR curve Birds*

Figure 5.7 – *Precision vs Recall plot for each category and for the whole dataset.*

present the Nearest Neighbour (NN), First Tier (FT), Second Tier (ST) and the E-measure scores in table 5.1. The E-measure only considers the first 10 retrieved models for every query and calculates the Precision and recall values over those results since the user is more interested in the very first retrieved results than in the later ones.

Table 5.1 – *Retrieval statistics*

Class	NN	FT	ST	E-Measure
All classes	**0.81**	**0.57**	**0.71**	**0.59**
Humans	0.70	0.35	0.40	0.34
Cups	0.60	0.30	0.40	0.28
Glasses	0.50	0.41	0.52	0.41
Planes	1	0.78	0.94	0.76
Ants	1	0.87	1	0.80
Chairs	0.9	0.52	0.70	0.49
Octopuses	0.5	0.39	0.56	0.38
Tables	1	0.45	0.56	0.43
Teddies	1	0.68	0.85	0.67
Hands	0.9	0.64	0.78	0.63
Pliers	1	0.99	1	0.94
Sharks	0.9	0.52	0.66	0.51
Birds	0.5	0.45	0.79	0.39

These scores show the excellent results of our method for some classes like *Planes*, *Ants*, *Pliers*, *Hands*, *Teddies* but limited results for shapes as *Cups* which affect to decrease the score of the whole dataset. This is due to a few number of feature points detected in objects like cups. Also, we present samples of retrieved objects in figure 5.8. Hands results and the human model with topology changes used as a query show the robustness of our method toward topology change pointed by a red arrow in figure 5.8.
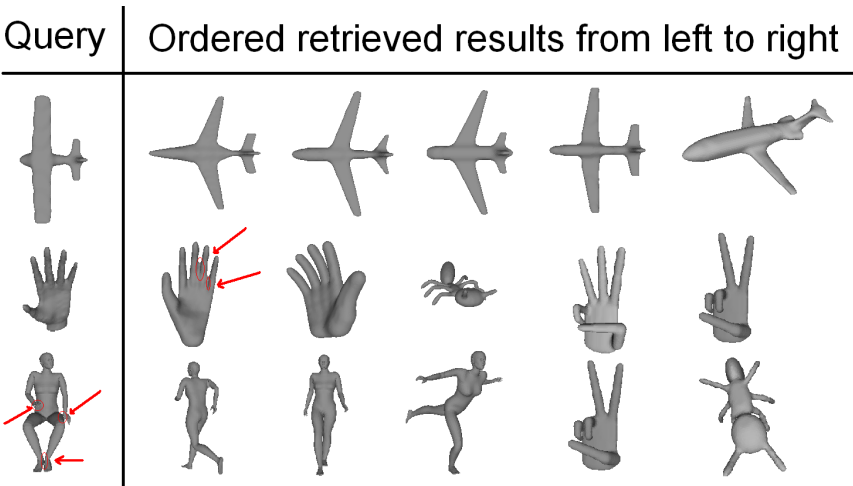
Figure 5.8 – *Example of retrieved results.*

5.6.2   Experiments compared to existing methods

To assess the efficiency of our approach, we have compared it to existing methods in the state-of-the-art and we used another database to evaluated its performance.
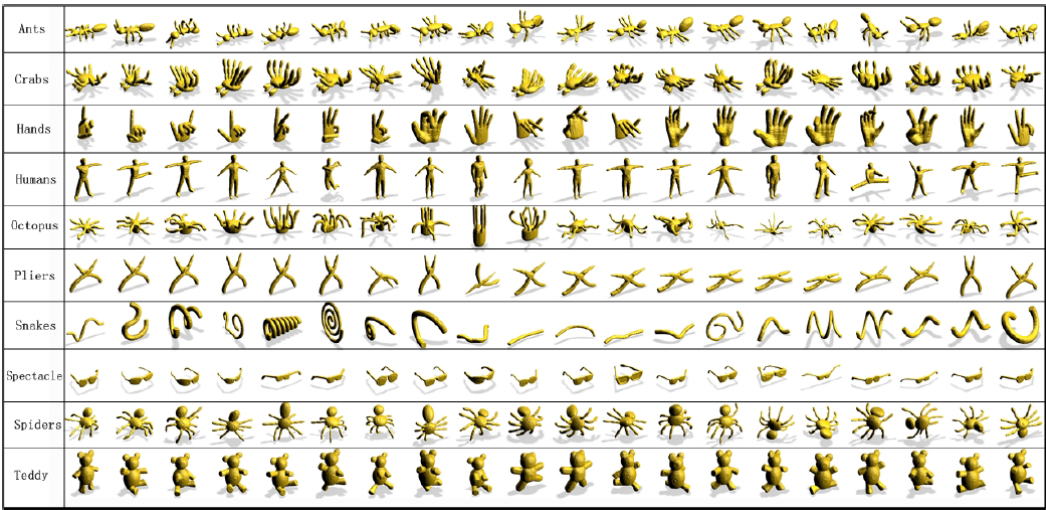

Figure 5.9 – *3D-Models of the database is classified into 10 categories. Each category contains 20 3D-models. Figure taken from (60).*

The second database is a dataset for SHREC 2010 - Shape Retrieval Contest of Non-rigid 3D Models[2]. It consists of 200 watertight 3D triangular meshes, which are selected from the McGill Articulated Shape

---

[2]http://www.itl.nist.gov/iad/vug/sharp/contest/2010/NonRigidShapes/index.html

Benchmark database[3]. It is equally classified into 10 categories based on their semantic meanings, as shown in figure 5.9.

We compared our results to three different existing methods in the state-of-the-art denoted as (Ohbuchi, Smeets and Wuhrer) and described as follows:

Furuya and Ohbuchi (35) present an approach to compare 3D-models based on their appearance. The authors rendered a set of depth images from multiple view points, and for each image they extracted local features based on the *Scale Invariant Feature Transform* (SIFT) algorithm introduces by Lowe (63). The set of all features of all 3D-models is integrated into a feature vector per 3D-model by using bag-of-features approach. The algorithm is called Bag-of- Features Dense-SIFT with Extremely randomized tree.

Smeets *et al.* (97) present an approach based on two invariant matrices for inelastic deformation invariant modal recognition. First, a pairwise matrix between points on the surface containing the average of diffusion distance (This distance is the probability that a particle, started in one point, arrives at the other point after a diffusion process ran for a certain time) for different diffusion times. Second one is pairwise matrix between points based on the geodesic distance (This distance is the length of the shortest path on the object surface between two points on the object). To handle the sampling order of points on the surfaces, the authors used the modal representation of these matrices by using a fixed number of the largest singular values of them.

Wuhrer *et al.*(117) compute the canonical form (the mapping of the surface to a target space) of a 3D-model using two steps: compute the canonical form at low resolution then adding the remaining vertices by minimizing the least squares energy function. Wuhrer and Shu's approach computes the dissimilarity of two models by computing the euclidean distance between their canonical forms.

Table 5.2 – *Retrieval statistics of our method.*

Class	NN	FT	ST	E-Measure
All classes	**0.8150**	**0.5440**	**0.7445**	**0.5160**
Ants	1	0.525	0.905	0.5923
Crabs	0.95	0.6825	0.8850	0.6442
Hands	1	0.6425	0.8125	0.5808
Humans	0.80	0.375	0.5275	0.3673
Octopuses	0.6	0.33	0.6025	0.3962
Pliers	1	0.8925	0.9675	0.7250
Snakes	0.1	0.25	0.5175	0.2904
Spectacles	1	0.7025	0.7875	0.5846
Spiders	0.7	0.445	0.705	0.4596
Teddies	1	0.59	0.735	0.5192

Results

We carry out evaluations not only on the average performance of the whole database, but also on the result corresponding to each specific class. The performance measures adopted here are the four quantitative statistics the Nearest Neighbour (NN), the First Tier (FT), the Second Tier (ST), the E-measure (E). We consider only the first 32 retrieved objects for every query and computes the E-Measure over those results. Even though, the number of elements of each class is 20 then the ideal results for the E-Measure is not equal to 1. Table 5.2 shows the (NN, FT, ST and E) for the whole database and for each class of our method.

Also we compute the Precision vs Recall plots for the whole database (see figure 5.10) and for each class (see figure 5.11) of all methods. The definition of these measures used to evaluate our method and to compare it with other are discussed in section 5.5.

As we noticed from table 5.3 where the performance measures are reported, all methods obtain good results but none of them obtain the ideal results. Considering the values of NN Smeets's method gives the

---

[3]http://http://www.cim.mcgill.ca/

best results, while Ohbuchi's method gives better results if we based the evaluation on $FT, STandE$.

If we take a close look on the Precision vs Recall plots of each class 5.11, we notice that the four methods have limited results on the snakes category. Especially, our method where the 3D-snakes models do not have a lot of information facing the other categories. Thus, decomposing the snakes into small region and describing these regions by closed curves as we have defined our descriptor, we see that all the curves have almost the same form and very similar to region described by closed curves for other 3D-models like the fingers of hands category or the legs of the crabs category.

This category affects enormously our results and decrease the score of the whole dataset. In the other hand, we noticed that our method has good results and very close to the ideal horizontal line for pliers category in figure 5.11. Generally speaking, a method which gives the ideal results do not exist yet. To conlude using this database, Ohbuchi's method and Smeets's method outperform our method.
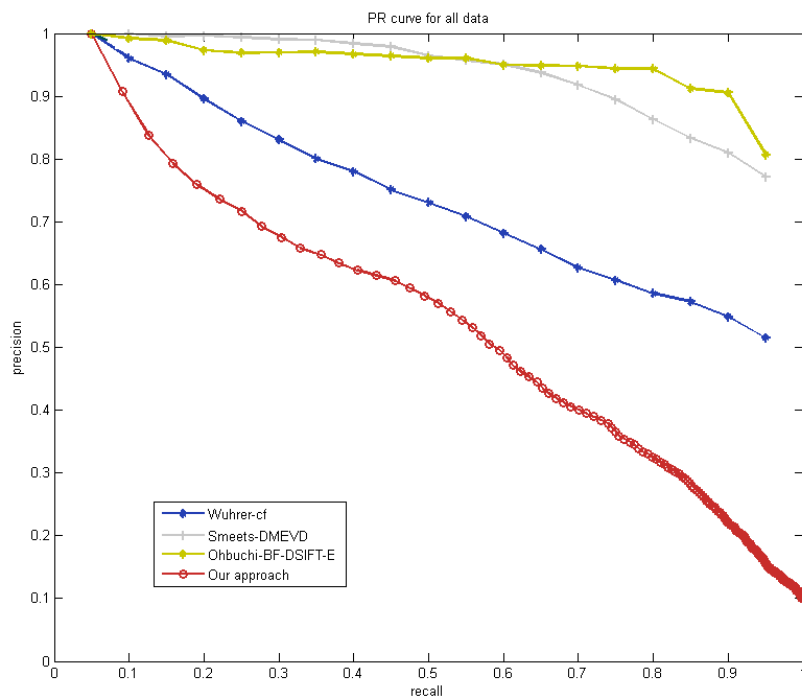

Figure 5.10 – *Precision vs Recall curves of all methods evaluated for the whole database.*

Table 5.3 – *Retrieval performance evaluated using four standard measures evaluated for the whole database.*

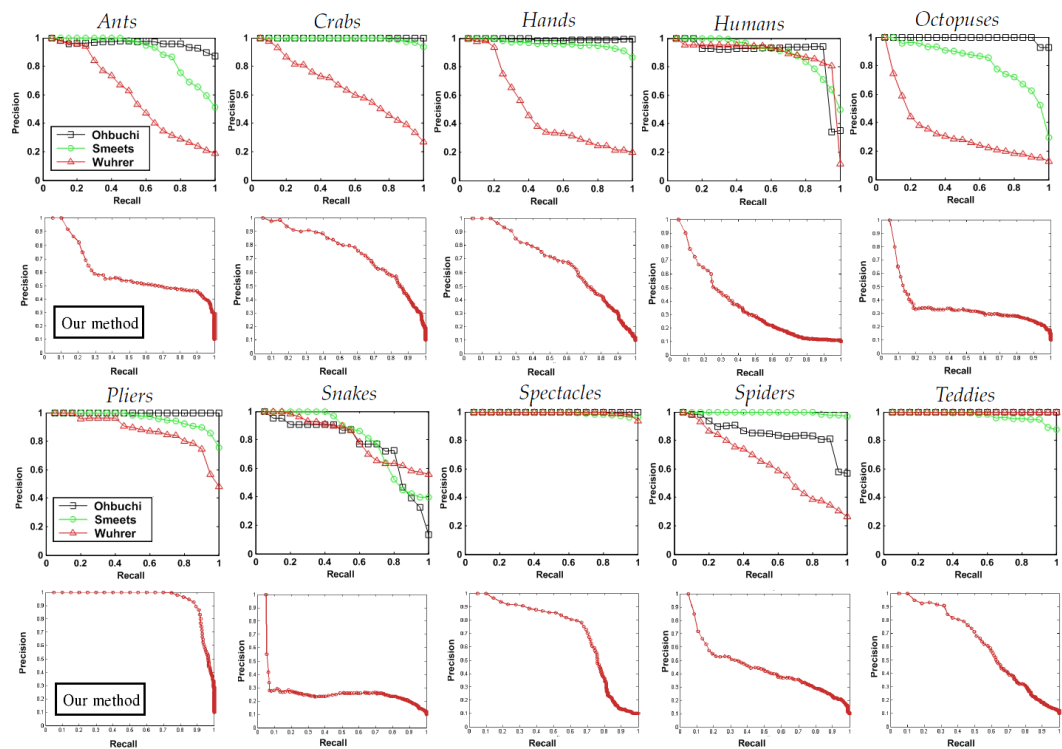| Partrticipant	NN	FT	ST	E-Measure
Ohbuchi, Ryutarou	0.9850	0.9092	0.9632	0.7055
Smeets, Dirk	1	0.8611	0.9571	0.7012
Wuhrer, Stefanie	0.9200	0.6347	0.7800	0.5527
Our method	**0.8150**	**0.5440**	**0.7445**	**0.5160**

Figure 5.11 – *Precision vs Recall curves of all methods evaluated for each class. The second and the fourth rows present the Precision vs Recall curves of our method.*

5.7 CONCLUSION

We presented in this chapter a novel approach for 3D-model retrieval. Our approach detect the center of the 3D-model automatically to generate closed curves describing the shape of the 3D-model. We generated indexed heat curves under different scale values of our invariant mapping function raised from the center of the 3D-model. Each curve describes a small region of the 3D-model. The set of all closed curves describes the whole 3D-model. We index all curves in order to save the spatial relationship between small region.

Finally we tested our approach on two datasets. We used the first dataset to evaluate our approach on different categories of 3D-models, each category contains 3D-models under different transformations and deformations. We used the second dataset to compare our approach to methods existed in the state-of-the-art and we discussed the results for 3-models retrieval.

In this chapter we realize that our experiments have very good results in some classes like (pliers and ants) and poor results in others like (cups). Also, our method has some limitation and do not handle partial 3D-model retrieval.

In the next chapter, we will present a novel method, which in turn is an enhancement of our indexed closed curves method by using the bag of feature technique. Also, our novel method that we will present handle partial 3D-model retrieval.

Bag of features for partial 3D-model retrieval

6

Tʜᴇ bag of features technique is a popular approach in areas of pattern recognition and computer vision. Recently, it plays an important role in shape analysis community and especially in 3D-model retrieval as we saw in the state of the art chapter in section 2.5.3.

Firstly, the bag of features framework has been proposed in the text retrieval and classification. Inspired by this approach, the bag of features techniques in 3D-models retrieval require to build a visual vocabulary which combines and merges local features into a global signature.

In the previous chapter, we generated a collection of closed curves from a source point detected automatically. Each curve describes a small region of the 3D-model. Based on the collection of all closed curves extracted, we construct our bag of features.

In this chapter, we discuss the main steps of the bag of features technique and we present our approach for 3d-model retrieval and also for partial 3d-model retrieval using this technique based on closed curves.

We organized this chapter as follows. In section 6.1 we discuss the main steps and the concept of the bag of features technique. Section 6.2 presents our approach by extracting and grouping local features described by closed curves to define our visual vocabulary. In section 6.3 we test our approach and compare our results to indexed closed curve approach presented in the previous chapter. Before concluding in section 6.5, the experiments that prove the efficiency of our approach towards partial 3d-model retrieval are explored in section 6.4.

Iɴᴛʀᴏᴅᴜᴄᴛɪᴏɴ (ᴇɴ ꜰʀᴀɴçᴀɪꜱ)

Les techniques *sacs de mots* sont des outils très populaires dans le domaine de la vision et de la reconnaissance des formes. Ces techniques ont récemment gagné une grande popularité dans la communauté de l'analyse de formes et surtout en indexation de modèles 3D, que nous avons vu dans le chapitre 2.

Tout d'abord, l'approche de *sacs de mots* a été proposé dans la classification et l'extraction de textes. Inspirée par cette approche, la technique *sacs de mots* pour l'indexation des modèles 3D a besoin d'élaborer un vo-

cabulaire visuel qui combine et fusionne les caractéristiques locales dans une signature globale.

Dans le chapitre précédent, nous avons généré une collection de courbes de niveaux fermées à partir d'un point source détecté automatiquement. Chaque courbe décrit une petite région du modèle 3D. Une fois la collection de toutes ces courbes fermées extraite, nous construisons nos *sacs de mots*.

Dans ce chapitre, nous introduisons les étapes principales de la technique de *sacs de mots* et nous présentons notre approche pour l'indexation des modèles 3D ainsi que l'indexation partielle de modèles 3D en utilisant cette technique basée sur les courbes fermées.

Nous avons organisé ce chapitre comme suit. Dans la section 6.1, nous discutons les étapes principales et le concept general de la technique de *sacs de mots*. La section 6.2 présente notre approche par l'extraction et le regroupement des caractéristiques locales décrites par des courbes fermées pour définir notre vocabulaire visuel.

Dans la section 6.3, nous testons notre approche et nous comparons nos résultats à l'approche présentée dans le chapitre précédent. Avant de conclure, dans la section 6.5, les expériences qui prouvent l'efficacité de notre approche sur l'indexation partielle de modèles 3D sont explorées dans la section 6.4.

6.1 Concept of bag of features technique

There is a common concept of the bag of features technique which is shared by all 3d-model retrieval existing methods in the state of the art. In summary, the concept is composed into four main steps (figure 6.1):
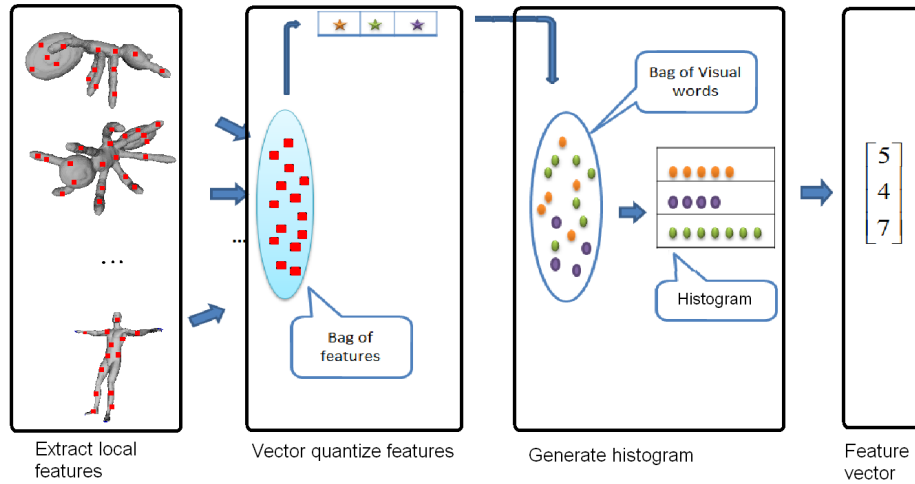

Figure 6.1 – *Bag-of-features technique.*

1. **Extraction of local features:** Local features extracted should be invariant to variation that are irrelevant to the retrieval task but also rich to carry enough information to be discriminative between dissimilar objects.

   There are different existing methods in the state of the art for local features extraction. Some authors use an object segmentation method (112). Other authors use a sampling method and select patches according to some geometric criterion (57), following successful approaches in 2D image recognition like SIFT (75). More recently, some authors propose the use of features points extraction algorithm for detecting points of interest around which they extract patches (104) for more details see chapter 2.

2. **Construction of the quantize features:** Local features are extracted and grouped. To quantify the local features extracted in the previous step, a common method is adapted by arranging the local features extracted into a finite number of clusters using a clustering techniques. The centers of the classes (*keywords* in text retrieval methods

or *keyshape* in 3d-model retrieval) obtained construct the quantize feature vector and determine the size of the vocabulary.

To distinguish relevant changes in 3d-model features, the size of the vocabulary plays an important role. It should not be too small, the details can not be distinguished, neither too large as to distinguish irrelevant variations such as noise. Most methods adapt the unsupervised learning for clustering .

3. **Generating the histogram:** A 3d-model query is represented by its *keyshapes* that are accumulated into an histogram having the number of the bins equal to the size of the vocabulary. The histogram becomes the feature vector of the corresponding 3d-model.

4. **feature vector and objects comparison:** The feature vector obtained in the previous step is a vector of occurrence counts of *keyshapes*. The frequency vector is normalized by the sum of occurrences of all terms of the considered vocabulary. 3d-models are compared by comparing their feature vectors.

## 6.2   Bag of features technique based on closed curves

Our approach for 3d model retrieval illustrated in figure 6.2 is developed following the main steps of the bag of features technique described in the previous section. We start by the training stage illustrated in figure 6.2a proceeding with the online stage illustrated in figure 6.2b.

6.2.1   Training stage

The training stage contains the first two steps of the bag of features technique: extraction of closed curves as local features then constructing our bag of features to generate the quantize feature vector.

Extraction of closed curves as local features

We need to extract local features describing the local region on the 3d-models. For each 3d-model in our database, we detect features points
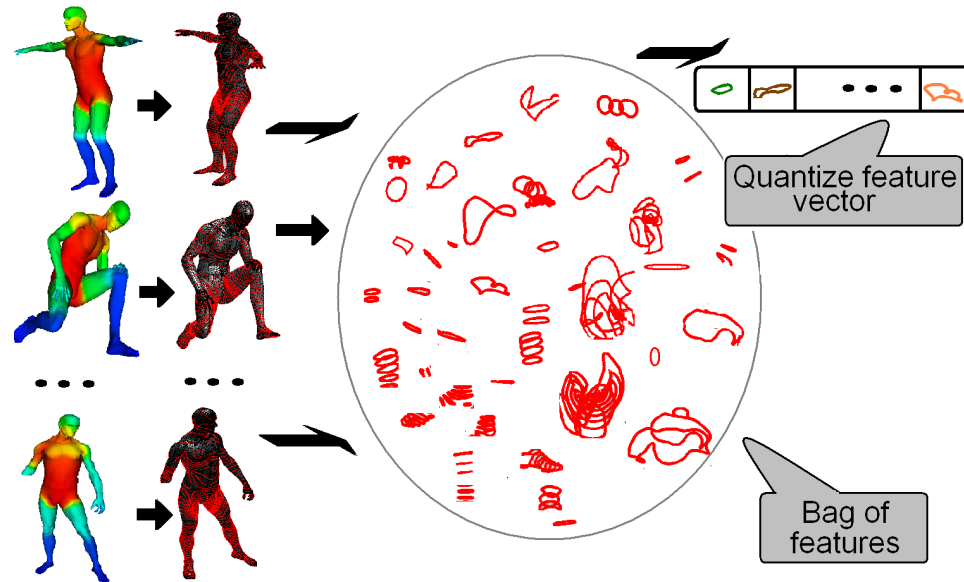
(a) *Training stage. From left to right: mapping function of the 3d models, extraction of closed curves, the collection of all curves to define the bag of features, quantize feature vector.*
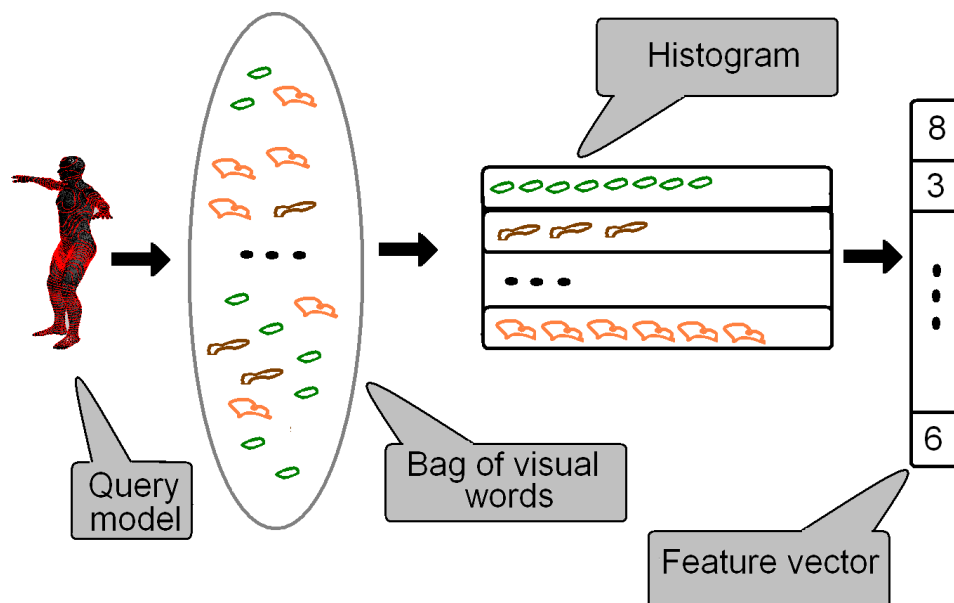
(b) *Online stage. Extraction of closed curves of a query 3d-model, the vector quantizes the local features into keyshapes that are accumulated into an histogram.*

Figure 6.2 – *The main steps of our bag of feature technique.*

located on the extremities. We used these features points as origins to define our scalar function based on the commute time distance presented on figure 6.2a. Detection of points located on the extremities and the definition of the scalar function are discussed in chapter 3.

We generate a collection of closed curves under a scale value of the scalar function from a source vertex detected automatically. Each model is described by 25 levels of closed curves in the database. A level could contain more than one curve (for example the average number of closed curves for each human 3d-model is 200). Each curve describes a small region of the 3d model. For more details on the extraction of the closed curves see section 5.1 in the previous chapter. These closed curves extracted are defined as local features.

Local features extracted should be invariant to rigid and non rigid transformations of the 3D-model. Indeed, for each 3D-model most of these local features are not affected to global variation even if a part of the 3D-model is missed (see figure 6.3).

Figure 6.3 – *A global variation even models with missing data in the 3D-models do not affect most of the closed curves extracted as local features.*

Generate the quantize feature vector

The collection of closed curves of all 3D-models are grouped to define our bag of features. To generate the quantize features and quantify the

local features extracted we need to cluster our bag of features. We analyse the curves extracted and define a distance between two curves for more details see section 5.3 in the previous chapter. The number of classes or the vocabulary should be large enough to distinguish relevant changes in local features but not too large to distinguish irrelevant variations such as noise. Thus, we choose the most widely used algorithms in computer vision community: *K means* (26) implemented in MatLab. Though, other methods K-medoids, histogram binning, etc. are certainly possible.

This algorithm is an iterative algorithm, in each iteration new cluster centres are computed and each data point is reassigned to its nearest centre. Two difficulties are that the K-means algorithm converges only to local optima of the squared distortion, and that it does not determine the parameter *K*. There exist methods allowing to automatically estimate the number of clusters. However, in our case we do not know the density or the compactness of our clusters. Moreover, we are not even interested in a correct clustering in the sense of features distributions, but rather in accurate categorization. We therefore run *K-means* several times with different number of desired representative vectors and different sets of initial cluster centres. We select the final clustering giving the best results. The centres of these classes are the *key-shapes*.

6.2.2 Online stage

The online stage is to categorize and describe the 3D-query by constructing the bag of words and generating the histogram to obtain the feature vector.

Constructing the bag of words

Having a query as a 3D-model, we extract the local features as closed curves. The objective here is to vector quantize the local features into *key-shapes* which will be the bag of words. Indeed, the bag of features is constructed from sub-parts of all 3D-models. Then, the 3D-model query is represented by its *keyshapes* that are accumulated into an histogram having

the number of the bins equal to the size of the vocabulary. The histogram becomes the feature vector of the corresponding 3D-model.

Computing the feature vector

The feature vector obtained defines the number of times a given *Keyshapes* appears in that 3D-model query. It is usual to normalize the feature vector obtained by the sum of occurrences of all terms of the considered 3D-model query. The feature vector will be used as a descriptor for the comparison of 3D-models.

6.3   EXPERIMENTS ON 3D-MODEL RETRIEVAL

In this section, we present the experimental results in order to evaluate our approach for 3D-model retrieval. For this purpose, our experiments were to retrieve 3D-models from a database containing different categories of 3D-models also similar 3D-models with different types of transformations.

The collection of 3D-models contained in our database are collected from the dataset for SHREC 2010 - Shape Retrieval Contest of Non-rigid 3D Models[1]. It consists of 25 watertight 3D triangular meshes, which are selected from the McGill Articulated Shape Benchmark database[2]. It is equally classified into 5 categories (ants, humans, pliers, snakes, teddies) based on their semantic meanings.

We use this database to perform our experiments. We evaluate our results using the evaluation tool: the Precision vs Recall plot, the Nearest Neighbour (NN), First Tier (FT), Second Tier (ST) and the E-measure scores. These measures are discussed in section 5.5 in the previous chapter. The E-measure only considers the first 10 retrieved models for every query. Also, we compare the results obtained by the bag of features technique based on closed curves to indexed closed curve method described in the previous chapter.

Figure 6.4 presents the Precision vs Recall curve for the whole dataset of the bag of features technique based on closed curves (figure 6.4a) and

---

[1]http://www.itl.nist.gov/iad/vug/sharp/contest/2010/NonRigidShapes/index.html
[2]http://http://www.cim.mcgill.ca/
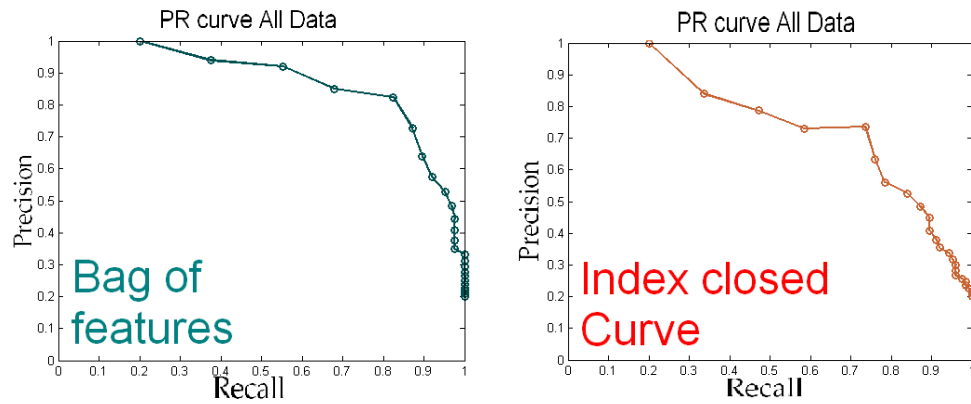
(a) *PR curve all data using the bag of features technique.*

(b) *PR curve all data using the indexed closed curves method.*

Figure 6.4 – *Precision vs Recall plot for each category and for the whole dataset.*

Table 6.1 – *Retrieval statistics using our approach*

Class	NN	FT	ST	E-Measure
All classes (Bag of features)	**0.8800**	**0.8240**	**0.9680**	**0.6453**
All classes (Indexed closed curves)	**0.76**	**0.7360**	**0.9440**	**0.6293**
Ants (Bag of features)	1	1	1	0.6667
Ants (Indexed closed curves)	0.8	0.72	1	0.6667
Humans (Bag of features)	0.8	0.68	1	0.6667
Humans(Indexed closed curves)	0.4	0.64	0.84	0.56
Pliers (Bag of features)	1	1	1	0.6667
Pliers (Indexed closed curves)	1	1	1	0.6667
Snakes (Bag of features)	1	1	1	0.6667
Snakes (Indexed closed curves)	0.6	0.6	1	0.6667
Teddies (Bag of features)	0.6	0.44	0.8	0.5333
Teddies(Indexed closed curves)	1	0.72	0.88	0.5867

of the method presented in the previous chapter (figure 6.4b). This figure clearly shows that the bag of features technique outperforms the indexed closed curves method on this database. Indeed, if we look to the Precision vs Recall curve for each class apart of the database presented in figure 6.5, we noticed that the bag of features have better results on most of the classes except for the class of teddies where the indexed closed curves method slightly outperforms the bag of features technique.

The Nearest Neighbour (NN), First Tier (FT), Second Tier (ST) and the E-measure scores of the two methods are presented in the table 6.1. These results show the effectiveness of our approach using the bag of features technique based on the closed curves.

This is due to the forms of the closed curves. Each curve describes a small region. The description of a small region is coded in the form of the curves. Two 3D-models look similarly, if the repartition of the form of their closed curves are similar.

Unlike, the indexed closed curves methods discussed in the previous chapter which matches two 3D-models by matching their corresponding curves of each level. Also, each level may have more than one curves. Thus, it's not easy to find their corresponding curves. This problem is solved by the definition of the bag of features technique itself. And this is shown in the results.
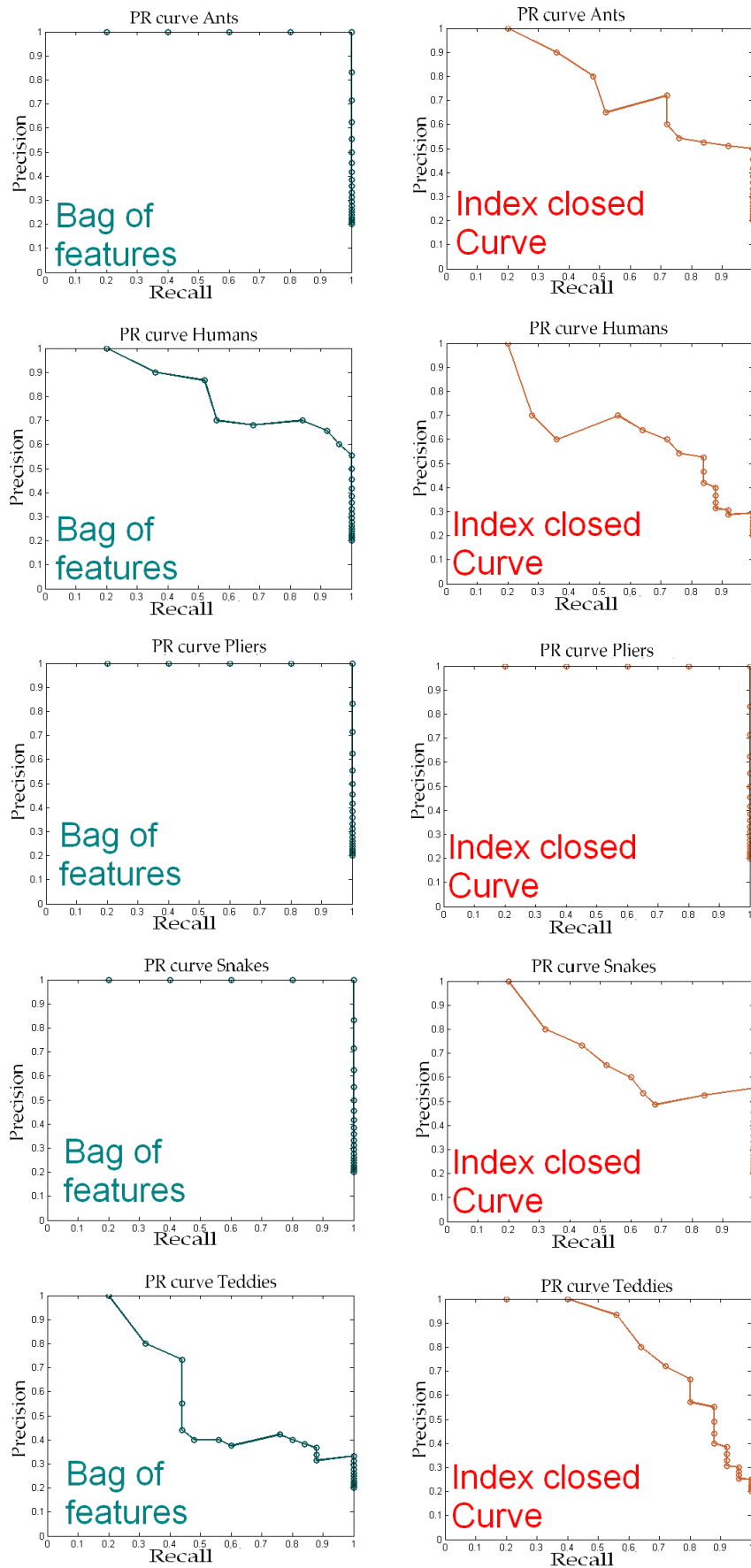
Figure 6.5 – *First column presents the Precision vs Recall plot of each category using*
our approach based on the bag of features technique.
Second column presents the Precision vs Recall plot of each category using our approach
based on the index closed curves.

6.4   EXPERIMENTS TOWARD PARTIAL 3D-MODEL RETRIEVAL

Partial 3D-model retrieval is a complex problem, which is facing two main difficulties. Matching partial or incomplete 3D-models and matching combined 3D-models.

Methods using global descriptors cannot describe partial 3D-models. Our approach based on closed curves which the form of each curve encodes the form of a small region on 3D-models. Then we group all these curves to construct our bag of features. Using the bag of features technique shows good results for partial 3D-models retrieval.

Indeed, figure 6.6 presents a 3D-model ant and a partial 3D-model ant which two of its legs are cut off. Figure 6.7 presents the mapping function which remain the same after two legs are cut off, which it leads to closed curves are generated from the same point and the forms of these curves remain the same.
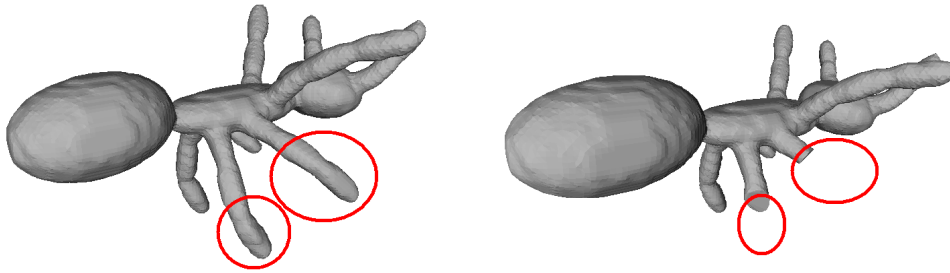
Figure 6.6 – *A 3D-model ant which two of their legs are cut off.*
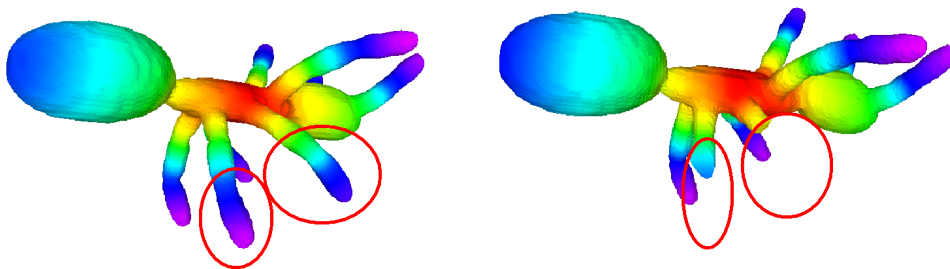
Figure 6.7 – *The mapping function of a 3D-model ant which two of their legs are cut off.*

Even if a part of a 3D-model is cut off including the source point, our approach will detect another source point in the centre of the object. Most of the closed curves generated from this new point will not be affected just

the first few ones. Since these curves are generated due to a scale value of the mapping function. The scale of the function will vary but the form of these curves remains almost the same. Because our mapping function based on the commute-time distance is defined on the evolution of the heat propagated in different time scale in the 3D-models by considering all paths connecting a pair of nodes. The heat propagates in the same manner in the same region and the form of these curves remains unchanged.

Also we tested our approach using a combined objects from our database as it shown in figure 6.8. As an example of retrieved results, figure 6.9 presents a partial and combined 3D-models used as a query and the top five retrieved returned by our algorithm. Also we added the partial 3D-model to the database and redo the experiments which leads to good results.
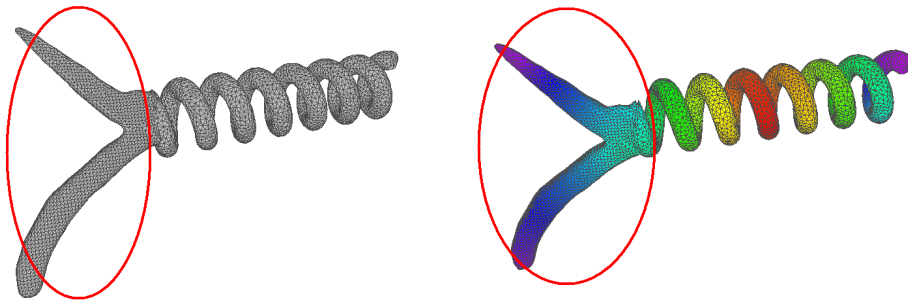
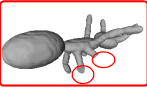Figure 6.8 – *Combined 3D-model (pliers and snake) and its mapping function.*

Figure 6.9 – *A partial 3D-model query retrieval results. First line to third line present partial 3D-models used as query. The last two lines partial 3D-models added to the database.*

6.5   CONCLUSION

In this chapter, we presented an approach of enhanced indexed closed curves to partial 3D-model retrieval using the bag of feature technique.

Our mapping function based on the commute-time distance is defined on the evolution of the heat propagated in different time scale in the 3D-models by considering all paths connecting a pair of nodes. This is done by summing the diffusion distance over the possible discrete time-steps on the graph.

A scale of our mapping function detects a region. The form of these region are encoded in the form of the closed curves. The heat propagates in the same manner in the same region and the form of these curves remain unchanged even if a part of the model is missed.

A local feature is a closed curve which describes a small region. The collection of the closed curves of all the database constitutes our bag of features. Clustering the bag of features in the sense in accurate categorization. The centres of classes which contains a form of similar small regions described by closed curves are defined as *keyshapes*. This method is particularly interesting in the sense of quantifying the 3D-model by its *keyshapes* that are accumulated into an histogram.

We demonstrated the effectiveness of our approach on two sets of experiments. First set of experiments are made to evaluate our approach for 3D-model retrieval and we compared our results to indexed closed curve approach defined in the previous chapter. The second set of experiments are made to validate our approach towards partial 3D-model retrieval.

To conclude, we introduced in this chapter a novel method for partial 3D-model retrieval based on bag of feature technique with very good results but not the ideal results. For future work we are going to investigate more in two parts of this technique: The first part by doing more experiments in different discretization of Laplace-Beltrami operator and the second part searching for a better method to cluster the bag of feature.

Conclusion

<div style="text-align: right; font-size: 3em;">7</div>

7.1 Summary

The rapid evolution of 3D graphics applications has created the need for 3D-model retrieval. Recently, researchers have investigated a lot of methods in this domain. We introduced the state-of-the-art for 3D-object retrieval and partial 3D-object retrieval. We discussed about the advantages and drawbacks of different techniques. Then we summarized them in a comparison table to allow the reader for a brief global view of the existing methods. Consequently, we concluded till present that an ideal approach for 3D-model retrieval and partial 3D-model retrieval does not exist.

In this thesis, we defined a novel invariant scalar function on the surface of the 3D-model based on the heat kernel. We started by developing a method to detect feature point located on the extremities that are used as origin to define this function which in turn it computes a real value for each vertex of the mesh and provides interesting insights to describe the topology structure and the geometry of the 3D-model. Also, it respects some important properties. It is invariant to rigid and non rigid transformations, it is insensitive to noise, it is robust to small topology changes, it does not depend on any parameters, and it is practical to compute on a discrete mesh. However, such a function that respects these basic properties do not exist in the state-of-the-art. Then we computed the Reeb graph to illustrate the stability and the invariance properties preserved by our function. Also, the topology graphs, such as Reeb graph is used in many applications like mesh segmentation, 3D-model retrieval, and so on. Then, we proposed two novel approaches for 3D-model retrieval and partial 3D-model retrieval using our scalar function mentioned above.

The first one is the indexed closed curve method for 3D-model re-
trieval. Our closed curves are generated from a source point detected
automatically at the centre of a 3D-model. For each scale of our scalar
function we detected a region. The form of these regions are encoded in
the form of the closed curves. All curves of each 3D-model are indexed
in correspondence to a scale value of our scalar function. The set of these
curves define our descriptor. We used Joshi *et al*'s method (49) to analyse
and compute the elastic metric between curves. Finally, we processed the
3D-model by analysing the shape of their corresponding level curves. Due
to the properties of the scalar function defined on the mesh, this function
describes 3D-models with different transformations similarly that leads to
detect small region described by closed curves similarly which made the
approach robust to isometric transformations as well as non rigid ones.
The second approach is an enhancement of the first method using the bag
of feature technique for partial 3D-model retrieval.

The first method has limited results on categories like snakes where
these 3D-models do not have a lot of information facing the other cate-
gories. Also, it is not very efficient for partial 3D-models retrieval. For
this purpose, we enhanced our approach by using the bag of feature tech-
nique. We validated our approach and tested on two sets of experiments.
First we compared our results to the previous method and the second we
validated it towards partial 3D-model retrieval. To conclude the results
show the effectiveness of our proposed method.

7.2 Open problems and directions

In this thesis, we presented new solutions for 3D models retrieval and
partial 3D model retrieval based on closed curve representation for 3D-
shape description. We believe that our solutions can be improved in three
points as follows:

Towards the definition of our scalar function. In the context of shape
similarity computing, we introduced two different methods: the indexed
closed curves approach and the bag of feature technique for partial 3D-

model retrieval. In the two methods, we are using our invariant scalar function based on Laplace-Beltrami operator. Marini *et al.* (66) investigate if the selection of a particular spectrum sequence is the best choice or if there exists other sequences of eigenvalues that provide better results. In my point of view, the computation of the spectrum depends on the mesh density. For both methods to be very accurate in the computation and the selection of a particular spectrum sequence an improvement can be done based on Marini *et al.* (66) investigations. Also, different authors proposed different discretization of Laplace-Beltramii operator. Recently, Reuter *et al.*(87) improve the discretization to a higher level of Laplace-Beltrami operator by adding extra nodes on each triangle in the mesh. We believe that using his discretization can improve our methods.

Towards shape analysis of parametrized surfaces. We proposed to analyse shapes of 3D-surfaces using indexed closed curves generated from a source point detected automatically based on our scalar function. In our opinion, this kind of representation could be improved to guarantee more robust results. In this sense, a pioneer work has been recently proposed by Kurtek et al. (55) where a novel Riemannian framework for shape analysis of parametrized surfaces is introduced. We are confident in the fact that such an approach can be integrated and automated in our framework.

Towards spatial information in the bag of feature technique. The disadvantage of bags of features is the fact that they consider only the distribution of the words and lose the relations between them. We believe that in the future, the integration of the spatial relation between feature could improve our approach for matching results. Computer vision techniques for graph matching (88) like Reeb graph can be applied to preserve the spatial relation between features in a given 3D-object.

Résumé (en français)

L'évolution rapide des applications 3D a créé le besoin pour l'indexation de modèles 3D. Récemment, les chercheurs ont fait des nombreuses

recherches concernant les problèmes liés à l'indexation des modèles 3D. Nous avons introduit l'état de l'art pour l'indexation de modèles 3D et l'indexation partielle de modèles 3D. Nous avons discuté les avantages et les inconvénients des différentes techniques. Puis, nous les avons résumées dans un tableau de comparaison pour permettre au lecteur une brève vue globale des méthodes existantes. Par conséquent, nous avons conclu jusqu'à présent qu'une approche idéale pour l'indexation et l'indexation partielle de modèles 3D n'existe pas.

Dans cette thèse, nous avons défini une nouvelle fonction scalaire appropriée qui respecte la stabilité et les propriétés d'invariance sur la surface du modèle 3D basée sur le noyau de la chaleur. Nous avons commencé à définir une méthode de détection de points caractéristiques situés aux extrémités du modèle 3D. Ces points caractéristiques sont utilisés comme origines pour définir notre fonction, qui à son tour, calcule une valeur réelle pour chaque sommet du maillage dans le but de décrire la structure topologique et la géométrie du modèle 3D. Cette dernière respecte les propriétés suivantes: elle est invariante aux transformations rigides et non rigides, elle décrit le modèle 3D, elle est insensible au bruit, elle est robuste aux changements topologiques, elle ne dépend pas de paramètres, elle est simple à calculer sur le maillage. Ensuite, nous avons calculé le graphe de Reeb pour illustrer la stabilité et les propriétés d'invariance conservées par notre fonction. En outre, les graphes de topologie, comme les graphes de Reeb, sont utilisés dans de nombreuses applications telles que la segmentation du maillage, l'indexation de modèles 3D, et ainsi de suite. Ensuite, nous avons proposé deux nouvelles approches pour l'indexation et l'indexation partielle de modèles 3D en utilisant notre fonction scalaire mentionnée ci-dessus.

La première approche pour l'indexation des modèles 3D est par la création de courbes de niveaux fermées dans $\Re^3$. Nos courbes sont générées à partir d'un point source détecté automatiquement au centre d'un modèle 3D. Pour chaque valeur de notre fonction scalaire, nous avons détecté une région. La forme de ces régions est encodée sous la forme de courbes fermées. Toutes les courbes de chaque modèle 3D sont indexées

en correspondance avec une valeur d'échelle de notre fonction scalaire. L'ensemble de ces courbes définissent notre descripteur. Nous avons utilisé la méthode proposée par Joshi *et al* (49) pour analyser et calculer la distance entre les courbes. Enfin, nous avons comparé deux modèles 3D par l'analyse de la forme de leurs courbes de niveau correspondantes. À cause des propriétés de la fonction scalaire définie sur le maillage, cette fonction décrit similairement les modèles 3D avec différentes transformations. Cette dernière mène à détecter similairement les petites régions décrites par des courbes fermées ce qui justifie la robustesse aux transformations rigides et non rigides. La deuxième approche est une amélioration de la première en utilisant la technique *sacs de mots* pour l'indexation partielle de modèles 3D.

La première méthode donne des résultats limités sur des catégories comme les serpents. En effet ces modèles 3D n'ont pas beaucoup d'informations par rapport aux autres catégories. En outre, cette méthode n'est pas très efficace pour l'indexation partielle de modèles 3D. À cette fin, nous avons amélioré notre approche en utilisant la technique *sacs de mots*. Nous avons validé notre approche et nous l'avons testée sur deux séries d'expériences. Premièrement, nous avons comparé nos résultats à la méthode précédente et deuxièmement nous avons validé notre approche pour l'indexation partielle de modèles 3D. Pour conclure, les résultats montrent l'efficacité de notre méthode.

Bibliography

[1] Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. Skeleton extraction by mesh contraction. In *ACM SIGGRAPH*, pages 44:1–44:10. ACM, 2008. (Cited page 29.)

[2] Lahoucine Ballihi, Boulbaba Ben Amor, Mohamed Daoudi, Anuj Srivastava, and Driss Aboutajdine. Selecting 3d curves on the nasal surface using adaboost for person authentication. In *3DOR'11*, pages 101–104, 2011. (Cited pages 88 and 89.)

[3] T. F. Banchoff. Critical points and curvature for embedded polyhedral surfaces. *The American Mathematical Monthly*, 77(5):pp. 475–485, 1970. (Cited page 73.)

[4] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15:1373–1396, June 2003. (Cited page 21.)

[5] K. Berthold and P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(2):1671–1686, 1984. (Cited page 23.)

[6] S. Biasotti, S. Marini, M. Mortara, and G. Patané. An overview on properties and efficacy of topological skeletons in shape modelling. In *Proceedings of the Shape Modeling International*. IEEE Computer Society, 2003. (Cited pages 68, 69, and 74.)

[7] Silvia Biasotti, Simone Marini, Michela Mortara, Giuseppe Patanè, Michela Spagnuolo, and Bianca Falcidieno. 3d shape matching through topological structures. In Ingela Nyström, Gabriella Sanniti di Baja, and Stina Svensson, editors, *Discrete Geometry for Computer Imagery*, volume 2886 of *Lecture Notes in Computer Science*, pages 194–203. Springer Berlin Heidelberg, 2003. (Cited page 29.)

[8] Silvia Biasotti, Simone Marini, Michela Spagnuolo, and Bianca Falcidieno. Sub-part correspondence by structural descriptors of 3d shapes. *Computer-Aided Design*, 38(9):1002 – 1019, 2006. (Cited pages 37 and 76.)

[9] Harry Blum. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, 1967. (Cited page 29.)

[10] P. T. Bremer, B. Hamann, H. Edelsbrunner, and V. Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10:385–396, July 2004. (Cited page 47.)

[11] Alexander M. Bronstein, Michael M. Bronstein, Leonidas J. Guibas, and Maks Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Trans. Graph.*, 30, February 2011. (Cited pages 22, 47, 52, and 55.)

[12] M.M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1704 –1711, june 2010. (Cited pages xi, 20, 22, and 23.)

[13] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić. Feature-based similarity search in 3d object databases. *ACM Comput. Surv.*, 37(4):345–387, December 2005. (Cited page 31.)

[14] N. Canterakis. 3d zernike moments and zernike affine invariants for 3d image analysis and recognition. In *In 11th Scandinavian Conference on Image Analysis*, pages 85–93, 1999. (Cited pages 2 and 8.)

[15] Marco Carcassoni and Edwin R. Hancock. Spectral correspondence for point pattern matching. *Pattern Recognition*, 36(1):193 – 204, 2003. (Cited page 20.)

[16] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung.

On visual similarity based 3d model retrieval. *Computer Graphics Forum*, 22(3):223–232, 2003. (Cited pages xi and 31.)

[17] Jen-Hi Chuang, Chi-Hao Tsai, and Min-Chi Ko. Skeletonization of three-dimensional object using generalized potential field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1241–1251, November 2000. (Cited page 36.)

[18] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5 – 30, 2006. (Cited page 52.)

[19] K. Cole McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Loops in reeb graphs of 2-manifolds. In *ACM Symposium on Computational Geometry*, pages 344–350, 2003. (Cited page 76.)

[20] Nicu D. Cornea, M. Fatih Demirci, Deborah Silver, Ali Shokoufandeh, Sven J. Dickinson, and Paul B. Kantor. 3d object retrieval using many-to-many matching of curve skeletons. In *Proceedings of the International Conference on Shape Modeling and Applications*, SMI '05, pages 368–373. IEEE Computer Society, 2005. (Cited page 36.)

[21] Nicu D. Cornea and Deborah Silver. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13:530–548, 2007. (Cited page 29.)

[22] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH, pages 317–324. ACM Press/Addison-Wesley Publishing Co., 1999. (Cited page 64.)

[23] Tamal K. Dey and Jian Sun. Defining and computing curve-skeletons with medial geodesic function. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP, pages 143–152. Eurographics Association, 2006. (Cited pages xi and 30.)

[24] T.K. Dey, K. Li, C. Luo, P. Ranjan, I. Safa, and Y. Wang. Persistent heat signature for pose-oblivious matching of incomplete models. *Computer Graphics Forum*, 29(5):1545–1554, 2010. (Cited page 35.)

[25] S. Dietmar and V-V Dejan. 3d model retrieval with spherical harmonics and moments. In *DAGM*, pages 392–397. Springer-Verlag, 2001. (Cited page 25.)

[26] O. Duda, P-E. Hart, and D-G Stork. *Pattern classification*. John Wiley and Sons, 2000. (Cited page 119.)

[27] Herbert Edelsbrunner. *Geometry and Topology for Mesh Generation.* 2001. (Cited page 71.)

[28] Herbert Edelsbrunner, John Harer, and Afra Zomorodian. Hierarchical morse complexes for piecewise linear 2-manifolds. In *Proceedings of the seventeenth annual symposium on Computational geometry*, SCG, pages 70–79. ACM, 2001. (Cited page 73.)

[29] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. In *Pattern Analysis and Machine Intelligence*, volume 25, pages 1285 – 1295. IEEE Transactions on Multimedia, oct. 2003. (Cited page 21.)

[30] Michael Elad, Ayellet Tal, and Sigal Ar. Content based retrieval of vrml objects: an iterative and interactive approach. In *Eurographics workshop on Multimedia*, pages 107–118. Springer-Verlag, 2002. (Cited pages 2 and 8.)

[31] T. Filali Ansary, M. Daoudi, and J-P. Vandeborre. A bayesian 3D search engine using adaptive views clustering. *IEEE Transactions on Multimedia*, 9(1):78–88, January 2007. (Cited page 32.)

[32] Anatoly T. Fomenko and Tosiyasu L.Kunii. *Topological Modeling for Visualization*. Springer, 1997. (Cited page 71.)

[33] T. Funkhouser and P. Shilane. Partial matching of 3d shapes with priority-driven search. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 131–142. Eurographics Association, 2006. (Cited page 34.)

[34] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David

Dobkin. Modeling by example. In *ACM SIGGRAPH*, pages 652–663. ACM, 2004. (Cited pages 35 and 36.)

[35] Takahiko Furuya and Ryutarou Ohbuchi. Dense sampling and fast encoding for 3d model retrieval using bag-of-visual features. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR, pages 26:1–26:8. ACM, 2009. (Cited page 105.)

[36] Nikhil Gagvani and Deborah Silver. Parameter-controlled volume thinning. *Graphical Models and Image Processing*, 61(3):149 – 164, 1999. (Cited page 29.)

[37] Ran Gal and Daniel Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.*, 25(1):130–150, January 2006. (Cited page 34.)

[38] Ran Gal, Ariel Shamir, and Daniel Cohen-Or. Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics*, 13:261–271, 2007. (Cited page 34.)

[39] K. Gebal, J-A. Baerentzen, H. Aanaes, and R. Larsen. Shape analysis using the auto diffusion function. In *Proceedings of the Symposium on Geometry Processing*, SGP, pages 1405–1413. Eurographics Association, 2009. (Cited pages 47, 49, 55, 63, and 83.)

[40] W. Gong and G. Bertrand. A simple parallel 3d thinning algorithm. In *10th International Conference on Pattern Recognition.Proceedings.*, volume i, pages 188 –190 vol.1, jun 1990. (Cited page 29.)

[41] Alexander Grigor'yan. Estimates of heat kernels on riemannian manifolds. *Spectral Theory and Geometry. ICMS Instructional Conference, Edinburgh, 1998*, 273:140–225, 1999. (Cited page 51.)

[42] John C. Hart. Morse theory for implicit surface modeling. In *Mathematical Visualization*, pages 257–268. Springer-Verlag, 1997. (Cited pages 71 and 72.)

[43] D-M. Healy, D-N. Rockmore, P-J. Kostelec, and S. Moore. Ffts for

the 2-sphere-improvements and variations. *Journal of Fourier Analysis and Applications*, 9(4):341–385, July 2003. (Cited page 25.)

[44] M. Heczko, D. Keim, D. Saupe, and D. Vranic. Methods for similarity search on 3d databases. *Datenbank-Spektrum (in German)*, 2(2):54–63, 2002. (Cited page 31.)

[45] Daniel Huber, Anuj Kapuria, Raghavendra Donamukkala, and Martial Hebert. Parts-based 3d object classification. In *conference on Computer vision and pattern recognition*, CVPR, pages 82–89. IEEE Computer Society, 2004. (Cited page 32.)

[46] S. Jahanbin, Hyohoon Choi, Yang Liu, and A.C. Bovik. Three dimensional face recognition using iso-geodesic and iso-depth curves. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1 –6, 2008. (Cited pages 88 and 89.)

[47] Varun Jain and Hao Zhang. Robust 3d shape correspondence in the spectral domain. In *IEEE International Conference on Shape Modeling and Applications*, 2006. (Cited page 20.)

[48] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, May 1999. (Cited page 34.)

[49] S.H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn. Removing shape-preserving transformations in square-root elastic (sre) framework for shape analysis of curves. In *EMMCVPR*, pages 387–398, 2007. (Cited pages 91, 94, 130, and 133.)

[50] S-B. Kang and K. Ikeuchi. The complex egi: a new representation for 3-d pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):707 –721, July 1993. (Cited page 23.)

[51] Sagi Katz, George Leifman, and Ayellet Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):649–658, 2005. (Cited pages 54 and 55.)

[52] M. Kazhdan and T. Funkhouser. Harmonic 3d shape matching. In *ACM SIGGRAPH conference abstracts and applications*, pages 191–191. ACM, 2002. (Cited pages xi and 25.)

[53] Yakov Keselman, Ali Shokoufandeh, M. Fatih Demirci, and Sven Dickinson. Many-to-many graph matching via metric embedding. In *IEEE Conference In Computer Vision and Pattern Recognition),vol.1*, pages 850–857, 2003. (Cited page 36.)

[54] E. Klassen and A. Srivastava. Geodesics between 3d closed curves using path-straightening. In *European Conference on Computer Vision (ECCV)*, pages 95–106, 2006. (Cited page 96.)

[55] S. Kurtek, E. Klassen, Zhaohua Ding, and A. Srivastava. A novel riemannian framework for shape analysis of 3d objects. In *IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*, pages 1625 –1632, june 2010. (Cited page 131.)

[56] Y. Laurent. Computable elastic distance between shapes. *SIAM Journal of Applied Mathematics*, 58(2):565–586, 1998. (Cited page 93.)

[57] Guillaume Lavoué. Bag of words and local spectral descriptor for 3d partial shape retrieval. In Eurographics, editor, *Eurographics Workshop on 3D Object Retrieval (3DOR)*, May 2011. (Cited pages 27, 35, and 115.)

[58] Francis Lazarus and Anne Verroust. Level set diagrams of polyhedral objects. In *Proceedings of the fifth ACM symposium on Solid modeling and applications*, SMA, pages 130–140. ACM, 1999. (Cited pages 47, 49, 79, and 86.)

[59] Bruno Levy. Laplace-beltrami eigenfunctions: Towards an algorithm that understands geometry. In *IEEE International Conference on Shape Modeling and Applications, invited talk*, 2006. (Cited pages 21 and 52.)

[60] Z. Lian, A. Godil, Thomas Fabry, T. Furuya, Jeroen Hermans, R. Ohbuchi, C. Shu, Dirk Smeets, Dirk Suetens, Paul andVandermeulen, and S. Wuhrer. Shrec 10 track: non-rigid 3d shape retrieval.

Norrköping, Sweden, 2 May 2010. Eurographics workshop on 3D object retrieval - 3DOR. (Cited page 104.)

[61] Yi Liu, Hongbin Zha, and Hong Qin. Shape topics: A compact representation and new algorithms for 3d partial shape retrieval. In *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. (Cited pages 34 and 37.)

[62] Elmustapha Ait Lmaati, Ahmed El Oirrak, and M. N. Kaddioui. A 3d search engine based on 3d curve analysis. *Signal, Image and Video Processing*, pages 89–98, 2010. (Cited pages 24, 88, and 89.)

[63] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. (Cited pages 24 and 105.)

[64] M. Mahmoudi and G. Sapiro. Three-dimensional point cloud recognition via distributions of geometric distances. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. CVPRW.*, pages 1 –8, june 2008. (Cited pages 20 and 21.)

[65] Manju Mandot and K. Venugopalan. A survey of surface retrieval techniques for 3d models. In *Proceedings of the 2008 First International Conference on Emerging Trends in Engineering and Technology*, pages 48–51, 2008. (Cited page 16.)

[66] Simone Marini, Giuseppe Patanè, Michela Spagnuolo, and Bianca Falcidieno. Feature selection for enhanced spectral shape comparison. In *Eurographics workshop on 3D object retrieval -3DOR*, pages 31–38, 2010. (Cited pages 22 and 131.)

[67] H. Masaki, S. Yoshihisa, K. Taku, and K. Tosiyasu. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH*, pages 203–212, 2001. (Cited pages 28, 47, 68, 69, 74, and 76.)

[68] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Discrete differential-geometry operators for triangulated manifolds. 2002. (Cited page 64.)

[69] P-W. Michor and D. Mumford. Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc.*, 8:1–48, 2006. (Cited page 93.)

[70] Marston Morse. Relations between the critical points of a real function of n independent variables. *Transactions of the American Mathematical Society*, 27(3):pp. 345–396, 1925. (Cited page 71.)

[71] M. Mortara and G Patanè. Affine-invariant skeleton of 3d shapes. In *IEEE Shape Modeling International*, pages 245–252, 2002. (Cited pages 47, 49, and 54.)

[72] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001. (Cited page 20.)

[73] Xinlai Ni, Michael Garland, and John C. Hart. Fair morse functions for extracting the topological structure of a surface mesh. *ACM Trans. Graph.*, 23:613–622, August 2004. (Cited pages 47, 48, and 49.)

[74] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno. Salient local visual features for shape-based 3d model retrieval. In *International Conference on Shape Modeling and Applications (SMI)*, 2008. (Cited page 37.)

[75] Ryutarou Ohbuchi and Takahiko Furuya. Accelerating bag-of-features sift algorithm for 3d model retrieval. In *Proc. SAMT 2008 Workshop on Semantic 3D Media (S-3D)*, 2008. (Cited page 115.)

[76] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics (TOG)*, 21(4):807–832, 2002. (Cited pages xi, 19, and 20.)

[77] M. Ovsjanikov, Alexander M. Bronstein, Michael M. Bronstein, and L-J. Guibas. Shape google: a computer vision approach for invariant shape retrieval. In *Workshop on Nonrigid Shape Analysis and Deformable Image Alignment (NORDIA)*, 2009. (Cited page 38.)

[78] E. Paquet and M. Rioux. Nefertiti: a query by content system for three-dimensional model and image databases management. *Image and Vision Computing*, 17:157–166, 1999. (Cited page 17.)

[79] Valerio Pascucci, Giorgio Scorzelli, Peer-Timo Bremer, and Ajith Mascarenhas. Robust on-line computation of reeb graphs: simplicity and speed. In *ACM SIGGRAPH*. ACM, 2007. (Cited pages 76, 79, and 81.)

[80] Ulrich Pinkall, Strasse Des Juni, and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2:15–36, 1993. (Cited page 47.)

[81] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 1993. (Cited pages 63 and 64.)

[82] Zheng Qin, Ji Jia, and Jun Qin. Content based 3d model retrieval: A survey. In *International Workshop on Content-Based Multimedia Indexing. CBMI.*, pages 249 –256, june 2008. (Cited page 16.)

[83] Huaijun Qiu and Edwin R. Hancock. Clustering and embedding using commute times. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1873–1890, November 2007. (Cited page 53.)

[84] G. Reeb. Sur les points singuliers d'une forme de pfaff complètement intégrableou d'une fonction numérique. *In Comptes Rendus, Académie des Sciences de Paris*, 222:847–849, 1946. (Cited pages 28, 68, and 74.)

[85] Martin Reuter. Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions. *International Journal of Computer Vision*, 89(2):287–308, 2010. (Cited page 21.)

[86] Martin Reuter, Silvia Biasotti, Daniela Giorgi, Giuseppe Patanè, and Michela Spagnuolo. Discrete laplace-beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390, 2009. (Cited page 21.)

[87] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. Laplace-beltrami spectra as 'shape-dna' of surfaces and solids. *Comput. Aided Des.*, 38(4):342–366, April 2006. (Cited pages xi, 20, 21, 22, 52, and 131.)

[88] J. Revaud, G. Lavoué, Y. Ariki, and A. Baskurt. Scale-invariant proximity graph for fast probabilistic object recognition. In *Conference on Image and Video Retrieval*, 2010. (Cited page 131.)

[89] J. Ricard, D. Coeurjolly, and A. Baskurt. Generalizations of angular radial transform for 2d and 3d shape retrieval. *Pattern Recogn. Lett.*, 26, October 2005. (Cited pages 2 and 8.)

[90] Mauro R. Ruggeri, Giuseppe Patanè, Michela Spagnuolo, and Dietmar Saupe. Spectral-driven isometry-invariant matching of 3d shapes. *Internationa Journal of Computer Visionl*, 89(2-3):248–265, September 2010. (Cited page 26.)

[91] Raif M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Eurographics symposium on Geometry processing*, 2007. (Cited page 22.)

[92] C. Samir, A. Srivastava, M. Daoudi, and E. Klassen. An intrinsic framework for analysis of facial surfaces. *Internationa Journal of Computer Visionl*, 82:80–95, April 2009. (Cited pages 88 and 89.)

[93] A. Shamir. A formulation of boundary mesh segmentation. In *Symposium on 3D Data Processing, Visualization and Transmission*, pages 82 – 89, September 2004. (Cited pages 47, 68, 69, and 74.)

[94] Yu-Te Shen, Ding-Yun Chen, Xiao-Pei Tian, and Ming Ouhyoung. 3d model search engine based on lightfield descriptors. In *Eurographics*, 2003. (Cited page 32.)

[95] Y. Shinagawa, T-L. Kunii, and Y-L. Kergosien. Surface coding based on morse theory. *Computer Graphics and Applications*, 11:66–78, September 1991. (Cited pages 28, 47, and 76.)

[96] Y. Shinagawa and T.L. Kunii. Constructing a reeb graph automatically from cross sections. *Computer Graphics and Applications*, 11(6):44 –51, nov 1991. (Cited pages 47 and 49.)

[97] Dirk Smeets, Thomas Fabry, Jeroen Hermans, Dirk Vandermeulen, and Paul Suetens. Inelastic deformation invariant modal represen-

tation for non-rigid 3d object recognition. In *International Conference on Articulated Motion and Deformable Objects*, AMDO, pages 162–171, Berlin, Heidelberg, 2010. Springer-Verlag. (Cited page 105.)

[98] Olga Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 2006. (Cited page 21.)

[99] Dvir Steiner and Anath Fischer. Cutting 3d freeform objects with genus-n into single boundary surfaces using topological graphs. In *Symposium on Solid Modeling and Applications*, pages 336–343, 2002. (Cited pages 68, 69, and 74.)

[100] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009. (Cited pages 26, 27, 35, 52, and 55.)

[101] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Shape Modeling International*, 2003. (Cited page 29.)

[102] H. Tabia, M. Daoudi, J.-P. Vandeborre, and O. Colot. A new 3d-matching method of nonrigid and partially similar models using curve analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):852 –858, april 2011. (Cited page 97.)

[103] Hedi Tabia, Mohamed Daoudi, Jean Philippe Vandeborre, and Olivier Colot. Local visual patch for 3d shape retrieval. In *Proceedings of the ACM workshop on 3D object retrieval*, 3DOR, pages 3–8. ACM, 2010. (Cited pages 24, 88, and 89.)

[104] Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, and Olivier Colot. Deformable shape retrieval using bag-of-feature techniques. In *SPIE Electronic Imaging Symposium (3D Image Processing (3DIP) and Applications II)*, 2011. (Cited page 115.)

[105] Johan W. H. Tangelder and Remco C. Veltkamp. A survey of content based 3d shape retrieval methods. In *Multimedia Tools and Applications*, pages 441–471, 2008. (Cited page 16.)

[106] Sergey P. Tarasov and Michael N. Vyalyi. Construction of contour trees in 3d in o(n log n) steps. In *Proceedings of the fourteenth annual symposium on Computational geometry*, SCG, pages 68–75. ACM, 1998. (Cited page 73.)

[107] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive technnulliques*, SIGGRAPH, pages 351–358. ACM, 1995. (Cited pages 51 and 63.)

[108] J. Tierny, J.-P. Vandeborre, and M. Daoudi. Fast and precise kinematic skeleton extraction of 3d dynamic meshes. In *19th International Conference on Pattern Recognition*, December 2008. (Cited page 86.)

[109] J. Tierny, J-P. Vandeborre, and M. Daoudi. Partial 3d shape retrieval by reeb pattern unfolding. *Computer Graphics Forum - Eurographics Association*, 28:41–55, March 2009. (Cited pages xi, xii, 36, 37, and 48.)

[110] Julien Tierny. *Reeb graph based 3D shape modeling and applications*. PhD thesis, Université des Sciences et Techonologies de Lille, 2008. (Cited pages xii, 68, 69, 74, 75, 76, and 77.)

[111] Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. 3d mesh skeleton extraction using topological and geometrical analyses. In *14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006)*, pages 85–94. (Cited pages 24, 28, 47, 49, 55, 56, and 58.)

[112] R. Toldo, U. Castellani, and A. Fusiello. A bag of words approach for 3d object categorization. In *Proceedings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques*, 2009. (Cited pages 38 and 115.)

[113] Tony Tung. *Indexation 3D de bases de données d'objets par graphes de Reeb améliorés*. PhD thesis, ENST Paris, 2005. (Cited pages xi, 28, and 29.)

[114] Bruno Vallet and Bruno Lévy. Manifold harmonics. Technical report, INRIA - ALICE Project Team, 2007. (Cited page 65.)

[115] D-V. Vranic. An improvement of rotation invariant 3d-shape based on functions on concentric spheres. In *2003 International Conference on Image Processing. ICIP. Proceedings.*, volume 3, pages III – 757–60 vol.2, 2003. (Cited page 26.)

[116] Yu-Shuen Wang and Tong-Yee Lee. Curve-skeleton extraction using iterative least squares optimization. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):926–936, July 2008. (Cited page 29.)

[117] Stefanie Wuhrer, Chang Shu, Prosenjit Bose, and Zouhour Ben Azouz. Posture invariant correspondence of incomplete triangular manifolds. *International Journal of Shape Modeling*, pages 139–157, 2007. (Cited page 105.)

[118] A-J. Yezzi and A. Mennucci. Conformal metrics and true "gradient flows" for curves. In *IEEE International Conference on Computer Vision*, volume 1, pages 913–919, 2005. (Cited page 94.)

[119] T. Zaharia and F. Preteux. Indexation de maillages 3d par descripteurs de forme. In *Congrès francophone sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA)*, pages 48–57, 2002. (Cited page 19.)

[120] Cha Zhang and Tsuhan Chen. Efficient feature extraction for 2d/3d objects in mesh representation. In *International Conference on Image Processing (ICIP)*, volume 3, 2001. (Cited page 19.)

[121] Hao Zhang and Rong Liu. Mesh segmentation via recursive and visually salient spectral cuts. In *Proc. of Vision, Modeling, and Visualization*, pages 429–436, 2005. (Cited page 20.)

Titre Indexation partielle de modèles 3D

Résumé Un nombre croissant d'applications graphiques 3D ont un impact sur notre société. Ces applications sont utilisées dans plusieurs domaines allant des produits de divertissement numérique, la conception assistée par ordinateur, aux applications médicales. Dans ce contexte, un moteur de recherche d'objets 3D avec de bonnes performances en résultats et en temps d'exécution devient indispensable. Nous proposons une nouvelle méthode pour l'indexation de modèles 3D basée sur des courbes fermées. Nous proposons ensuite une amélioration de notre méthode pour l'indexation partielle de modèles 3D. Notre approche commence par la définition d'une nouvelle fonction d'application invariante. Notre fonction d'application possède des propriétés importantes : elle est invariante aux transformations rigides et non rigides, elle est insensible au bruit, elle est robuste à de petits changements topologiques et elle ne dépend pas de paramètres. Cependant, dans la littérature, une telle fonction qui respecte toutes ces propriétés n'existe pas. Pour respecter ces propriétés, nous définissons notre fonction basée sur la distance de diffusion et la distance de migration pendulaire. Pour prouver les propriétés de notre fonction, nous calculons le graphe de Reeb de modèles 3D. Pour décrire un modèle 3D complet, en utilisant notre fonction d'application, nous définissons des courbes de niveaux fermées à partir d'un point source détecté automatiquement au centre du modèle 3D. Chaque courbe décrit alors une région du modèle 3D. Ces courbes créent un descripteur invariant à différentes transformations. Pour montrer la robustesse de notre méthode sur différentes classes de modèles 3D dans différentes poses, nous utilisons des objets provenant de SHREC 2012. Nous comparons également notre approche aux méthodes de l'état de l'art à l'aide de la base SHREC 2010. Pour l'indexation partielle de modèles 3D, nous améliorons notre approche en utilisant la technique *sacs de mots*, construits à partir des courbes fermées extraites, et montrons leurs bonnes performances à l'aide de la base précédente.

Mots-clés Modèles 3D ; Noyau de la chaleur ; Distance de diffusion ; Distance de migration pendulaire ; Graphes de Reeb ; Indexation ; Indexation partielle ; Sacs de mots.

Title   Partial 3D-shape indexing and retrieval

Abstract   A growing number of 3D graphic applications have an impact on today's society. These applications are being used in several domains ranging from digital entertainment, computer aided design, to medical applications. In this context, a 3D object search engine with a good performance in time consuming and results becomes mandatory. We propose a novel approach for 3D-model retrieval based on closed curves. Then we enhance our method to handle partial 3D-model retrieval. Our method starts by the definition of an invariant mapping function. The important properties of a mapping function are its invariance to rigid and non rigid transformations, the correct description of the 3D-model, its insensitivity to noise, its robustness to topology changes, and its independance on parameters. However, current state-of-the-art methods do not respect all these properties. To respect these properties, we define our mapping function based on the diffusion and the commute-time distances. To prove the properties of this function, we compute the Reeb graph of the 3D-models. To describe the whole 3D-model, using our mapping function, we generate indexed closed curves from a source point detected automatically at the center of a 3D-model. Each curve describes a small region of the 3D-model. These curves lead to create an invariant descriptor to different transformations. To show the robustness of our method on various classes of 3D-models with different poses, we use shapes from SHREC 2012. We also compare our approach to existing methods in the state-of-the-art with a dataset from SHREC 2010. For partial 3D-model retrieval, we enhance the proposed method using the Bag-Of-Features built with all the extracted closed curves, and show the accurate performances using the same dataset.