Numéro d'ordre: 40582

Université Lille 1 Sciences et Techonologies

Laboratoire d'Automatique, Génie Informatique et Signal

These

Présentée en premiere version en vu d'obtenir le grade de Docteur, spécialite Automatique, Génie informatique, Traitement du Signal et des Images

par

Hedi Tabia

Contributions to 3D-shape matching, retrieval and classification

Thèse soutenue le le 27 Septembre 2011 devant le jury compose de :

M	Atilla Baskurt	Professeur, LIRIS/INSA Lyon	(Rapporteur)
M	Alberto Del Bimbo	Professeur, University of Firenze	(Rapporteur)
M	Michael M. Bronstein	Professeur, Università della Svizzera Italiana	(Examinateur)
M	Denis Pellerin	Professeur, Université Joseph Fourier	(Examinateur)
M	Olivier Colot	Professeur, LAGIS/Université de Lille 1	(Co-directeur)
M	Mohamed Daoudi	Professeur, LIFL/TELECOM Lille 1	(Co-directeur)
M	Jean-Philippe Vandeborre	Maître de Conférences, LIFL/TELECOM Lille 1	(Examinateur)

This manuscript is dedicated

to all those who contributed, from near or far, to its elaboration.

Acknowledgements

In the first place, I would like to record my gratitude to my two supervisors, Pr Olivier Colot and Pr. Mohamed Daoudi for thier supervision, advice, and guidance from the very early stage of this research as well as giving me extraordinary experiences through out the scientific research world. I deeply thank Dr. Jean-Philippe Vandeborre whose help, advice and supervision was invaluable.

Special thank goes to the members of committee, in particular, Mr. Atilla Baskurt (Professeur at LIRIS/INSA Lyon) and Mr. Alberto Del Bimbo (Professeur, University of Firenze) for accepting to review my thesis and for their interesting comments. I also thank the rest of the members; Michael M. Bronstein (Professeur, Università della Svizzera Italiana) and Mr. Denis Pellerin (Professeur, Université Joseph Fourier). All these people made me the honor 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.

Words fail me to express my gratitude to my family: my parents, my brother and my sisters who always were present to encourage me and support me.

Finally, I thank people at TELECOM Lille 1. Their hospitality and support during the period this research took place is greatly acknowledged.

Author's publications

Journals:

- **Hedi Tabia**, Mohamed Daoudi, Jean-Philippe Vandeborre and Olivier Colot "A new 3D-matching method of non-rigid and partially similar models using curve analysis" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, April 2011 (vol. 33 no. 4).
- **Hedi Tabia**, Mohamed Daoudi, Jean-Philippe Vandeborre and Olivier Colot "A part-based approach for automatic 3D-shape categorization" *ACM Transactions on Intelligent Systems and Technology*, (under revision).
- **Hedi Tabia**, Mohamed Daoudi, Olivier Colot, Jean-Philippe Vandeborre "3D-object retrieval based on vector quantization of invariant descriptors" *Journal of Electronic Imaging*, (under revision).

International Conferences:

- **Hedi Tabia**, Mohamed Daoudi, Jean-Philippe Vandeborre, Olivier Colot "Non-Rigid 3D Shape Classification using BOF Techniques" *ICME*-July 2011.
- **Hedi Tabia**, Mohamed Daoudi, Jean-Philippe Vandeborre, Olivier Colot "Deformable Shape Retrieval using Bag-of-Feature techniques" *Electronic Imaging Conference, 3D Image Processing* (3DIP) and Application-January 2011

- **Hedi Tabia**, Mohamed Daoudi, Jean-Philippe Vandeborre, Olivier Colot "Local Visual Patch for 3D Shape Retrieval" *ACM International Workshop on 3D Object Retrieval* (in conjunction with ACM Multimedia 2010), Firenze, Italy, October 25, 2010.

- **Hedi Tabia**, Mohamed Daoudi, Jean-Philippe Vandeborre, Olivier Colot "3D-shape retrieval using curves and HMM" 20th *IEEE International Conference on Pattern Recognition* (ICPR 2010), Istanbul, Turkey, August 23-26, 2010.

- Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H.V. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, **H. Tabia**, D. Vandermeulen "SHREC'11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes" *Eurographics Workshop on 3D Object Retrieval* April 10, 2011.

French national conferences:

- **Hedi Tabia**, Mohamed Daoudi, Jean-Philippe Vandeborre, Olivier Colot, Une approche pour la catégorisation des objets 3D basée sur la théorie des fonctions de croyance, dans Compression et Représentation des Signaux Audiovisuels, *CORESA'2010*, Oct. 2010.

Contents

List of Figures

1 Introduction

Three dimensional (3D) object representations have become an integral part of modern computer graphics applications, such as computer-aided design, game development and film production. At the same time, 3D data have become very common in domains such as computer vision, computational geometry, molecular biology and medicine. The rapid evolution in graphics hardware and software development, in particular the availability of low cost 3D scanners, has greatly facilitated 3D model acquisition, creation and manipulation, giving the opportunity to experience applications using 3D models to a large user community. Figure 1.1 shows the Google 3D-warehouse. It is a collection of 3D-data such as buildings, bridges, cars and so on. It is getting bigger every day and it is completely accessible for every one to use.

Content-based search is a necessary solution for structuring, managing these multimedia data, and browsing within these data collections. In this context, we are looking for a system that can automatically retrieve the 3D-models visually similar to a requested 3D-object.

Most proposed approaches for content-based indexing used 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 1.2). The mesh representa-

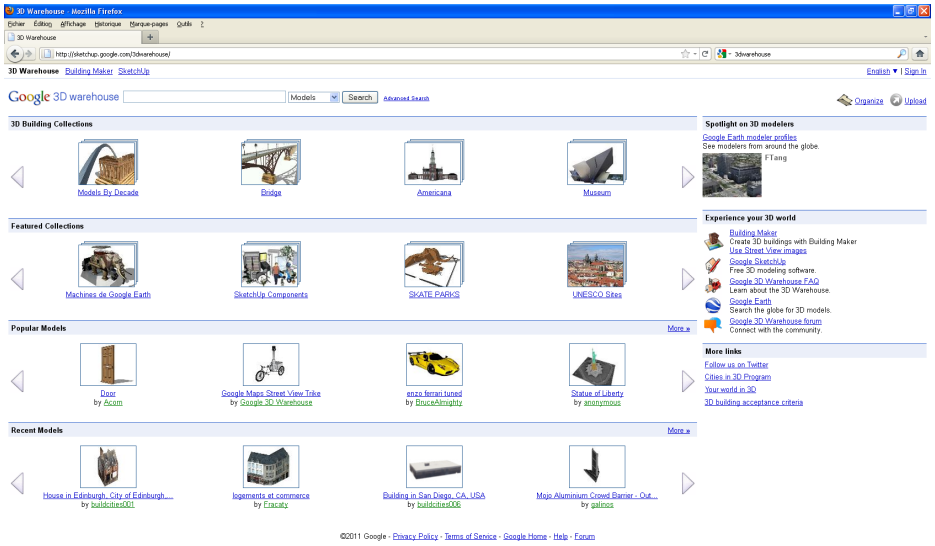


Figure 1.1 – *The Google 3D-warehouse (http://sketchup.google.com/3dwarehouse/)*

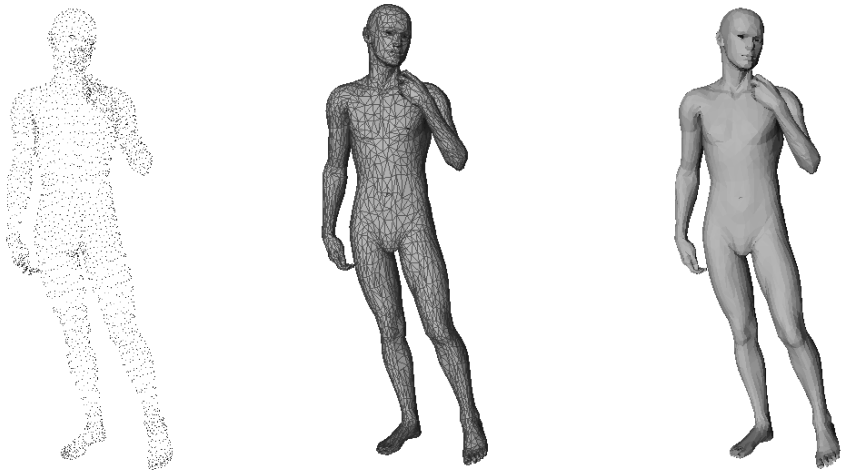


Figure 1.2 – *An example of 3D-object. From left to right: points, flat lines and flat Rendering.*

tion is the simplest and the most frequently encountered representation everywhere [DBD08].

In the literature, different kind of shape indexing approaches exist. The oldest methods are inspired from 2D-methods like Fourier descriptors [MAS01], invariant moments calculations [Can99], median line extraction, angular radial transform [RCB05] 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 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, they suffer from high variability towards shape-preserving transformations like affine or isometric transformations (non-rigid transformations). Figure 1.3 shows different shapes under non rigid transformation and with some topological changes.

The aim of this thesis is to propose a 3D-matching method that can be robust under non-rigid transformations as well as rigid transformations.

1.1 Contributions

In this thesis, we develop novel approaches for 3D object matching. The approaches are robust to isometric transformation as well as non rigid ones. First, we propose two 3D-shape descriptors, namely *Curve descriptor* and *Geodesic cord descriptors*. These descriptors which reflect the local geometric characteristics of given 3D-objects, are computed on patches of the objects. To separate the different patches of a given object, we propose the use of feature points extraction algorithm.

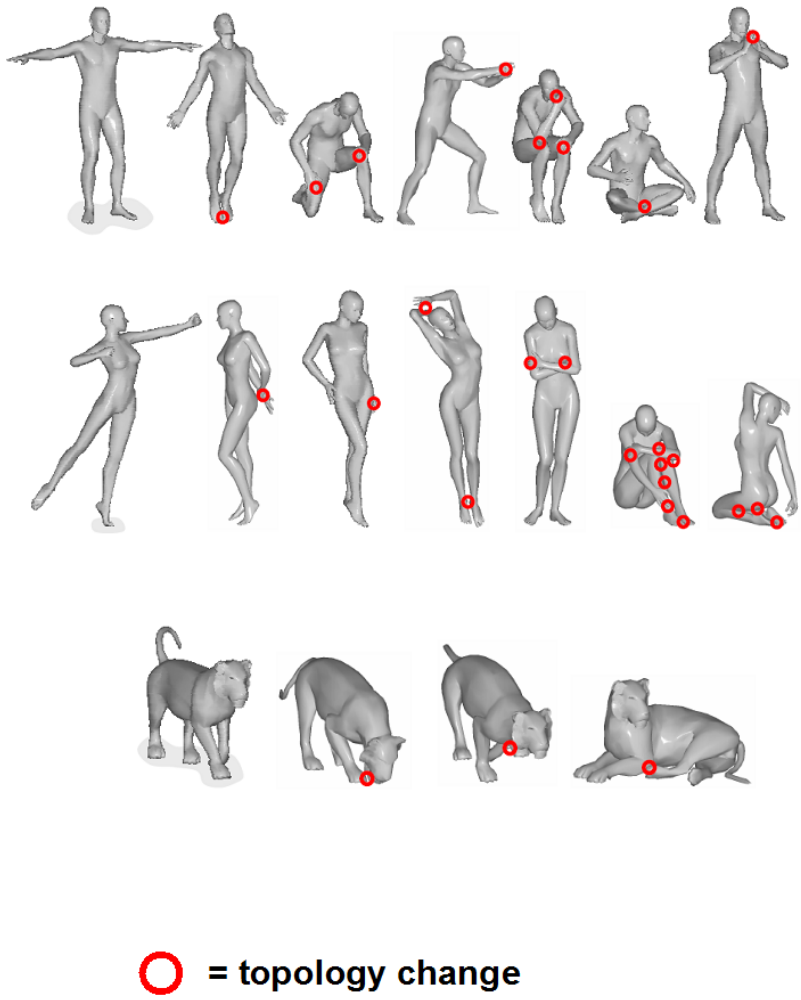


Figure 1.3 – *Example of 3D-objects under some deformations with topological changes. Objects from TOSCA dataset*

The descriptors that we propose are based on the intrinsic propriety of the shapes of 3D-patches. It is in turn invariant to the rigid as well as non-rigid transformations of the surface.

The curve descriptor consists to represent a 3D-patch by an indexed collection of closed level curves in $\mathbb{R}^3$ extracted around each feature point. Then, tools from shape analysis of curves are applied to analyze and to compare curves. We use an extension of the Riemannian framework proposed by Joshi et al. [JKSJ07] to: (i) compute distances

between curves to quantify differences in their shapes, (ii) find optimal deformations between curves, and (iii) define and compute average of a given set of curves.

The Geodesic cord descriptor represents the shape signature for a 3D-patch as a probability distribution sampled from a shape function measuring intrinsic properties of the 3D-patch. The distribution is sampled from an intrinsic distance function on the 3D-surface. The distance function used here is based on the geodesic distances between the feature point and all the points on the 3D-patch surface.

Second, we use the belief functions, as fusion technique, to define a global distance between 3D-objects. We experiment this technique in the retrieval and classification tasks. Our approach is based on the use of information extracted from the different patches (descriptors) of the 3D-object. Each patch, provides an information source regarding the shape of that object. A combination of these information is necessary in order to increase the recognition rate of the object.

At last, we propose the use of *Bag of Feature* (BoF) techniques in 3D-object retrieval and classification. The method of BoF is largely inspired by the *Bag of Words* concept which has been used in text retrieval and classification for quite some time. Even though there are countless variations of algorithms emerging under the label Bag of Features and it is hard to capture the actual BoF algorithm, there is a common concept which is shared by all of theses methods. The vocabulary construction and the bag of word histogram computing.

1.2 Outline

The rest of this manuscript is laid out as follows.

In chapter 2, we review the existing solutions for 3D shape description and present the technique used for shape matching.

In chapter 3, we propose two feature descriptors used in our framework.

In chapter 4, we propose a belief function based technique for 3D-shape matching . We present then its application in the case of the retrieval task.

In chapter 5, we present an automatic classification framework for categorizing 3D-objects.

Chapter 6 details another method proposed under this thesis. The method is based on the Bag of Feature (BoF) techniques. This method is also applied in the retrieval and the classification tasks.

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

1.3 French Introduction

La représentation trois dimensionnelle d'objets (3D) est devenue une partie intégrante de différentes applications modernes, telles que la conception assistée par ordinateur, le développement de jeux vidéo et la production cinématographique. Dans le même temps, les données 3D sont devenues très communs dans des domaines tels que la vision par ordinateur, la géométrie algorithmique, la biologie moléculaire et la médecine. L'évolution rapide du matériel et des logiciels graphiques, en particulier la disponibilité du faible coût de scanners 3D, a grandement facilité l'acquisition, la création et la manipulation des modèles 3D. La figure 1.1 montre l'entrepôt de données 3D de Google. Il s'agit d'une collection de d'objets 3D telles que les bâtiments, les ponts, les voitures etc. Cette collection s'agrandit chaque jour et elle est entièrement accessible pour tout le monde à utiliser. La recherche d'objets 3D par le contenu est une solution nécessaire à la structuration, la gestion de ces

données multimédia. Dans ce contexte, nous sommes à la recherche d'un système qui peut automatiquement retrouver des modèles 3D visuellement similaires à un modèle 3D requête. En général, les approches proposées pour l'indexation par le contenu utilisent des histogrammes statistiques mesurant certaines caractéristiques géométriques des objets 3D. Les différentes mesures de ces caractéristiques sont calculées à partir d'un maillage 3D des surfaces triangulées (voir figure 1.2). La représentation en maillage 3D la représentation la plus fréquemment rencontrée partout [DBD08].

Dans la littérature, différents types d'approches indexation 3D existent. Les plus anciennes méthodes sont inspirées des méthodes 2D comme les descripteurs de Fourier [MAS01], les calculs des moments [Can99], l'extraction de la ligne médiane, angulaire radiale transformation angulaire radiale [RCB05] etc. La littérature concernant les descripteurs d'objets 3D est très riche. On peut classifier ces descripteurs en deux catégories. Les méthodes basées sur une description globale d'objets qui caractérise la forme des objets 3D d'une manière grossière. Ces méthodes sont souvent efficaces pour discriminer des formes simples mais insuffisamment discriminants pour des formes plus complexes. Inversement, la 3D-local des approches caractériser les propriétés locales de la surface de l'objet.

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, ils sont moins robustes aux transformations affines ou isométriques (transformation non rigide) qu'un objet peut subir. La figure 1.3 montre différentes formes de transformation non rigide d'un objet 3D.

Dans cette thèse, nous développons de nouvelles approches pour la mise en correspondance d'objet 3D. Les approches sont robustes aux

transformations isométriques ainsi que aux transformations non rigides. Premièrement, nous proposons deux descripteurs de forme 3D, à savoir descripteur basé courbes et descripteur de cordes géodésiques. Ces descripteurs qui reflètent les caractéristiques géométriques locales de l'objet 3D, sont calculés sur des patches extraits des objets. Pour séparer les différents patches d'un objet donné, nous développons un algorithme d'extraction des points algorithme. Les descripteurs que nous proposons sont basés sur la propriété intrinsèque de la forme des objets 3D. Ils sont donc invariants aux transformations rigides ainsi qu'aux transformations non-rigides. Le descripteur basé courbes consiste à représenter un patch 3D par une collection indexée des courbes de niveau fermé dans $\mathbb{R}^3$ extraite autour de chaque point caractéristique. Des outils de l'analyse de la forme des courbes sont appliqués afin de comparer les descripteurs. Nous utilisons une extension du cadre riemannien proposé par Joshi et al. [JKSJ07] qui consiste à :(i) calculer les distances entre les courbes pour quantifier la différence entre leurs formes, (ii) trouver les déformations optimales entre les courbes, et (iii) définir et calculer la moyenne d'un ensemble donné de courbes.

Le descripteur de cordes géodésiques représente la forme d'un patch 3D comme une distribution de probabilité d'une fonction mesurant des propriétés intrinsèques du patch 3D. La fonction utilisée ici est basée sur les distances géodésiques entre un point caractéristique et tous les points de la surface 3D du patch. Deuxièmement, nous utilisons les fonctions de croyance, comme technique de fusion, de définir une distance globale entre les objets 3D. Nous expérimentons cette technique dans les tâches de recherche et de classification. Notre approche est basée sur l'utilisation des informations extraites des différents patches (descripteurs) de l'objet 3D. Chaque patch, fournit une source d'information concernant la forme de cet objet. Une combinaison

de ces informations est nécessaire afin d'augmenter le taux de reconnaissance de l'objet. Enfin, nous proposons l'utilisation des techniques inspirés par le concept *sac de mots* qui a été utilisé dans la recherche textuelle. Nous avons adapté ces techniques pour la recherche et la classification d'objets 3D.

Le reste de ce manuscrit est exposé comme suit.

Au chapitre 2, nous passons en revue les solutions existantes pour la description d'objets 3D en présentant les différentes techniques utilisées pour la mise en correspondance.

Dans le troisième chapitre, nous proposons deux descripteurs développés dans le cadre de cette thèse.

Dans le quatrième chapitre, nous proposons une technique basée sur les fonctions de croyance pour la mise en correspondance d'objets 3D. Nous présentons ensuite son application dans le cas de la recherche et la classification des objets 3D.

Dans le chapitre 5, nous présentons un système automatique de classification d'objets 3D.

Chapitre 6 détaille une deuxième méthode proposée dans cette thèse pour la recherche et la classification d'objets 3D. La méthode est basée sur les techniques de sac de mots (BoF).

Enfin, nous concluons ce manuscrit en résumant les contributions de cette thèse, énumérant les problèmes qui subsistent ouvert et proposer des orientations pour de futures recherches.

2 State-of-the-art of 3D-object matching methods

Contents

This chapter introduces existing solutions for three dimensional object description. This short survey also presents 3D-matching methods used to determine how similar two shapes are.

2.1 Introduction

3D-object classification and retrieval based on content, require having an automatic method for measuring the similarity between two objects. The general principle of such method is based on the assumption that the measure of similarity between two 3D-objects can be reduced to a distance computing between their two descriptors. A comparison between two 3D-objects usually involves two steps which are: signature extraction (or object description), and similarity computing.

The extraction of the signature This step consists to describe the 3D object (as a vector, a graph, sequence ...) obtained by means of one or more shape descriptors. In a 3D-object classification and retrieval based on content system, the signature of the query object is the search key with which 3D-objects in the database will be compared.

Similarity computing The similarity measure between two 3D objects results from the comparison between their two extracted signatures. Several approaches for similarity computing have been proposed in the literature. Most of them are based on the distances between signatures of the objects, and use machine learning techniques in order to match between them.

This chapter gives an overview of the 3D shape matching literature. First, we present the main existing methods for 3D-object description. Then, we present some existing techniques that are used for similarity computing between objects.

French Introduction

Une recherche par le contenu d'objets 3D nécessite de disposer d'une méthode automatique pour mesurer la similarité entre deux objets. le

principe général d'une telle méthode repose sur l'hypothèse que la mesure de similarité entre deux objets 3D peut se ramener au calcul de la distance entre deux descripteurs de ces objets. Un processus comparant deux objets comporte généralement deux étapes principales qui sont la description d'objets 3D, et la mesure de leur similarité.

Description d'objets 3D: consiste à décrire l'objets 3D au moyen d'un ou plusieurs descripteur de forme. Dans un processus général de recherche d'objets 3D. La signature de l'objet requête est la clé de recherche avec laquelle les éléments de la base vont pouvoir être comparés.

Mesure de similarité entre deux objet 3D : cette étape consiste à comparer les deux signatures extraite en utilisant un certain critère dit de similarité. Pour effectuer une requête de similarité par l'exemple, ce critère sert à évaluer la similarité entre l'objet exemple et les objets de la base. Ainsi, les objets vont pouvoir être classés selon leur ordre de ressemblance.

Ce chapitre donne un aperçu sur l'état de l'art de la mise en correspondance entre les objets 3D. Tout d'abord, nous présentons les principales méthodes existantes pour la description d'objet 3D. Ensuite, nous présenterons quelques techniques existantes qui sont utilisées pour la mesure de similarité entre deux objets 3D.

2.2 3D-object description methods

In the literature, we can distinguish four main families of approaches that are used to describe 3D-objects.

2.2.1 Global methods

A global descriptor takes into account the geometrical characteristics of the whole object. This section presents the main existing global methods and their respective advantages and weaknesses.

The cord histogram

Among the first who have developed a method for 3D indexing, we find Paquet and Rioux [PR99] who released in 1997 their first descriptor named the *cord descriptor*. The database, used by Paquet and Rioux, 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 color of any 3D-object.

The proposed indexing approach is based on some statistics on the cords of the object to be indexed. The cords are defined here as the segments connecting the center of gravity of the object and centers 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

In Paquet and Rioux's paper [PR99], descriptors characterizing the RGB color are also proposed in order to allow research using textures associated with objects. Under the Nefertiti project, a search engine for 3D object shape similarity was developed. It uses objects from their database. The queries are based on the different 3D criteria including the shape and the color of objects.

The feature vectors associated with 3D-models are cord and color histograms, the cord characteristics are computed on the whole mesh. According to the authors, the optimal size of histogram would be 64 bins. The technique is not very discriminative for objects of complex shape, and is not robust to perturbations on the connectivity of the mesh.

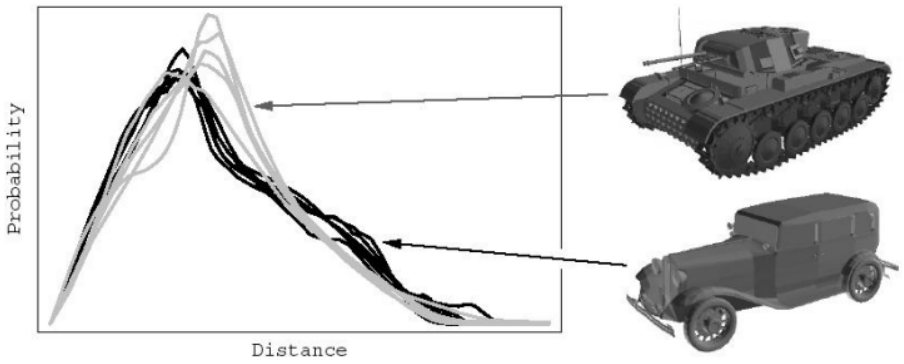


Figure 2.1 – *D2 shape distribution of different vehicles. [OFCD02]*

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** [OFCD02]. 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 distance between pairs of points chosen randomly on the surface of the object (see Figure 2.1).

The authors use histograms containing 64 bins to represent their descriptor, 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. In addi-

tion, the authors propose a research tool 2D-3D where the query is a 2D sketch drawn by the user and is supposed to correspond to 3D-object in the database.

D2 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 robustness with respect to the mesh noise (connectivity, decimation). The descriptors characterize the global shape of objects but not the details. The method seems to be more adapted to the research of similar objects in databases containing very different shapes.

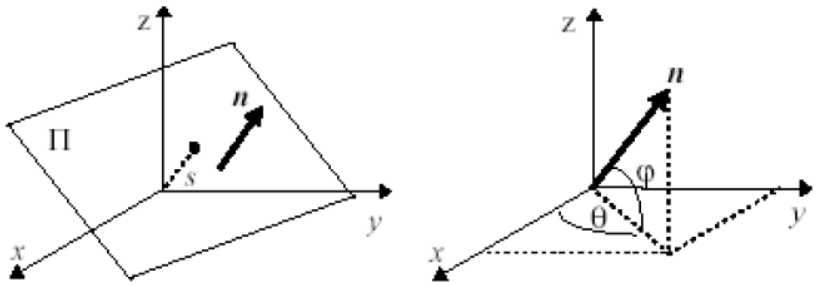


Figure 2.2 – *Switching to spherical coordinates for the Hough transform. The plan is described by 3 parameters*

3D Hough Descriptor

The 3D Hough Descriptor (3DHD) was proposed by Zaharia and Prêteux [ZP02] after the proposition of the 3D shape descriptor which they developed for MPEG-7 (see the local descriptors in Section 2.2.4). The Hough descriptor consists to accumulate the parameters of the representative planes defined by the triangles in a given 3D-mesh. In spherical coordinates, a plan Π is described by the triple (r, θ, ϕ), where $r \geq 0$ represents the distance from Π to the origin, $\theta \in [0, 2\pi[$ is the azimuth

angle and $\phi \in [\frac{-\pi}{2}, \frac{\pi}{2}[$ the elevation angle, (θ, ϕ) express the direction of the normal of the plan Π (see Figure 2.2).

Each axis of the spherical coordinate system is usually uniformly sampled. It then creates a 3D histogram for holding the different discretized triplets (r, θ, ϕ) where each triangle will make a contribution proportional to its area. The accumulation of contributions in the 3D histogram will characterize the surface of the 3D object. In order to overcome the problem of invariance to translations and rotations, a CPA must be implemented first. Furthermore, the authors propose to project the normals of the mesh faces on a regular octahedron subdivided and reprojected on the unit sphere, in order to overcome the problem of over-representation of triplets located near the poles produced by the uniform sampling of the spherical coordinates. The 3DHD proved its effectiveness on the MPEG-7 data, and its robustness with respect to possible changes in connectivity of the mesh The authors used two levels of decomposition for the octahedron (128 facets) and $N_s = 20$ (N_s is the number of bins to quantify s). The descriptor is then composed of 2560 elements. To reduce the size of the descriptor, the same authors [ZP04] propose a new version of the 3DHD with vector quantization, the QV-O3DHD, which returns essentially the same results as their previous descriptor but with feature vectors of 128 elements.

2.2.2 View based methods

The idea of using 2D-views to index 3D-models are 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 shapes of the objects. This section will briefly cover some of these techniques.

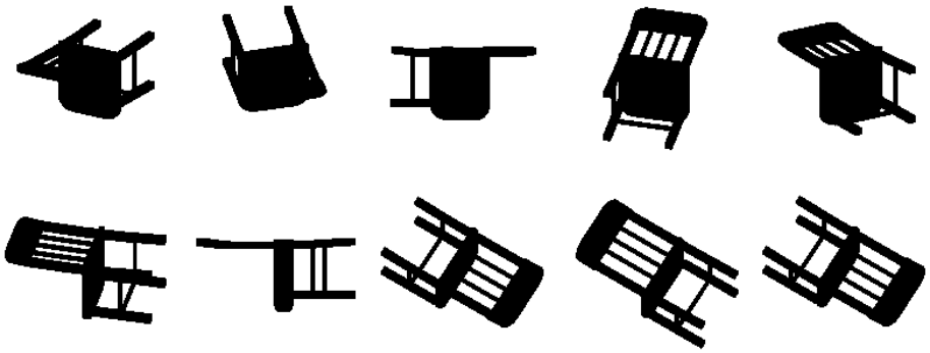


Figure 2.3 – *A typical example of the 10 silhouettes for a 3D model. [CTSO03]*

Silhouettes

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. The 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 Figure2.3).

However, the problem with this kind of descriptors is that, in theory, different 3D shapes might have the same set of silhouette images.

Aspect Graphs

The Aspect Graph descriptors is based on the fact that "3D shapes look different when viewed from different viewpoints". Based on this idea, the space of views can be partitioned into view classes or characteristic views. Within each class, the views share a certain property. A clustering algorithm might be used to generate the view classes. A view class representation, called an aspect graph, was proposed by Koenderink

and van Doorn in 1979 [KvD79]. The nodes of the graph represent the aspects, namely a class of views, and the edges connect different nodes which have a certain change in aspect.

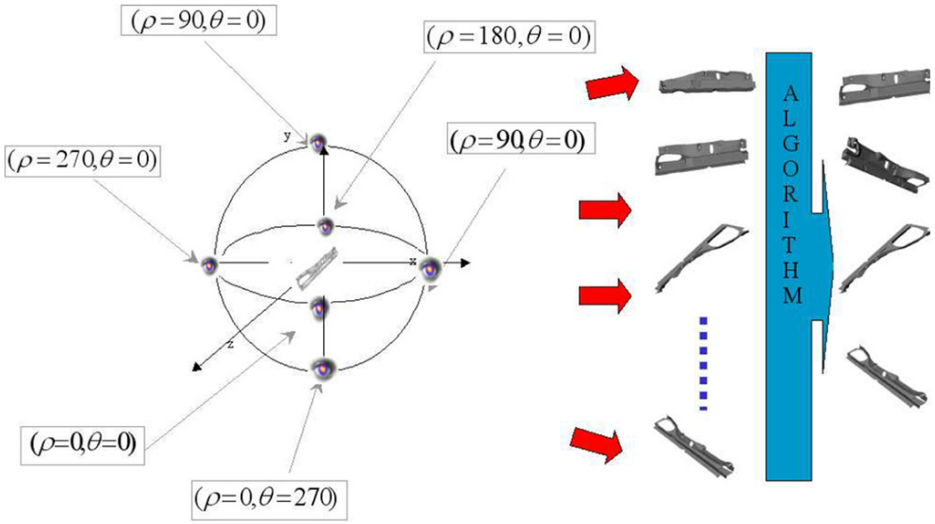


Figure 2.4 – *The AVC view selection process. [FADV07].*

Adaptive Views Clustering AVC

In Filali et al.'s paper, [FADV07], authors propose a method for 3D-model indexing based on 2D-views (see Figure 2.4), which they called adaptive views clustering. The goal of that method is to provide an "optimal" selection of 2D-views from a 3D-model, and a probabilistic Bayesian method for 3D-model retrieval from these views. The characteristic view selection algorithm is based on an adaptive clustering algorithm and uses statistical model distribution scores to select the optimal number of views. Starting from the fact that all views do not have equal importance, authors also introduce a novel Bayesian approach to improve the retrieval. They presented results and compared the method on the Princeton 3-D Shape Benchmark database and a 3-D-CAD-models database supplied by the car manufacturer Renault. Figure 2.5 shows a snapshot of the 3D-search engine provided by the authors.

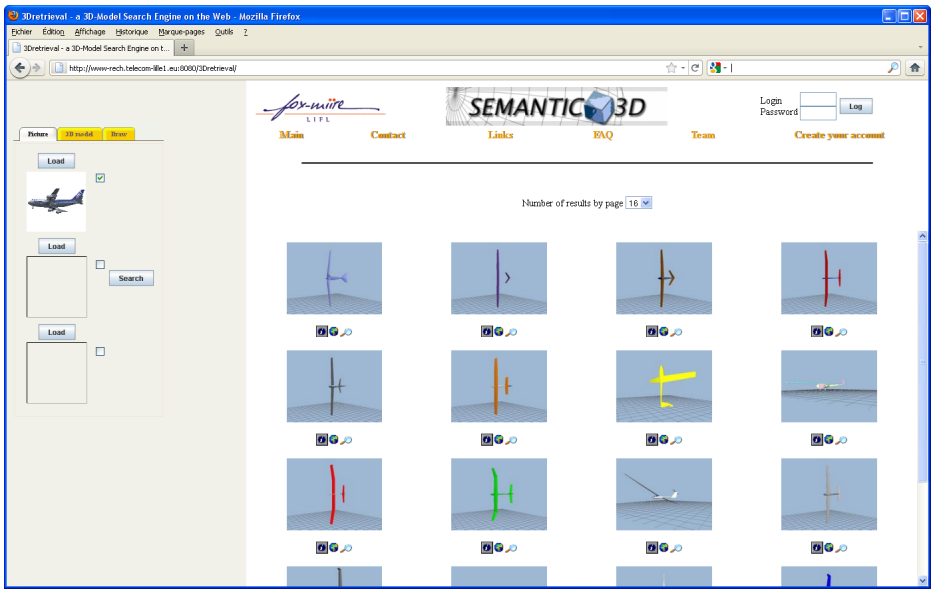


Figure 2.5 – *3D search engine of the MIIRE research team.*

2.2.3 Graph based methods

The graph based methods attempt to extract a geometric meaning from a 3D shape using a graph showing how shape components are linked together. Graph based methods can be divided into three broad categories according to the type of graph used [TV04]: (1) model graphs, (2) Reeb graphs, and (3) skeletons. Efficient computation of existing graph metrics for general graphs is not possible: computing the edit distance is NP-hard [ZWS95] and computing the maximal common subgraph [GJ90] is even NP-hard. Sebastian et al. [SKK01] describe an approach to compute a pseudo-metric between shock graphs. It is obtained by exhaustively searching for the optimal deformation path between two 2D-shapes, and using the cost of this path as a distance between two shapes. But the computation time of this method is too high for practical application, and it is not straightforwardly generalized to 3D.

Reeb graph Reeb graphs can obtain a representation of type skeleton preserving the topological structure of objects [Ree46]. Indeed, their

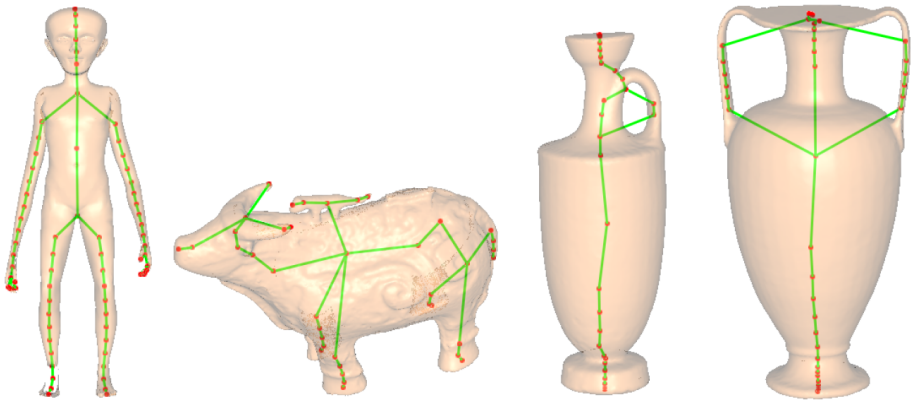


Figure 2.6 – *we find so many cycles in graphs as holes in objects. [Tuno5]*

constructions are based on Morse theory [SKK91] that characterizes the topology of closed surfaces, we find as many cycles in graphs as holes in objects (see Figure 2.6). In Biasotti et al's paper [BMM*03], a graph matching method based on the propagation of matched subgraphs was proposed. The method is of quadratic complexity and uses extended Reeb graphs (ERG) Reeb graphs are oriented and possess information to edges. For an extensive discussion of Reeb graphs and skeletons we refer the reader to the paper of Berretti et al. [BDBP09] and the paper of Tierny et al. [TVD09].

Multiresolution Reeb graph Hilaga et al. [MYTT01] 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 [Tuno5].

2.2.4 Local methods

The local shape description is founded on the premise that 3D-objects can be characterized by attributes computed on patches of the object. To separate the different patches 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. More recently, some authors propose the use of feature points extraction algorithm for detecting points of interest around which they extract patches. Then for each patch they calculate a specific feature. A local descriptor reflects the local geometric characteristics of a 3D-object, as opposed to global methods which tend to describe in a grossly way the shape of objects. Here, we first review some feature detection algorithm and then we describe some local based methods.

Feature detectors

Harris 3D The Harris operator, has been initially introduced in images [HS88] was extended to 3D shapes by Glomb [Glo09] and Sipiran and Bustos [IB10]. It is an effective feature detection method. It is based on the shape variability measure in a local neighborhood of the point, by fitting a function to the neighborhood, and identifying feature points as points where the derivatives of this function are high [BB10].

Mesh DOG A number of methods used to feature detection are inspired by the difference of Gaussians (DOG). The DOG is a conventional feature detection approach used in computer vision. The mesh DOG approach has been introduced in the first by [AEKH09]. The authors apply Gaussian filtering to functions (e.g. mean or Gauss curvature) defined on the shape. This creates a representation of the function in scale space, and feature points are prominent maxima of the scale space across scales. In [CCFM08], the authors apply Gaussian filtering directly on the mesh geometry, and use a robust method to detect feature points as points with greatest displacement in the normal direction.

Heat kernel feature detectors Recently, several authors like Sun et al. [SOG09], Gebal et al. [GBAL09] and Bronstein et al. [BBGO11] proposed feature detection methods based on the heat kernel. A quantity of heat remaining at a point after large time given a point source at time = 0, is measured. Feature points are defined as the set of local maxima of that measurement.

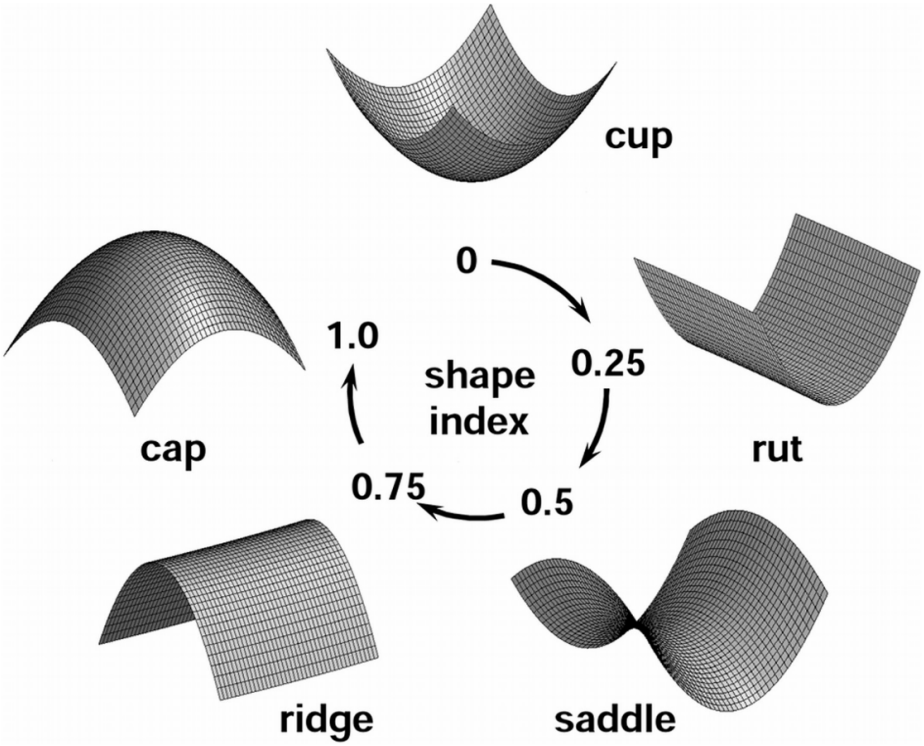


Figure 2.7 – *The shape index can characterize the local surface curvature.*

Local descriptors

Shape index histogram The 3D-shape index descriptor (3DSID) which was adopted as shape descriptor for MPEG-7 format, was proposed by Zaharia and Prêteux [ZP02]. The descriptor characterizes the local curvatures of the surface of 3D objects using their shape index [KvD79]. Formally, let p be a point on a regular 3D surface, and k_p^1 and k_p^2 are the principal curvatures of the surface associated to p. The shape index I_p

is written as:

$$I_p = \frac{1}{2} - \frac{1}{\pi} arctan \frac{k_p^1 + k_p^2}{k_p^1 - k_p^2}, \text{ with } k_p^1 \geq k_p^2$$

I_p takes value in [0,1] and it is not defined for planar surfaces for which we have: $k_1 = k_2 = 0$ (see Figure 2.7).

The principal curvatures are estimated directly from the triangulated mesh which should be already smoothed with a low-pass filter (example Laplacian filter) in order to eliminate the noise related to the position of the vertices. The principal curvatures are estimated by computing a second degree polynomial surface that approximates locally the 3D-mesh. In each vertex v of the mesh, the normal of the polynomial surface is equal to the sum of the normal of adjacent triangles containing v , the triangles are weighted by their surface. The method is invariant to rigid transformations, but depends on the connectivity of the mesh, and very sensitive to its quality (simplification, remeshing, etc.).

Among other approaches which aim to characterize 3D objects by estimating their local curvatures, we can find [AAV04] who determine the curvatures using "cones" formed around vertices and their associated N-rings 2. We can also find [ADBP03], who use a projection of curvature on a 2D map. These methods require regular meshes with very well oriented normals. Moreover, a pre-processing step is necessary to get a manifold mesh and to apply a low-pass filter in order to smooth the surface. The descriptor of 3D shape index is sensitive to details of objects and hence the relevance of signatures greatly depends on the nature of models.

Extended Gaussian Images The Extended Gaussian Images (EGI) descriptor consists to map a function that synthesizes some informations 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 [BH84].

There exists a variant of EGI, **the Complex Extended Gaussian Images** (CEGI) give 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 [KI93]. This representation allows to discriminate primitive shapes. In addition, it also allows to obtain many useful informations 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, 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.

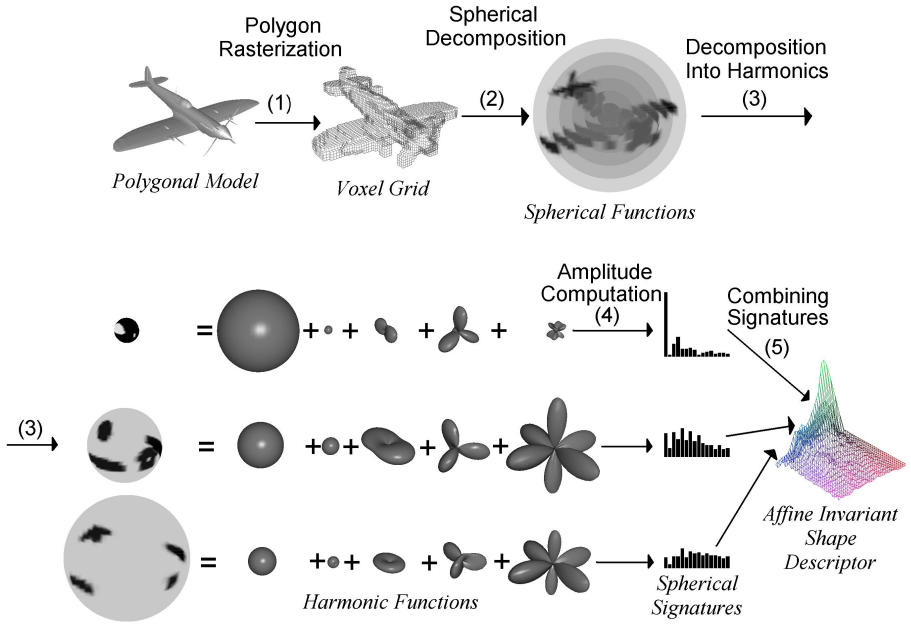


Figure 2.8 – *Computing the Harmonic Shape Representation. [KF02]*

Spherical Harmonics Saupe and Vranic [DD01] have proposed to apply a Fourier transform on the sphere S_2 by applying the spherical harmonics formulas proposed by [HRKM03]. Then, to overcome the problem of invariance to rotations, Kazhdan and Funkhouser [KF02] 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).

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 detail in the description the object.

Vranic [Vra03] proposed to apply the method directly on 3D-meshes with new 3D spherical functions. The results they obtained on their database with their 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 main limitation being the number of concentric spheres and the number of harmonic coefficients which remain may be too low. The authors choose in practice 32 and 16 concentric spheres by harmonic spheres, or a descriptor of 32 * 16 = 512 coefficients.

Spin images The spin image is a surface representation technique which was initially introduced by Andrew E. Johnson in [JH99] and is used for surface matching and object recognition in 3D-scenes. Spin images encode the overall properties of any surface in an object-oriented coordinate system rather than in a viewer-oriented coordinate system. Object-oriented coordinate systems are coordinate systems fixed on a surface or an object while viewer-oriented coordinate systems are based on the viewpoint of the observer of the surface. By using object-oriented

coordinate systems, the description of a surface or an object is view-independent and it does not change as the viewpoint changes [JH99]. Figure 2.9 shows the Spin image creation process for three different oriented points in the surface of an object.

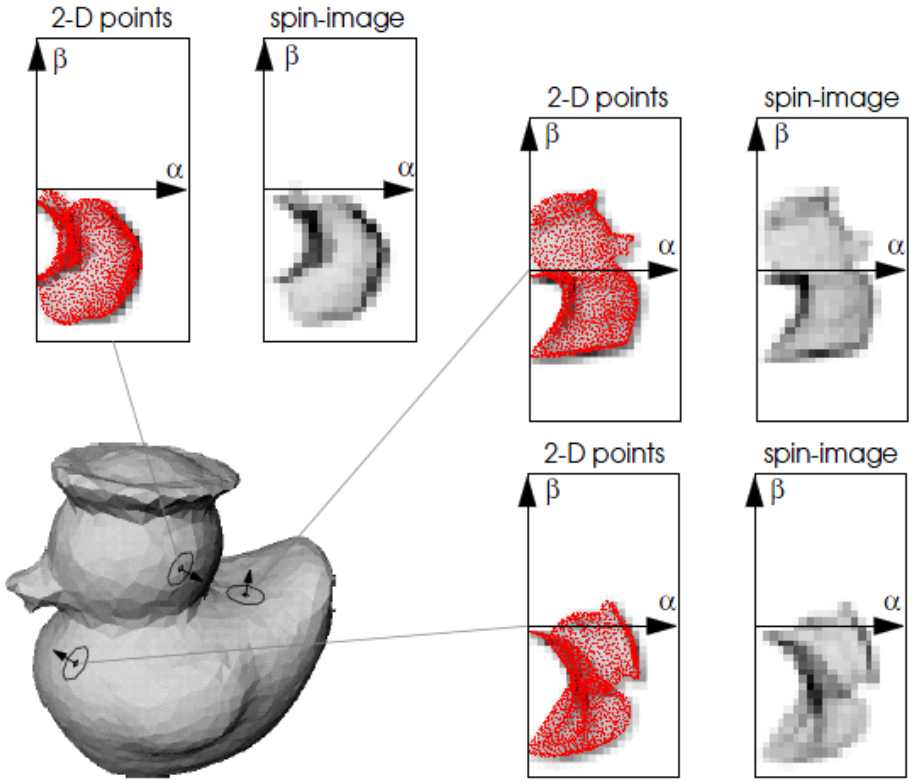


Figure 2.9 – *Spin image creation for three different oriented points in the surface of an object. [JH99]*

Heat kernel signature Just a while ago, the diffusion geometry has emerged as an important tool for shape recognition [Rus07, OBBG09, BBGO11]. This kind of geometry derived from the heat equation,

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

governing the conduction of the heat u on the surface X (in the equation, Δ_X indicates the negative semi-definite Laplace-Beltrami operator,

which is a generalization of the Laplacian to non-Euclidean domains). The fundamental solution $K_t(x,z)$ of the heat equation, also called the heat kernel

In [SOG09], Sun et al. used the diagonal of the heat kernel as a local descriptor, referred to as the heat kernel signatures (HKS). 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 intrinsic and thus isometry-invariant (two isometric shapes have equal HKS) (Figure 2.10), multi-scale and thus capture both local features and global shape structure, and also informative: under mild conditions, if two shapes have equal heat kernel signatures, they are isometric [SOG09].

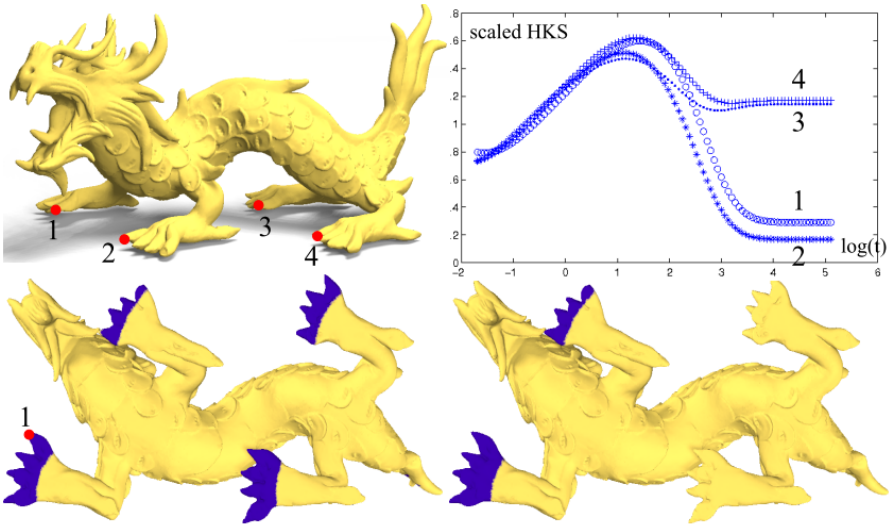


Figure 2.10 – *top left: dragon model; top right: scaled HKS at points 1, 2, 3 and 4. all four signatures are close at small t's while big t's separate the points on the front claws from those on back; bottom left: the points (blue), whose signature is close to the signature of point 1 based on the smaller half of the t's; bottom right: based on the entire range of t's. [SOG09]*

Among the different descriptors presented here, we find that lo-

cal descriptor based methods significantly improves the performance of classification or retrieval as it can represent more relevant informations.

2.3 Shape similarity computing

The different shape descriptors presented earlier are used to extract feature vectors which are introduced into the database as keys to represent 3D models and thus accelerate research. In order to measure the similarity between 3D models, it is necessary to define cost functions that, when applied to these models will return a score quantifying the similarity. This similarity can be used, in a next step, to classify or to index 3D-models.

2.3.1 Distance based method

There are different distance based methods that allow the computing of this similarity. The simplest way is to compute the distance as:

$$Dist(Q,O) = \sum_j min_k(d(q_j, o_k)),$$

where $d(q_j, o_k)$ represents the distance between $q_j \in Q$ and $o_k \in O$

We can also find the Hausdorff distance which represents the "maximum distance between a set of points and the closest point in another set of points"

$$h(Q,O) = max_{q \in Q} \{min_{o \in O} \{d(q,o)\}\},$$

where $d(q,o)$ represents the distance between q and o

2.3.2 Bayesian framework for 3D-matching

Several authors have developed probabilistic approaches for 3D-models retrieval and classification. For exemple, Yi and Chelberg [YC98] employed a Bayesian framework to achieve indexing of 3D-models. A

decision-theoretic measure of the discriminatory power of a feature for a model object is defined in terms of posterior probability. Domain-specific knowledge compiled off-line from CAD model data is used in order to estimate posterior probabilities that define the discriminatory power of features for model objects. In order to speed up the indexing or selection of correct objects, they generate and verify the object hypotheses for features detected in a scene in the order of the discriminatory power of these features for model objects. Based on these principles, they implemented a working prototype vision system using a feature structure called an LSG (local surface group) for generating object hypotheses. This object recognition system employ a wide class of features for generation of object hypotheses. In order to verify an object hypothesis, they estimate the view of the hypothesized model object and render the model object for the computed view. The object hypothesis is then verified by finding additional features in the scene that match those present in the rendered image. Experiments have been made on a small database of 20 3D-models.

Shimshoni and Ponce [SP00] present a probabilistic 3D object recognition algorithm where, in order to guide the recognition process, the probability that match hypotheses between image features and model features are correct is computed. They developed a model which uses the probabilistic peaking effect of measured angles and ratios of lengths by tracing iso-angle and iso-ratio curves on the viewing sphere. There model also accounts for various types of uncertainty in the input such as incomplete and inexact edge detection. For each match hypothesis, the pose of the object and the pose uncertainty, which is due to the uncertainty in vertex position, are recovered. This is used to end sets of hypotheses which reinforce each other by matching features of the same object with compatible uncertainty regions. A probabilistic expression

is used to rank these hypothesis sets. The hypothesis sets with the highest rank are output. The algorithm has been tested on real images but on a very small database.

In [FADV07], authors proposed the use of a probabilistic Bayesian method for 3-D-model retrieval from different views extracted from a given query. They consider that views do not have equal importance. That means, there are views which represent the 3-D model better than others. Each model of the collection is represented by a set of characteristic views. Considering a 3-D-request-model, the authors compute the probability of it corresponding to the query.

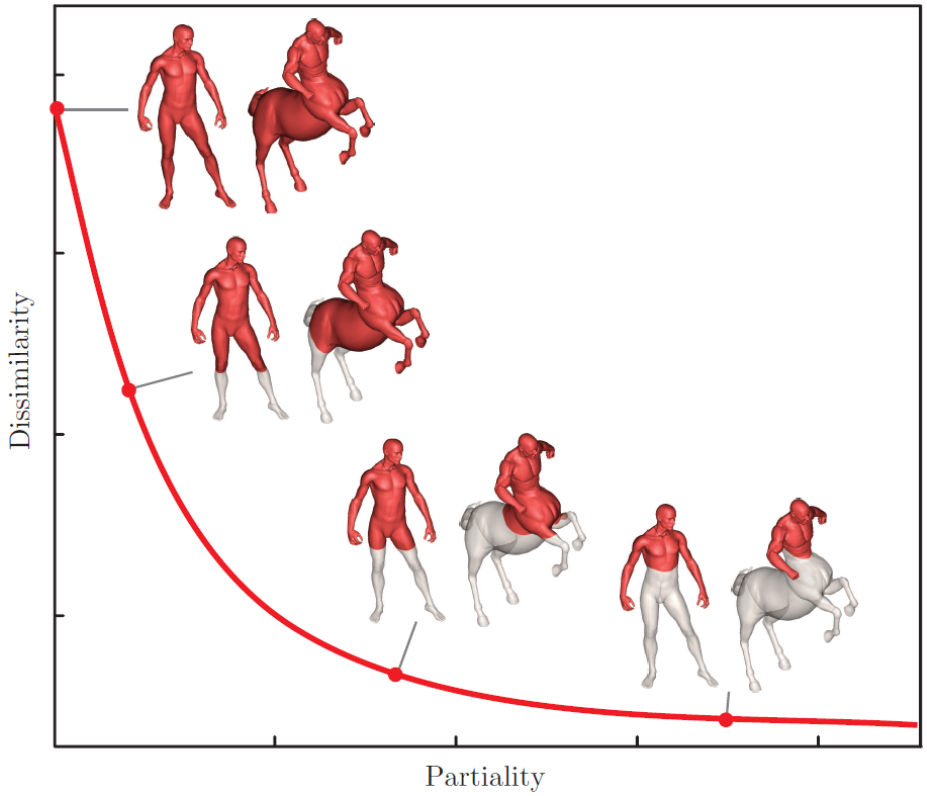


Figure 2.11 – *Illustration of the notion of Pareto optimality and set-valued distance. [BBBK08b]*

2.3.3 Pareto framework for Shape matching

The main goal of the work proposed by Bronstein et al [BBBKo8b], is to give a quantitative interpretations of what we mean by "large" and "similar", and draw a coherent relationship between these terms.

This enables to develop a computationally tractable difficulty of finding the biggest most similar parts. In their approach, authors use the formalism of Pareto optimality and multicriterion optimization Figure 2.11.

The authors show particular examples of partial similarity of rigid and non-rigid two and three-dimensional objects and text sequences, and elaborate the numerical aspects of their computation. In addition, they show the extension of their methods to images and shapes with texture.

Let X and Y be two shapes to be compared. The shape X and Y are partially matching if there exist parts $X' \subseteq X$ and $Y' \subseteq Y$ which are similar and significant. The degree of dissimilarity of parts X and Y can be expressed by a non-negative function $d : \sum_X \times \sum_Y \rightarrow \mathbb{R}_+$ (here $\sum_X$ and $\sum_Y$ denote the collection of all the parts of the shapes X and Y , respectively). The approach models a shape X as a metric space (X, d_X), where $d_X(x, x')$ is the geodesic metric, measuring the length of the shortest path between two points x, x' on X.

The similarity criterion used in [BBBKo8b] is similar to the stress function proposed in multidimensional scaling (MDS) problems [BBBKo8b] and also be related to the Gromov-Hausdorff distance [BBBKo8a] between metric spaces (X, d_X) and (Y, d_Y). As the measure of insignificance, Bronstein et al. used the partiality function

$$p(X') = area(X) - area(X') = \int_{XX'} d\mu_X$$

where μX indicates the area measure on X. In this formulation, authors

stated partial matching as a problem of simultaneous minimization of d and p over pairs of all the possible parts,

$$min_{X',Y'} \left(d(X', Y'), p(X') + p(Y') \right). \tag{2.1}$$

According to Bronstein et al [BBBK08b], the solution of the multi-criterion optimization problem in eq 2.1 is the set of parts (X^*, Y^*) achieving an optimal tradeoff between dissimilarity and partiality, in the sense that there exists no other pair of parts (X', Y') with both $d(X', Y') < d(X^*, Y^*)$ and $p(X') + p(Y') < p(X^*) + p(Y^*)$. This solution is called Pareto optimal (See Figure 2.11).

2.3.4 Bag of Features based methods

Recently, a resurgence of methods based on Bag of Feature techniques have been proposed in the literature. Liu et al. [LZQ06] 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 quantizing Spin images of the object. Ohbuchi et al. [OOFB08] introduced a view-based method using salient local features. They represented 3D-objects as word histograms derived from the vector quantization of salient local descriptors extracted on the depth-buffer views captured uniformly around the objects. Ovsjanikov et al. [OBBG09] 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. [TCF09] has used the BoF 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.

Conclusion

In this chapter, we have presented an overview of some existing techniques for 3D-shape description and similarity computing. The description techniques can be grouped into four categories. The global methods which take only the global characteristics of the shape, the view-based techniques, which consider that two 3D-shapes are similar when they are similar from different angles of view, the graph-based methods which consist to extract a graph (or skeleton) from the 3D-shape and reduce the problem of 3D-comparison to graph matching, and finally the local based descriptors which take into account the local characteristics of the shapes. The local-based method contains two steps namely feature extraction and feature description. The extraction of feature can be performed in a random way(extract randomly a set of points from the 3D-shapes), it can be done in an uniform way (uniform sampling of points in the shape) or it can be performed using sophisticated techniques that extract only interest points. The step of description consists to describe the set of points, already extracted, by using local descriptors. In section 2 of this chapter, we presented some existing techniques for similarity computing. This techniques which aim to match between two 3D-shape, are usually applied to 3D-retrieval and 3D-classification. In the next chapter, we present our proposed descriptors. Since, they are local-based descriptors, they need tow steps. The first step is for 3D-feature extraction and the second step for the description. For the description, we present two descriptor namely, *the curve descriptor* and the *geodesic cord descriptor*.

French conclusion

Dans ce chapitre, nous avons donné un aperçu sur quelques techniques existantes pour la description de la forme d'objets 3D ainsi que les différentes techniques utilisées pour la mesure de similarité.

Les techniques de description peuvent être regroupés en quatre catégories. Les méthodes globales, qui ne prennent que les caractéristiques globales de la forme de l'objets 3D, les méthodes basées sur les vues 2D, qui considèrent que deux formes d'objets 3D sont similaires s'ils le sont à partir des différents angles de vue, les méthodes basée sur les graphes et qui consistent à extraire un graphe (ou squelette) de la forme d'objet 3D et de réduire le problème de la comparaison d'objets 3D à un simple appariement de graphes, et enfin les méthodes basées descripteurs locaux qui prennent en compte les caractéristiques locales des objets 3D. Les méthodes locales contiennent en général deux étapes à savoir extraction des patches locaux et de la description de ces patches. Le processus d'extraction peut être effectué de façon aléatoire, uniforme (échantillonnage uniforme des points dans la forme) ou cela peut se réaliser en utilisant des techniques plus sophistiquées pour extraire des points d'intérêt qui donnent plus d'information sur l'objet que le reste des points. L'étape de la description consiste à décrire l'ensemble des points, déjà extrait, en utilisant un ensemble de descripteurs locaux. Dans la section 2 de ce chapitre, nous avons présenté quelques techniques existantes pour le calcul de similarité. Cette étape, qui visent à correspondre deux formes d'objets 3D, est généralement appliquée pour la recherche et 3D la classification d'objets 3D.

Dans le prochain chapitre, nous présentons les différents descripteurs qu' on propose pour analyser la forme des objets 3D.

3D-Object Features

3

Contents

In this chapter, we present two new shape descriptors. These descriptors belong to local-based description category. So that, a feature extraction step has to be performed before. This step is based on the concept of the feature points introduced by Tierny et al [TVD09]. The

method consists to extract features located in the prominent components of a given 3D-object. The descriptors, we propose are based on an intrinsic function computed around feature points. The first descriptor is based on curve analysis. It consists to represent a 3D-surface as a collection of 3D curves extracted uniformly. The second descriptor is a probability distribution that represents the shape of the 3D-object, as other similar descriptor for 3D-object recognition [OFCD02]. In the first section of this chapter, we present the feature point extraction algorithm. In Section 2, we details the curve descriptor. The third section presents the geodesic cord descriptor.

French introduction

Dans ce chapitre, nous présentons deux nouvaux descripteurs de forme d'objets 3D. Ces descripteurs appartiennent à la catégorie des descripteurs locaux. Une étape d'extraction de caractéristiques est necessaire avant la description.

Cette étape est basée sur le concept des points d'interêts proposés par Tierny et al [TVD09].

Les descripteurs que nous proposons sont basés sur des propriétés intrinsèque calculée autour des points caractéristiques. Le premier descripteur est basé sur l'analyse des formes des courbes. Il consiste à représenter une surface 3D par une collection de courbes 3D extraites de manière uniforme. Le second descripteur appelé *descripteur des cordes géodésiques* est une distribution de probabilité qui représente la forme de l'objet 3D.

Dans la première section de ce chapitre, nous présentons l'algorithme d'extraction des points caractéristiques. Dans la section 2, nous présentons le descripteur de la courbe. La troisième section présente le descripteur des cordes géodésiques.

3.1 Feature point extraction

The feature points is a concept, introduced by several authors [MP02, KLT05], for which it is hard however to find a formal definition. It refers to the points of a surface located at the extremity of its prominent components. Feature points can also be described as the set of points that are the furthest away (in the geodesic sense) from all the other points of the surface.

In this section, we present the algorithm used in the feature point extraction process. This algorithm is based on tools proposed by Tierny

et al. [TVD09]. Using a diversity of surfaces, the proposed algorithm produces robust and well-localized feature points.

Let v_1 and v_2 be the farthest vertices (in the geodesic sense) on a connected triangulated surface S. Let f_1 and f_2 be two scalar functions defined on each vertex v of the surface S , as follows: $f_1(v) = \delta(v, v_1)$ and $f_2(v) = \delta(v, v_2)$ where $\delta(x, y)$ is the geodesic distance between points x and y on the surface.

As mentioned by [CMEH*03], in a critical point classification, a local minimum of $f_i(v)$ is defined as a vertex v_{min} such that all its level-one neighbors have a higher function value. Reciprocally, a local maximum is a vertex v_{max} such that all its level-one neighbors have a lower function value. Let F_1 be the set of local extrema (minima and maxima) of f_1 and F_2 be the set of local extrema of f_2. We define the set of feature points F of the triangulated surface S as the closest *intersecting* points in the sets F_1 and F_2.

Figure 3.1 – *F_1, F_2 and F sets on a triangulated surface. [TVD09]*

In practice, f_1 and f_2 local extrema do not appear exactly on the same vertices but in the same geodesic neighborhood. Consequently, we define the intersection operator $\cap$ with the following set of con-

straints, where δ_n stands for the normalized geodesic distance function (to impose scale invariance) (See Figure 3.1):

$$V \in F = F_1 \cap F_2 \Leftrightarrow \begin{cases} \exists v_{F_1} \in F_1 \ / \ \delta_n(V, v_{F_1}) < \epsilon \\ \exists v_{F_2} \in F_2 \ / \ \delta_n(V, v_{F_2}) < \epsilon \\ \delta_n(V, v_i) > \epsilon \ \forall v_i \in F \\ \epsilon, \delta_n \in [0, 1] \end{cases}$$

This algorithm detects all the feature points required in the subsequent analysis. They are accurately localized and their localization is robust with respect to rigid and non-rigid transformations, because of the use of geodesic distance in f_1 and f_2 functions.

In the sequel, we will denote a 3D-surface extracted around one feature point by *3D-patch*.

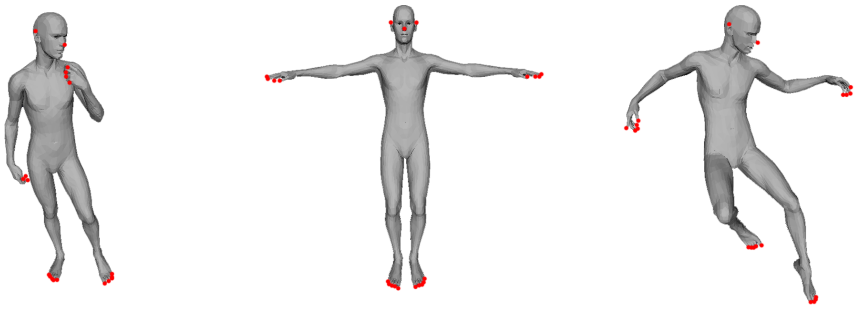


Figure 3.2 – *Feature points extracted from different poses of a 3D-model.*

3.2 Curve based descriptor

In this section, we analyze shapes of 3D-surfaces using shapes of curves extracted around a set of feature points. In other words, we divide a 3D-surface into a set of patches. Each patch consists of one feature point and an indexed collection of simple, closed curves in $\mathbb{R}^3$. The geometry of 3D-patches is then studied using the geometries of the associated curves. Curves are defined as level curves of an intrinsic distance func-

tion on the 3D-patch surface. Their geometries in turn are invariant to the rigid transformations of the 3D-patch surface. These curves jointly contain all the information about the surface and it is possible to go back-and-forth between the surface and the curves without any ambiguity. In this section, we firstly describe how to extract a significant set of curves around feature points. We explain then the framework of curves analysis and its extension to comparison of 3D-patch surfaces.

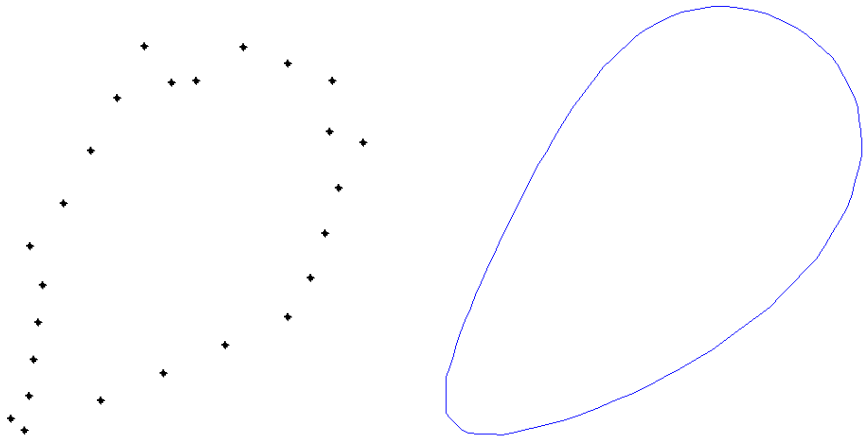
Let V_i be a feature point on a 3D-triangulated surface. A geodesic distance function is defined on that 3D-surface such that V_i be its origin. The geodesic function is split into a set of levels. Vertices which are in the same level are extracted with respect to an arbitrary order. For each level, we use the fast-marching method to extract the equidistant points. This is done by computing the geodetic distance from the feature point to all other vertices of the mesh. The equidistant curve is then formed by connecting linear segments between points on the edges at the given level (distance).

Let λ be a level set corresponding to the geodesic distance function f. The set of ordered vertices v such that $f(v) = \lambda$ builds one curve.

Figure 3.3 shows one level curve. Figure 3.3a shows the equidistant points to a given feature point. The set of obtained points is then arranged and oriented in order to build a 3D-closed curve(see Figure 3.3b). Figure 3.4 shows the sets of curves extracted from a cow 3D-patch. Figure 3.4a shows the original 3D patch extracted around one feature points (shown in red). Figure 3.4b shows the set of level curves extracted from the head cow patch around the feature point.

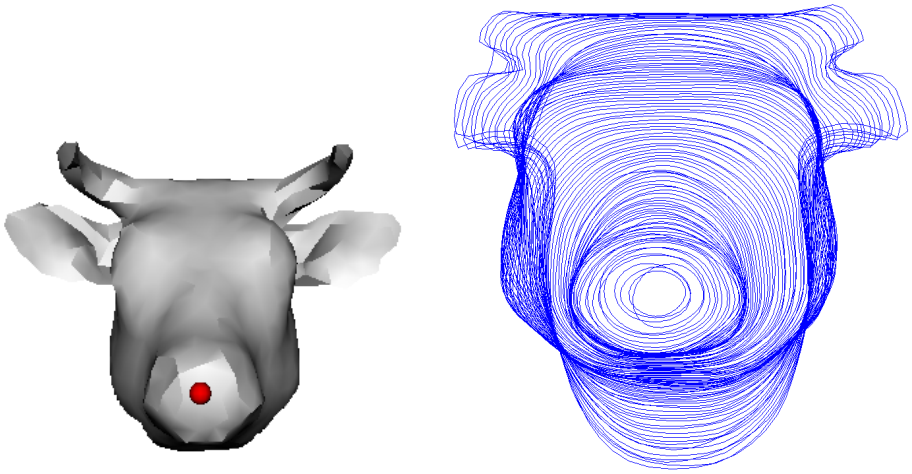
3.2.1 Curves analysis

The analysis of 3D-shapes via their associated curve space has a technical sound foundation in the Morse theory [LHGQ06]. In our approach,



(a) (b)

Figure 3.3 – *3D-curve extraction and smoothing.*



(a) (b)

Figure 3.4 – *Representation of patches by an indexed collection of curves.*

we treat curves as closed, parameterized in $\mathbb{R}^3$ with fixed origins for parameterizations 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, as one curve has to (locally) stretch, compress and bend to match the other.

Several authors, starting with Younes [Lau98], followed by Michor and Mumford [MM06] and others, have studied curves for planar shapes. More recently Joshi et al. [JKSJ07] have extended it to curves in $\mathbb{R}^n$ using an efficient representation of curves. Other authors, including Yezzi and Mennucci [YM05], 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 [JKSJ07] because it simplifies the elastic shape analysis. The main steps are: (i) define a space of closed curves of interest, (ii) impose a Riemannian structure on this space using the elastic metric, and (iii) compute geodesic paths under this metric. These geodesic paths can then be interpreted as optimal elastic deformations of curves.

We start by considering a closed curve β in $\mathbb{R}^3$. Since it is a closed curve, it is parameterizable using $\beta : \mathbb{S}^1 \rightarrow \mathbb{R}^3$. We will assume that the parameterization 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 parameterization is not assumed to be arc-length; we allow a larger class of parameterizations for improved analysis. To analyze the shape of β, we shall represent it mathematically using a *square-root velocity function* (SRVF), denoted by $q(t)$, according to:

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

$q(t)$ 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:

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

which is the length of β. Therefore, the $\mathbb{L}^2$-norm is convenient to analyze curves of specific lengths. Secondly, as shown in [JKSJ07], 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:

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

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:

$$\langle u, v \rangle = \int_{S^1} \langle u(t), v(t) \rangle \, dt \ . \tag{3.1}$$

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:

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

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

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

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-parameterized; a re-parameterization 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-parameterization group Γ as the equivalence classes in $\mathcal{C}$. Here, Γ is the set of all orientation-preserving diffeomorphisms of $\mathbb{S}^1$ (to itself) and the elements of Γ are viewed as re-parameterization functions. For example, for a curve $\beta : \mathbb{S}^1 \to \mathbb{R}^3$ and a function $\gamma : \mathbb{S}^1 \to \mathbb{S}^1$, $\gamma \in \Gamma$, the curve $\beta(\gamma)$ is a re-parameterization 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:

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

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

construction of a geodesic between any two elements of $\mathcal{S}$, under the Riemannian metric given in Eqn. 3.1. 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 [KS06]. 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

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

.

The interesting part is that the gradient of E has been derived analytically and can be used directly for updating α. As shown in [KS06], 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 denote the geodesic distance, or the length of the geodesic in $\mathcal{S}$, between the two curves β_1 and β_2. Shown in the bottom row of Figure 3.5 is an example of geodesic path between two 3D-curves extracted from two different 3D-patch surfaces; the cow-head patch (left side) and the horse-head patch (right side).

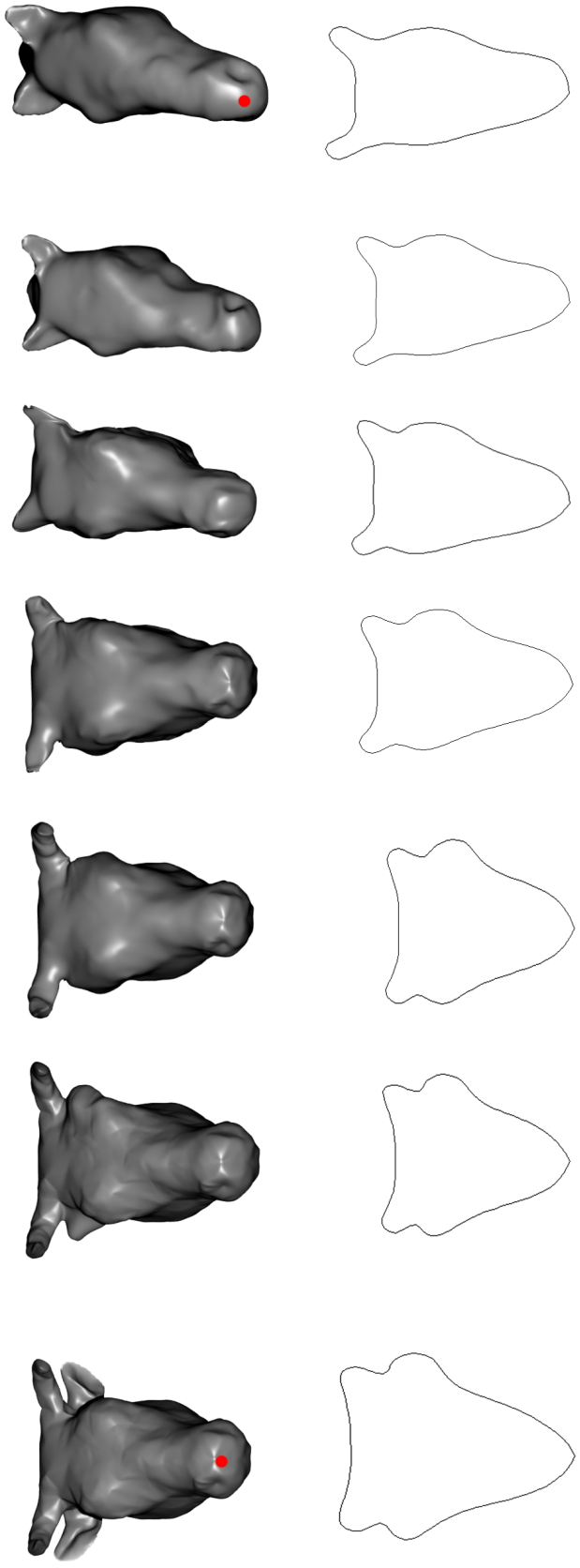


Figure 3.5 – *Geodesic path between cow-head patch and horse-head patch.*

3.2.2 3D-patch matching

Now we extend ideas developed in the previous section for analyzing shapes of curves to the shapes of full 3D-patch surfaces. As mentioned earlier, a 3D-patch surface P is represented with an indexed collection of the level curves of the f function. That is, P is equivalent to the set $\{c_\lambda \in [0, L]\}$, where c_λ is the level set associated with the distance function value equal to λ. Through this relation, each 3D-patch has been represented as an element of the set $\mathcal{C}^{[0,L]}$. In our framework, the shapes of any two patches are compared by comparing their corresponding curves. Lets D denotes the global distance between two patches.

Given any two surfaces patches P_1 and P_2, and their collection of curves $\{c^1_\lambda, \lambda \in [0, L]\}$ and $\{c^2_\lambda, \lambda \in [0, L]\}$, respectively, our idea is to compare the curves c^1_λ and c^2_λ, and to accumulate these distances over all λ. Formally, we define a distance: $D : \mathcal{C}^{[0,L]} \times \mathcal{C}^{[0,L]} \rightarrow \mathbb{R}_{\geq 0}$, given by

$$D(P_1, P_2) = \int_0^L d(c^1_\lambda, c^2_\lambda) d\lambda.$$

Here, the distance inside the integral is the geodesic distance function between the shapes of any curves, described in the last section. In Figure 3.5, The top row shows the two 3D-patches surfaces (cow-head and horse-head) and the resulting geodesic path between them. Middle surfaces denote five equally spaced points (in the space of all parameterized paths) along the geodesic path. With respect to the chosen Riemannian metric, this path denotes the optimal deformation from the cow-head patch to the horse-head patch.

3.2.3 Karcher Means of surfaces

The Riemannian structure defined on the manifold of 3D-surfaces enables us to perform some statistical analysis for computing 3D-surfaces mean and variance. The *Karcher mean* is based on the intrinsic geome-

try of the manifold to define and compute a mean on that manifold. It is defined as follows. Let $D(P_i, P_j)$ denotes the length of the geodesic from a 3D-patch P_i to another 3D-patch P_j. To calculate the Karcher mean of a set of 3D-patches $\{P_1, ..., P_n\}$, we define the variance function $\vartheta : \mathcal{C}^{[0,L]} \longrightarrow \mathbb{R}, \vartheta(P) = \sum_{i=1}^{n} D(P, P_i)^2$. The Karcher mean is then defined by :

$$T = argmin_{P \in \mathcal{C}^{[0,L]}} \vartheta(P) \tag{3.2}$$

The intrinsic mean may not be unique, i.e. there may be a set of points in $\mathcal{C}^{[0,L]}$ for which the minimizer of ϑ is obtained. To interpret geometrically, T is an element of $\mathcal{C}^{[0,L]}$, that has the smallest total deformation from all given 3D-surfaces.

Figure 3.6 shows the shape of the Karcher mean of four 3D-objects. The mean shape is located at the center of the figure.

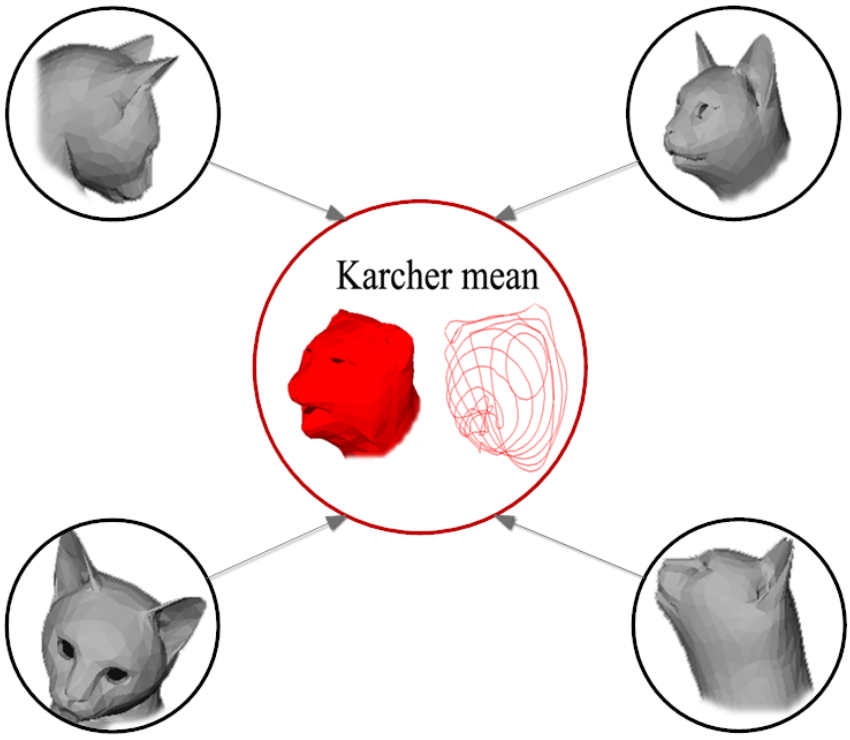


Figure 3.6 – *The shape of the Karcher mean of four 3D-objects and its associated sets of curves*

3.3 Geodesic cord descriptor

In this section, we present another descriptor. It represents the shape signature for a 3D-patch as a probability distribution sampled from a shape function measuring intrinsic properties of the 3D-patch. We call this generalization of intrinsic histogram a geodesic cord descriptor. The geodesic cord descriptor is a probability distribution that represents the shape of the 3D-patch as other similar approaches for 3D-object recognition [PR99, OFCD02].

The distribution is sampled from an intrinsic distance function on the 3D-surface. The distance function used in here is based on the geodesic distances between the feature point and all the points on the 3D-patch surface. It is in turn invariant to the rigid and non-rigid transformations of the surface.

This descriptor can be computed quickly and easily. Once we have computed the shape distributions for two objects, the dissimilarity between the 3D-patches can be evaluated using any metric that measures distance between distributions (e.g., L_n norm), possibly with a normalization step for matching scales.

The key idea is to transform an arbitrary 3D-patch into a parameterized function that can easily be compared with others. In our case, the domain of the shape function provides the parameterization. The primary advantage of this approach is its simplicity. The 3D-patch shape matching problem is reduced to a comparison of probability distributions, which are relatively simple tasks when compared to prior methods that required reconstructing a solid object or manifold surface.

In spite of its simplicity, we expect that this descriptor is useful for discriminating 3D-patches with different shapes. This hypothesis satisfies several properties desirable for a shape matching method:

Invariance: Geodesic cord descriptor has all the transformation invariance properties of the sampled shape function. For instance, the intrinsic function yields invariance under rigid and non-rigid transformations. In this case, invariance under scaling can be added by normalization of Geodesic cord distributions before comparing them and/or by factoring out scale during the comparison.

Metric The dissimilarity measure produced by this descriptor adopts the properties of the norm we use to compare Geodesic cord distributions. In particular, if the norm is a metric, so is our dissimilarity measure. This property holds for most common norms, including L_n norms, Earth Mover's Distance, and so forth.

In the following sections, we provide a detailed description of the methods we use to build geodesic cord descriptors from 3D-patches and compute a measure of their dissimilarities.

3.3.1 Selecting a Shape Function

The first and most interesting issue is to select a function whose distribution provides an accurate signature for the shape of a 3D-patch. Ideally, the distribution should be invariant under rigid and non rigid transformations, and it should be insensitive to noise. Here we choose to use the geodesic distance between the feature point, associated to the 3D-patch, and the rest of points in the patch.

3.3.2 Constructing of Geodesic cords

Having chosen the geodesic distance as the intrinsic function, the next issue is to compute and store a representation of its distribution. Specifically, we evaluate this function in the 3D-patch and construct a histogram by counting how many points fall into each of R fixed sized

bins. The histogram forms our representation for the Geodesic cord distribution. We compute the Geodesic cord descriptor once for each model and store it as a sequence of R integers. More formally, given a 3D-patch P associated to the feature point F_i, we define a descriptor $P(F_i)$ and consider the geodesic distances $\{d(F_i, v); \forall v \in V\}$ with V is the set of all the vertices on the 3D-patch surface. Considering f the distribution of vertices according to these distances, we define the descriptor $P(F_i)$ as an R-dimensional vector:

$$P(F_i) = (p_1, ..., p_R)$$

where

$$p_r = \int_{(r-1)/R}^{r/R} f(d)\delta d$$

$P(F_i)$ is an R-bin histogram of vertex distribution of geodesic distances measured from F_i.

3.3.3 Comparing Geodesic cord descriptors

Having constructed the shape distributions for two 3D models, we are left with the task of comparing them to produce a dissimilarity measure. There are many standard ways of comparing two descriptors P_1 and P_2 representing probability distributions.

The Minkowski norms The Minkowski norms, also called L_n norms, are vector norms. The most widely used are the 1-norm, 2-norm, and ∞-norm:

$$\|P_1 - P_2\|_1 = (|P_1(1) - P_2(1)| + ... + |P_1(n) - P_2(n)|)$$

$$\|P_1 - P_2\|_2 = \sqrt{\left(|P_1(1) - P_2(1)|^2 + ... + |P_1(n) - P_2(n)|^2\right)}$$

$$\|P_1 - P_2\|_\infty = max_{1 \leq i \leq n} |P_1(i) - P_2(i)|$$

Kullback-Leibler divergence For probability distributions, P_1 and P_2, the KL is defined to be

$$KL(P_1, P_2) = \sum_i P_1(i) \cdot log(P_1(i)/P_2(i))$$

Bhattacharyya distance Bhattacharyya distance can be also used to compare two geodesic cord descriptors, P_1 and P_2:

$$D = 1 - \sum_i \sqrt{P_1(i)}\sqrt{P_2(i)}$$

Many other distance can be used also to compare to descriptors such as Kolmogorov-Smirnov distance and Earth Mover's distance. In our implementation, we have experimented many distance measure and we find that the L_2 distance is the most efficient in both speed and performance.

Figure 3.7 shows the descriptor corresponding to four feature points extracted from two different cats. We can notice that the tail-patch descriptor Figure 3.7a of the first cat is similar to the tail-patch descriptor Figure 3.7b of the second cat. The leg-patches of the two cats (Figure 3.7c and Figure 3.7d) also have similar distance distributions.

3.4 Experiments and Results

To assess the efficiency of the proposed algorithms, in this section we evaluate their performance. In the first, we study the robustness of the proposed algorithm for the feature point detection. Then, we compare the performance of the two proposed descriptors. The comparison is conducted on a dataset composed of 50 mesh patches.

3.4.1 Feature point evaluation

To test the affine transformation invariance and the robustness to noise, we made several experiments. The criterion to evaluate the experiments

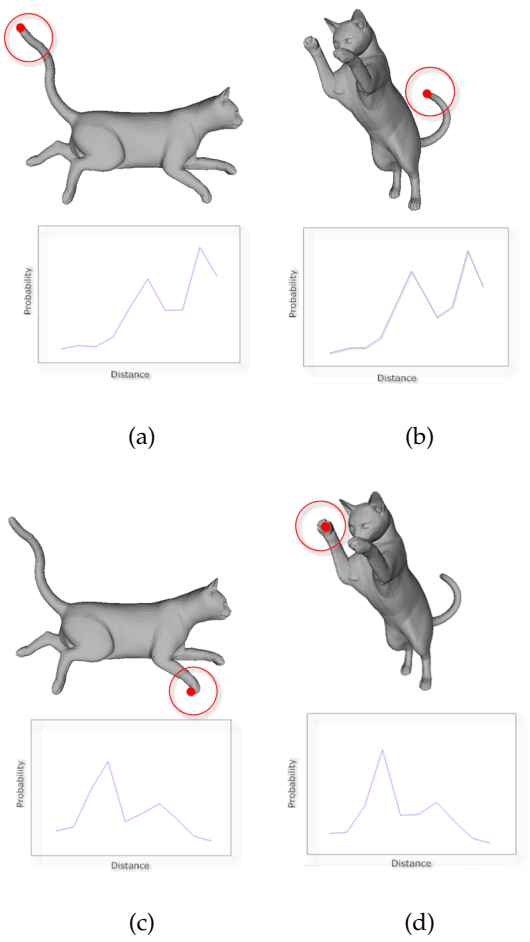


Figure 3.7 – *Distance-distributions on four different feature points.*

was the repeatability of the feature points. Given an object O and a transformation function T, which can be a translation, scaling, rotation, isometric, or noise addition, $T(O)$ is a transformed object. In addition, F represents the set of feature points extracted from O. Thus, the repeatability is defined as:

$$R_{O,T(O)} = \frac{\left|P_O \cap P_{T(O)}\right|}{|P_O|} \tag{3.3}$$

With respect to the used data, we pick models from the TOSCA dataset[1]. We calculate the repeatability between the null object and each transformed object, obtaining an average for each object in our collection. Finally, we calculate the mean of average repeatabilities of each object. For the whole collection, we obtained a repeatability of 0.98.

3.4.2 Descriptor performance evaluation

In this section, we provide a set of experimental results to assess the effectiveness of the proposed shape descriptor methods. To this purpose, we are using a pre-classified test database of 50 mesh patches grouped in 5 classes, each one consisting of 10 elements. See Figure 3.9.

We evaluated the two descriptors using the standard measures briefly described below.

Precision and recall The Precision and recall are two fundamental measures often used in evaluating search strategies. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database, while precision is the ratio of the number of relevant records retrieved to the size of the return vector. In our experiment, for each query the total number of relevant records in the database is always 10, that is the size of each class. Figure 3.10 shows

[1] http://tosca.cs.technion.ac.il/

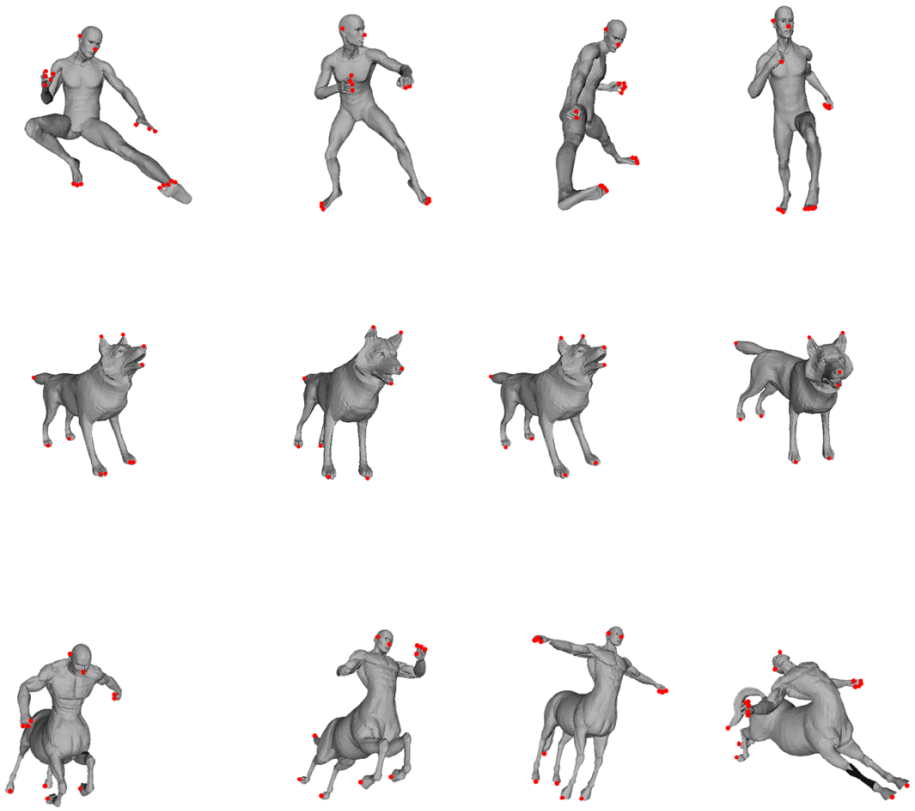


Figure 3.8 – *Example of feature point position with respect to isometric transformations. The feature points are colored in red. The shapes are taken from the TOSCA dataset.*

Figure 3.9 – *Five 3D-patches taken from the evaluation dataset. The shapes are taken from the TOSCA dataset.*

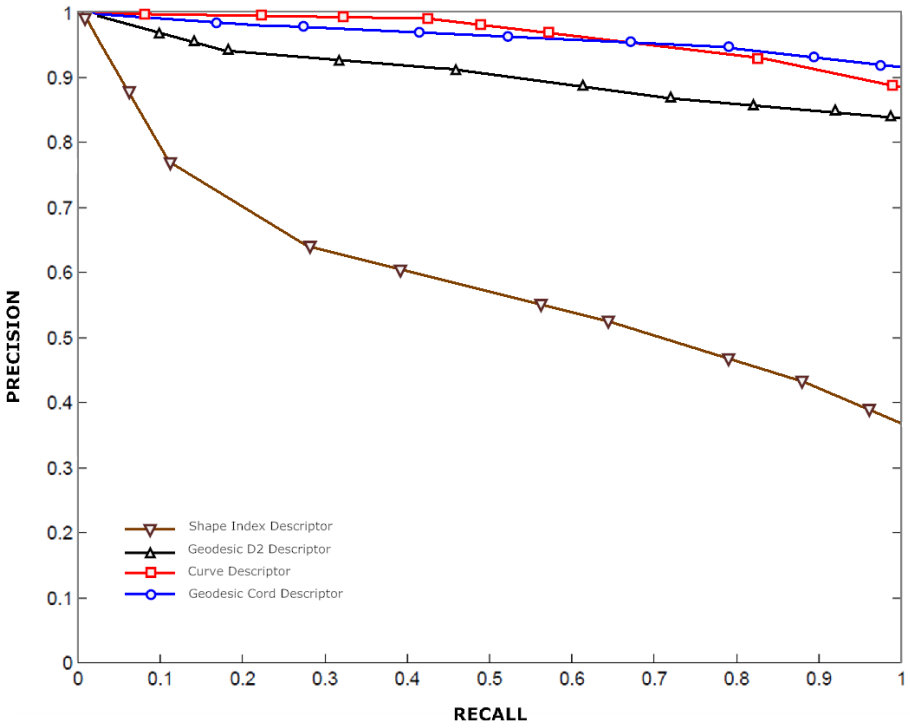


Figure 3.10 – *Precision vs Recall plots comparing the Curve descriptor and Geodesic Cord descriptor with other descriptors.*

the Precision vs Recall plots. It can be seen that the curve descriptor performs the best results in term of precision. The geodesic cord is quite effecient compared to the shape index descriptor and the geodesic D2 one

Percentage of success We compute the percentage of success for the first (PF) and the second (PS) retrieved items, i.e. the probability of the first and second elements in the return vector to be relevant to the given query, and average them over the whole set of queries. For an ideal method PF = PS = 100%.

Returning to the whole collection, we observed that all descriptors guarantee the identity property (i.e. $D(P_i, P_i) = 0$, P_i is the ith patch in the database), so that the percentage of success PF of the first retrieved

Descriptor	Curve	Geodesic Cord	Geodesic D2	Shape index
PF	100%	100%	100%	100%
PS	100%	99.66%	98.72%	89%

Table 3.1 – *The percentages of success PF et PS*

item is always 100%. The percentages of success PS the second retrieved item are listed in the table 3.1.

Conclusion

In this chapter, we have presented a new method for 3D-shape feature extraction and description. First, we presented an automatic algorithm for feature point extraction. The algorithm detects points located in the prominent component of 3D-shapes. Then, we presented a patch descriptor based on curve analyses. The descriptor represents a collection of 3D-curves extracted around one feature point already extracted. The shape of the 3D-patch is studied then by analyzing the shapes of its curves. For that, we have presented a Riemannian framework for curve comparison. We have also presented another shape descriptor which is based on a probability distribution. The distribution is sampled from an intrinsic distance function on the 3D-surface. The distance function used is based on the geodesic distances between the feature point and all the points on the 3D-object surface. We have evaluated the proposed descriptors and we demonstrated their effectiveness compared with some well known descriptors.

In the next chapter, we will present a new method for 3D-shape similarity computing. We will use the belief function technique to compute a global similarity between two given 3D-objects. For this end, 3D-patches are considered as information sources. The global similarity is computed by merging all information.

French conclusion

Dans ce chapitre, nous avons présenté une nouvelle méthode pour la description d'objets 3D. Premièrement, nous avons présenté un algorithme automatique pour l'extraction de points caractéristiques. Ensuite, nous avons présenté un descripteur de forme 3D basé sur des courbes 3D. Le descripteur représente une collection de 3D-courbes extraites autour d'un point caractéristique déjà extrait. La descripteur est évalué en analysant les formes des courbes qui lui sont associées. L'analyse des formes des courbes est effectuée dans un cadre riemannien.

Nous avons également présenté un deuxième descripteur de forme 3D. Ce dernier est basé sur la distribution des distances géodésique entre un point caractéristique et tous les points de surface d'objets 3D.

L'évaluation des descripteurs proposés démontre leur efficacités par rapport à certains descripteurs de l'état de l'art.

Dans le chapitre suivant, nous présentons une nouvelle méthode pour la mesure de similarité entre les objets 3D.

4 Belief function based system for 3D-object retrieval

Contents

This chapter presents two belief based methods for 3D-object retrieval. First, we review needed belief function techniques for information fusion. Then, we present the methods and finally, we present the experimental results.

4.1 Introduction

This chapter presents an approach for 3D-object matching based on the use of information extracted from the different patches of the 3D-object. Each patch provides an information source regarding the shape of that object. A combination of these information is necessary in order to increase the recognition rate of the object. Patch information fusion can be seen as a problem of merging classification results of each patch (or fusion of classifiers). Conventional methods given by the theory of the uncertainty (fusion by weighted voting, Bayesian fusion, belief function based fusion, etc.) can be used to achieve this fusion. In these theories of uncertainty, two distinct concepts are required to fully model the imperfections of the information given by each patch: the uncertainty and imprecision. The uncertainty characterizes a degree of conformity to reality (qualitative lack of information), while the imprecision measures the quantitative missing of information (example measurement errors). In addition, the merger process should allow to take into account the relationship between the different sources of information (discrepancies, conflict, complementarity, refinement, etc.). Figure 4.1 summarizes the fusion process.

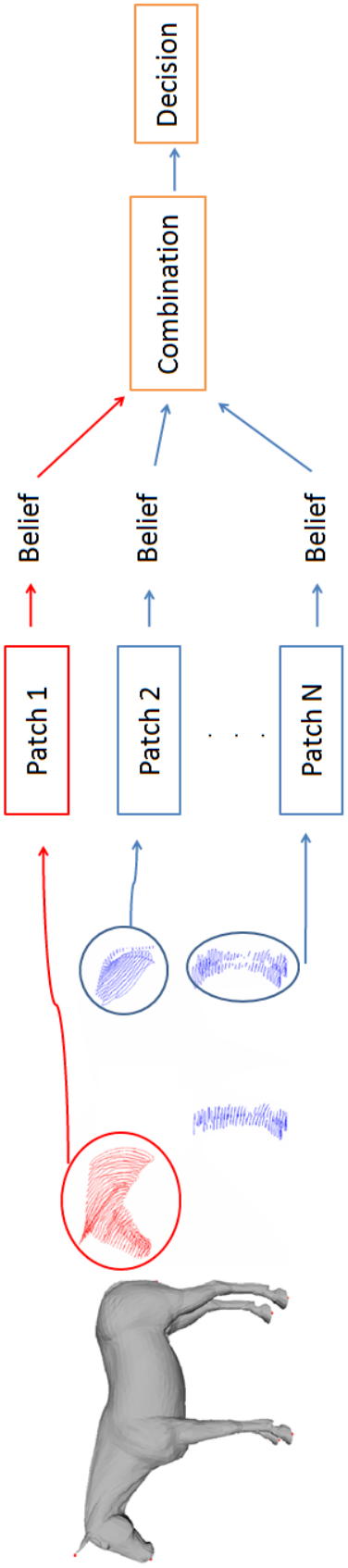


Figure 4.1 – *Patch evidence fusion.*

In the context of this thesis, we chose to work with the belief function theory. Indeed, this generic formalism includes several classical theories such as probability and possibility theory. Belief functions offer many mechanisms and tools to manage uncertainty, model disregard and take into account partial information. Its flexibility and ability to integrate concepts on the reliability of different sources of information contributes to conclude on its usefulness in our matching task.

First, a theoretical study of belief functions is presented. Then, we show its application in the 3D-retrieval context.

French introduction

Nous présentons dans ce chapitre une approche de fusion crédibiliste pour la recherche d'objets 3D à partir des différents descripteurs locaux calculés dans le chapitre précédant. Ces descripteurs sont considérés comme des sources d'informations qui peuvent être combinées pour fournir une decision sur la forme globale d'un objet 3D. La représentation et la combinaison de ces informations sont effectuées dans le cadre des fonctions de croyances.

Tout d'abord, une vue d'ensemble théorique sur les fonctions de croyance est présenté. Ensuite, nous montrons son application dans le cadre de la recherche d'objets 3D.

4.2 Belief function theory

The work of Dempster [Dem67] on the lower and upper bounds of a family of probability distributions has allowed Shafer [Sha76] to lay the foundations of the theory of belief functions. This theoretical formalism may take several interpretations and designations (theory of belief functions, theory of Dempster-Shafer, Evidence theory). Shafer has shown

the benefits of belief functions to model uncertain knowledge. The usefulness of these functions, as an alternative to subjective probabilities, has been demonstrated later axiomatically by Smets [SK94] through the *Transferable Belief Model* (TBM).

The TBM is a generic framework developed for the representation and the combination of knowledges. It is based on the definition of belief functions provided by several sources of information that can be complementary, redundant, and possibly non-independent. It proposes a set of operators for combining these functions. It is naturally used in the context of information fusion to improve the analysis and the interpretation of data taken from multiple information sources.

One of the fundamental points that characterizes the TBM is the differentiation of levels of knowledge representation and decision taking. This differentiation is much less prominent in other approaches, particularly for the probabilistic model for which the decision is often the only objective. The reasoning mechanisms of the TBM are grouped into two levels:

- The credal level, where knowledge is represented (static part), combinations and reasoning on this knowledge is performed (dynamic part).
- The pignistic level which involves a final decision by taking into account the possible risk and/or the gain associated with this decision.

In the following, we discuss some mathematical elements related to the belief functions. The concept of belief function is first presented, allowing the reader to understand how information is represented in a belief structure. Some tools associated with this formalism are then detailed, including combination and decision making.

4.2.1 Belief functions

Given a finite set $\Omega = \{\omega_1, \omega_2, ..., \omega_q\}$, usually called a frame of discernment. A belief function *bel* is a function defined from 2^Ω to $[0,1]$ by:

$$bel(\emptyset) = 0$$

$$\forall l \geq 1, \forall i = 1, ..., l \; A_i \subseteq \Omega :$$

$$bel(\bigcup_{i=1,...l} A_i) \geq \sum_{I \subseteq \{1,...l\}, I \neq \emptyset} (-1)^{|I|+1} bel(\bigcap_{i=1,...l} A_i) \tag{4.1}$$

The particular case where $l = 2$ and $A_1 \cap A_2 = \emptyset$ illustrates the fact that the belief attributed to the union of two disjoint subsets of Ω is greater than or equal to the sum of the beliefs assigned to each subset taken separately. If the inequality is transformed into equality in equation 4.1, the resulting function *bel*, usually called Bayesian belief function, becomes a function of probability. Thus, the classical probability is a special case of belief functions which can be interpreted as a generalization of probability theory to model inaccuracies and uncertainties.

A belief function can also be defined by a mass function, denoted m defined from 2^Ω to $[0,1]$ which satisfies:

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{4.2}$$

In contrast to the probability distribution the mass function does not preserve the monotony with respect to inclusion:

$$A \subset B \nRightarrow m(A) < m(B). \tag{4.3}$$

This makes the belief function superadditive. Each subset $A \subseteq \Omega$ such as $m(A) > 0$ is called a focal element of m. Thus, the mass $m(A)$ represents the degree of belief assigned to the proposal A that could not, given the state of knowledge, be assigned to a more specific subset than

A. A function such that $m(\emptyset) = 0$ is called normal. In the *Transferable Belief Model* (TBM) [SK94], the condition $\sum_{\emptyset \neq A \subseteq \Omega} m(A) = 1$ is assumed and $m(\emptyset) > 0$ is accepted. This specificity allows us to introduce the concept of open world assuming that the belief can not be placed on a subset of Ω. In this context, the empty set $\emptyset$ can be interpreted as a hypothesis which is not clearly defined throughout the frame of discernment in contrast to the closed world where all hypotheses is exhaustive. Given a mass function m, we can actually define *bel* the belief function and *pl* the plausibility function respectively as following:

$$bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B), \forall A \subseteq \Omega \tag{4.4}$$

and

$$pl(A) = \sum_{A \cap B \neq \emptyset} m(B), \forall A \subseteq \Omega \tag{4.5}$$

The functions m, *bel* and *pl* represent three facets of the same information.

4.2.2 Information Fusion

Let us assume that we have two mass functions m_1 and m_2 defined on the same frame of discernment Ω. These two functions can be aggregated by a conjunctive combination operator noted $\bigcirc$. The result of this operation leads to a single belief function which corresponds to a mass function, denoted by $m_{\bigcirc}$, which can be defined by:

$$m_{\bigcirc}(A) = (m_1 \bigcirc m_2)(A) = \sum_{B \cap C = A} m_1(B) \cdot m_2(C). \tag{4.6}$$

This combination rule is sometimes referred as unnormalized Dempster's combination rule. If necessary, the normalization assumption $m_{\bigcirc}(\emptyset) = 0$ can be found by dividing each mass by an appropriate

coefficient. The resulting operator, which is known as the Dempster's rule and noted $m_{\oplus}$, is defined by:

$$(m_1 \oplus m_2)(A) = \frac{(m_1 \textcircled{\cap} m_2)(A)}{1 - m_{\textcircled{\cap}}(\emptyset)} \; \forall A \subseteq \Omega. \tag{4.7}$$

The use of the Dempster's rule is possible if the mass functions m_1 and m_2 are not in total conflict. That means if there exist two focal elements B and C respectively, of m_1 and m_2 such that $B \cap C \neq \emptyset$. This rule allows to combine uncertain information modeled by belief functions.

4.2.3 Reliability and discount rate

When the source of information, the belief function is extracted from, is not completely reliable, it is possible to introduce an operation of weakening. In this case, a coefficient α which represents a kind of meta-knowledge about the reliability source, can transfer part of the belief to the set Ω. Thus, a mass function weakened, denoted m_α, can be deduced from m:

$$\begin{aligned} m_\alpha(A) &= \alpha \cdot m(A), \; \forall A \subsetneq \Omega \\ m_\alpha(\Omega) &= 1 - \alpha + \alpha \cdot m(\Omega) \end{aligned} \tag{4.8}$$

Thus, the belief function is able to represent several kind of knowledge and by this means it represent a rich and flexible framework for representing uncertain information. The coefficient α can be calculated using a statistical learning algorithm as in [SK94].

4.2.4 Decision making

At the pignistic level, the belief function resulting the available information in credal level, is used for decision making. Based its reasoning

on arguments of rationality developed in the transferable belief model, Smets [SK94] proposed to transform a mass function m into a probability function $BetP_m$ defined on Ω (called pignistic probability) that formalizes for all ω_k by:

$$BetP_m(\omega_k) = \frac{1}{1 - m(\emptyset)} \sum_{A \in 2^\Omega / \omega_k \in A} \frac{m(A)}{|A|}. \tag{4.9}$$

Where $|A|$ represents the cardinality of the subset $A \subseteq \Omega$ and $k \in \{1, 2, ..., q\}$. In this transformation, the mass of belief $m(A)$ is uniformly distributed among the different elements of A.

4.3 Belief function based system for 3D-object retrieval

In this section, we present the techniques used in our work to perform the fusion process of patch information. Based on two frame of discernments, we have developed two methods for 3D-shape retrieval. The first method is based on $\Omega = \{relevant, not - relevant\}$ as frame of discernment. The second method is based on $\Omega = \{O_j, 1 \leq j \leq M\}$ the set of all object in the dataset. We detail these methods in the following sections separately.

4.3.1 First retrieval method

The belief functions are used in this system to model ignorance and combine different information sources (given by each patch) using the combination rule of Dempster in a frame of discernment $\Omega = \{relevant, not - relevant\}$ where *relevant* means that a given 3D-object is relevant to index with respect to the query. We detail the sources of information used in this framework.

Information Sources

From the different patches extracted from the object query, we can compute information that can be combined to make a decision regarding the 3D-object. The idea consists to go from a simple patch matching to multi-patches in order to get more information about the whole object. Given an object query Q and its corresponding patches, the system computes information provided by each patch using the distance between the patch descriptor in the query and its closest one in the object to be compared with. Then a combination of these information is achieved. So that, the objects in the collection can be turned back to the user based on the set of patches information.

Given the query object Q and a set of P corresponding patches $\{P_i\}_{1\leq i\leq N}$ Information sources $\{S_1, S_2, ..., S_N\}$ derived from the query patches $\{P_i\}_{1\leq i\leq N}$ with the reliabilities $\{\alpha_1, \alpha_2, ..., \alpha_N\}$, where $\alpha_1, \alpha_2, ..., \alpha_N$ quantify the degree of reliabilities of the information sources extracted from the query patches. The reliability can be expressed as the significance of the patch in the object query. Finding a quantitative measure of the significance is a big challenge. As mentioned by Bronstein et al [BBBK08b], the main problem comes from the fact that unlike similarity between patches, which in many cases can have a strict definition, the significance is rather a semantic notion and thus much more difficult to quantify.

More formally, let $\Omega = \{relevant, not-relevant\}$ a frame of discernment. So, we have three possible focal elements, two assumptions ("object is relevant" and "object is not relevant") and a composite assumption Ω also called uncertainty. Each information source $S_i \in \{S_1, S_2, ..., S_N\}$ examines each object in the collection: is it a relevant object, not-relevant object or is there no opinion for this object O_j? For this reason, we propose a basic belief assignment $m[S_i, O_j]$,

which distributes the belief quantity between three elements: relevant, not-relevant and uncertainty Ω and satisfies:

$$m[S_i, O_j](relevant) + m[S_i, O_j](not - relevant) + m[S_i, O_j](\Omega) = 1 \quad (4.10)$$

The quantity of belief given to the assumption "relevant" by S_i to O_j must be proportional to the distance between the patch $P_i \in Q$ and its closest one $\in O_j$. This distance, noted by $d(P_i, O_j)$, can be computed by the measure of the distance presented in chapter 3. Formally d is given by:

$$d(P_i, O_j) = \min_{k, P_k \subset O_j} (d(P, P_k)) \quad (4.11)$$

The basic belief assignment $m[S_i, O_j](.)$ which is discounted by a coefficient α_i according to the reliability of source S_i can be computed as follows:

$$\begin{aligned} m^{\alpha_i}[S_i, O_j](relevant) &= \alpha_i \cdot (1 - d(P_i, O_j)) \\ m^{\alpha_i}[S_i, O_j](not - relevant) &= \alpha_i \cdot d(P_i, O_j) \\ m^{\alpha_i}[S_i, O_j](\Omega) &= 1 - \alpha_i \end{aligned} \quad (4.12)$$

Here, we define the reliability of a shape's patch as a function of the partiality function used in [BBBK08b]. The value α_i of a 3D-patch can be computed as $\alpha_i = exp(-npartiality(P_i)^2)$ where $npartiality$ is the normalized partiality function of the 3D-patch P_i. It is given by $npartiality(P_i) = \frac{area(Q) - area(P_i)}{area(Q)}$.

For each object O_j in the collection, we combine all basic belief assignments using the Dempster's combination rule. The aggregated mass function is:

$$m[Q, O_j] =$$

$$m^{\alpha_i}[S_i, O_j] \otimes m^{\alpha_2}[S_2, O_j] \otimes m^{\alpha_3}[S_3, O_j] ... \otimes m^{\alpha_N}[S_N, O_j] \tag{4.13}$$

After the discounting and combination phases, the system computes the pignistic probability. Finally, the system decides what the most similar object (relevant object) is, by selecting the object which has the biggest value of pignistic probability for the subset "relevant".

This method is summarized in algorithm 1

Algorithm 1: : 3D-shape search algorithm (1st method)

Given a 3D-object query Q.

1: *Extract a set of N feature points V_i from Q.*
2: **for** *each object O_j in the collection* **do**
3: **for** *each feature point V_i where $i = 1, ..., N$* **do**
4: *Compute a patch descriptor (ie. P_i).*
5: *Compute the distance between P_i and O_j using eq 4.11.*
6: *Compute a mass distribution for O_j using eq 4.12.*
7: **end for**
8: *Combine all masses m_{P_i} into a new mass function m_Q*
9: *Compute the pignistic probability induced by m_Q*
10: *Save the pignistic probability given to the Rel hypothesis.*
11: **end for**
12: *Display 3D-objects in the order according to the pignistic probability given to Rel.*

4.3.2 Second retrieval method

The second method developed in our work, consists to consider a different frame of discernment. Here, we consider a collection of M 3D-objects $\Omega = \{O_j, 1 \leq j \leq M\}$ and a set of N 3D-object patches $\{P_i, 1 \leq i \leq N\}$ extracted from a global 3D-object query Q. Our goal is to compute a global similarity metric between the 3D-object query Q and 3D-objects in the collection. This metric is based on the distance between patches presented in chapter 3 In order to achieve this compu-

tation, each patch P_i is modeled by one mass function m_{P_i}. This function is defined on Ω as frame of discernment and it measures the amount of belief committed to the assumption: "P_i belongs to O_j". Formally, m_{P_i} is a mass function $2^{\Omega} \rightarrow [0,1]$ given by:

$$m_{P_i}(O_j) = \alpha_i \cdot \frac{1 - d(P_i, O_j)}{\sum_{l=1}^{M}(1 - d(P_i, O_l))} \tag{4.14}$$

$$m_{P_i}(\Omega) = 1 - \alpha_i$$

where $d(P_i, O_j)$ is a distance measure defined between an object patch P_i of the query Q and a global 3D-object O_j in the collection. This measure represents the distance between the patch P_i and its closest one in O_j. $d(P_i, O_j)$ is computed using equation 4.11.

It is required to normalize this distance in order to obtain a distribution on $[0,1]$ which allows us a correct construction of the mass function. In equation 4.14, α_i is a confidence coefficient on $[0,1]$. This coefficient is interpreted as the significance given to each patch in a global 3D object. Here, we use the same significance defined in Section 4.3.1. Then all patches P_i of the query Q are modeled by their corresponding mass functions, we can apply the Dempster's combination rule in order to get a mass function which measures the amount of belief committed to the assumption: "Q is similar to O_j". Lastly, a decision is made from the obtained mass function and resulting objects are sorted based on a pignistic probability distribution.

This approach is summarized in algorithm 2.

4.3.3 Experiments and Results

In this section, we present the experimental results of the proposed 3D-retrieval methods. First, we present the databases on which conducted the experiments. Then we present some state-of-the-art shape-matching algorithms to compare with, the evaluation criteria and the

Algorithm 2: : 3D-shape search algorithm (2nd method)

Given a 3D-object query Q.

1: *Extract a set of N feature points V_i from Q.*
2: **for** *each feature point V_i where $i = 1, ..., N$* **do**
3: *Extract a set of closed curves (ie. patch P_i).*
4: *Compute the distance between P_i and all objects in Ω using eq 4.11.*
5: *Set a mass value for all the objects $m_{P_i}(O_j)$ using eq 4.14.*
6: **end for**
7: *Combine all masses m_{P_i} into a new mass function m_Q*
8: *Compute the pignistic probability induced by m_Q*
9: *Display 3D-objects in the order according to the pignistic probability.*

experimental results. Data with noise are considered in the sequel and a quantitative analysis of the experimental results are reported. The algorithms that have been described in the previous sections have been implemented using MATLAB software. The framework encloses an off-line feature point extraction algorithm and a curve extraction algorithm, and an on-line retrieval process. To evaluate our method, we used two different databases. The TOSCA [BBK07] data set for non-rigid shapes and the SHREC07 [MPB07] shape benchmark for partially similar models.

Please note that both presented approaches have been tested with the two proposed descriptors (earlier presented in chapter 3). In the following we will call:

- Belief1_Curve, the first retrieval method using curve descriptor.
- Belief1_Gcord, the first retrieval method using Geodesic cord descriptor.
- Belief2_Curve, the second retrieval method using curve descriptor.

- Belief2_Gcord, the second retrieval method using Geodesic cord descriptor.

Data set description

- The TOSCA[1] dataset: This data set has been proposed by Bronstein et al. [BBK07]. It is an interesting database for non-rigid shape correspondence measures. The database consists of 148 models, enclosing 12 classes. Each class contains one 3D-shape under a variety of poses (between 1 and 20 poses for each class). This classification is provided with the dataset.

- SHREC07 dataset: This benchmark[2] is composed of a dataset of 400 manifold models (see Figure 4.2) and of a query set of 30 manifold models composed of composite models as shown in Figure 4.3. Hence, it is an interesting database for partial shape retrieval. The dataset exhibits diverse variations, from pose change to shape variability within a same class or topology variations (notice 4 of the 20 classes contain non zero genus surfaces) [MPB07]. The ground-truth is provided with the dataset.

Some state-of-the-art algorithms

In order to evaluate the 3D-global matching approach, we compare our method with some state-of-the-art shape-matching algorithms.

- **Extended Reeb Graphs** (ERG): it is a structural based 3D matching method. It works with Reeb graph properties [MPB07].

- **Ray-based approach with Spherical Harmonic representation** (RSH): Saupe and Vranic [DD01] use PCA to align the models into

[1] http://tosca.cs.technion.ac.il/
[2] http://partial.ge.imati.cnr.it/

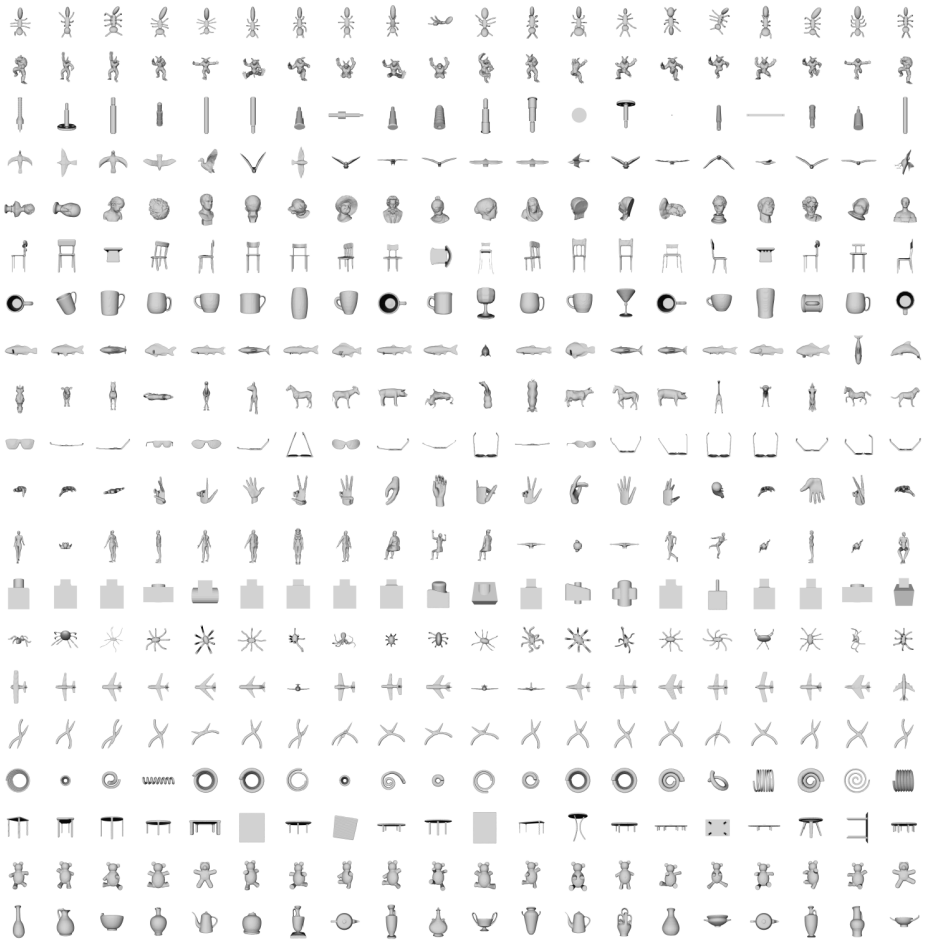


Figure 4.2 – *SHREC07 data-set snapshot.*

Figure 4.3 – *SHREC07 query-set snapshot.*

the canonical position. Then the maximal extents are extracted, and finally the spherical harmonic is applied.

- **The hybrid feature vector** (DSR): it is a combination of two view based descriptors: the depth buffer and the silhouette and radialized extent function descriptor [Vra04].
- **The Geodesic D2**: it is an extension of the Euclidean D2 [OFCD02]. It is computed as a global distribution of geodesic distances in 3D-shapes.

Evaluation criterion

There are several different performance measures which can evaluate retrieval methods. In this section, we test the robustness of our approach using the Precision vs Recall plots. They are well known in the literature of content-based retrieval. The precision and recall are defined as follow:

$$Precision = \frac{N}{A}, Recall = \frac{N}{R}$$

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.

Results

During the off-line process each model in the database has been remeshed to get a regular model. Feature points have been extracted and their related sets of curves (parts) have been extracted and stored into indexed files. During the on-line process, feature points and related curves of the query object are extracted.

Results on the TOSCA dataset In order to show the main contribution of our approaches, some results are shown as a matrix in Figure 4.4.

In this visualization of the matrix, the lightness of each element $(i; j)$ is proportional to the magnitude of the distances between 3D-objects i and j. That is, each square, in this matrix, represents the distances between two 3D-objects. Darker elements represent better matches, while lighter elements indicate worse matches.

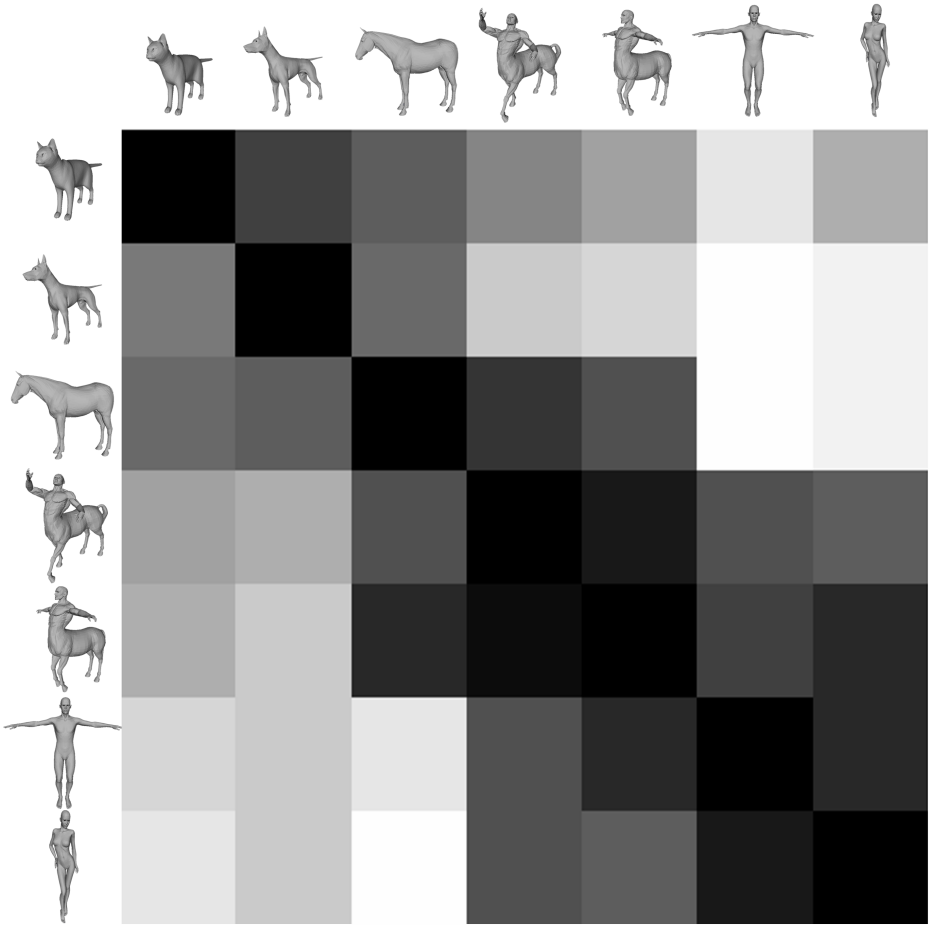


Figure 4.4 – *Matrix of pairwise distances between seven 3D-objects. Using the Belief1_Curve method.*

This matrix is not symmetric because our similarity measure based on belief function is not symmetric either. In this matrix, humans are similar together and partially similar to centaurs. Centaurs are similar together and partially similar to horses and humans. According to the lightness of the matrix square, we can easily do the distinction between

three different classes. The first class contains the first three animals, the second class encloses centaurs and horse and the third class contains humans and centaurs. Other visual results are shown in Figure 4.5. The first column contains the query while the rest are the four retrieved 3D-objects. An interesting result is shown in the last row of Figure 4.5, where the query is a centaur, the forth retrieved 3D-object is a horse which is partially similar to the centaur.

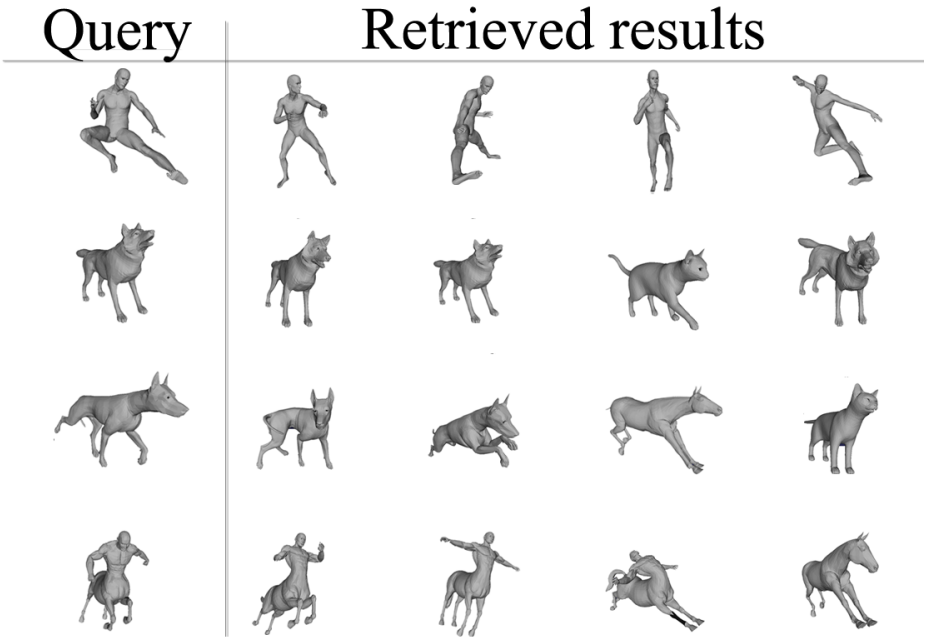


Figure 4.5 – *First retrieved results on the TOSCA dataset. Using the Belief1_Curve method.*

Quantitative analysis of our experimental results are depicted in Figure 4.6. Figure 4.6 shows the Precision vs Recall plots for our approaches and some well-known descriptors. We find that all approaches that we proposed provide the best retrieval precision in this experiment. For instance, using the Belief1_Curve method we obtain the best results, while although Belief2_Gcord performs worst, it outperforms others state-of-the art methods. One can notice here that the curve based descriptor gives more efficient results compared to the geodesic cord descriptor in

both proposed matching methods. Analyzing Figure 4.6, we can also notice that the first proposed matching method outperforms the second one.

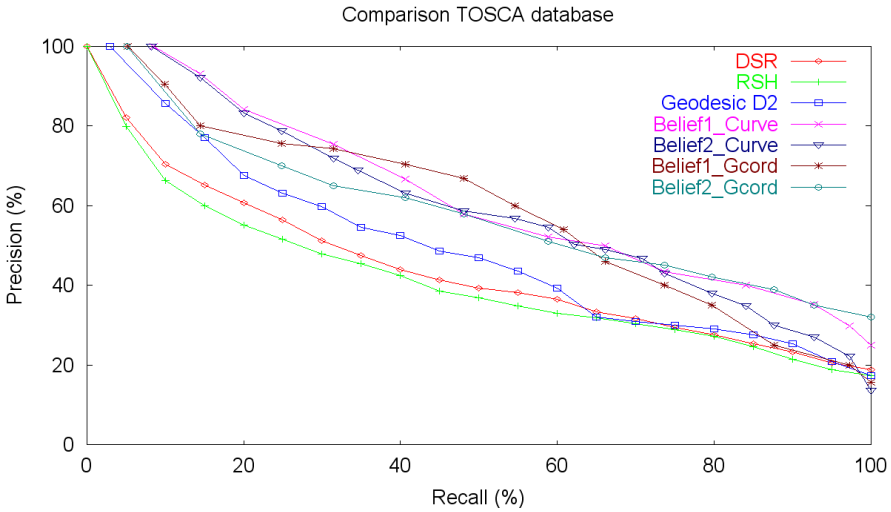


Figure 4.6 – *Precision vs Recall plots comparing the proposed belief based methods to RSH, D2 and DSR algorithms on the TOSCA dataset.*

Results on the SHREC07 dataset First, from a qualitative point of view, Figure 4.7 gives a good overview of the efficiency of the framework. For example, in the first row, the query is a centaur and thus most of the top-results are humanoid and animal models.

From a more quantitative point of view, we compare the Precision vs Recall plot of our approaches with those of the methods competing the contest. Such a plot is the average of the 30 Precision vs Recall plots corresponding to the 30 models of the query-set. Figure 4.8 shows these plots . Here, we notice the same behavior of all methods as in shown in Figure 4.6. Using the Belief1_Curve method we obtain the best results.

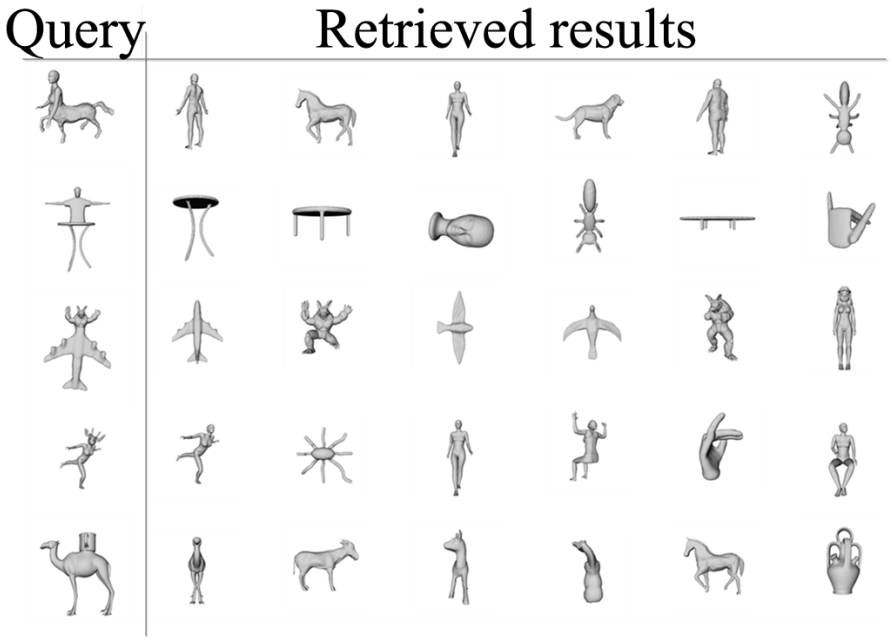


Figure 4.7 – *First retrieved results on the SHREC07 dataset. Using the Belief1_Curve method.*

Robustness to query noises

We investigate the framework robustness against surface noise. For each element of the query-set, we added a surface noise. The noise is $\pm 0.2\%$ and $\pm 0.5\%$ the length of the model's bounding box. It consists of three random translations in the three directions X, Y and Z. Figure 4.9 shows some examples of the noise effect.

Precision vs Recall plot are computed for two proposed approaches (the belief1_curve method and the belief1_Gcord method) under the noise and compared with the original Precision vs Recall plot. From this results shown in Figure 4.10, the noise addition effects can be observed. The performance of the belief1_curve method is robust to $\pm 0.2\%$ of noise. However in the case of $\pm 0.5\%$ of noise, a remarkable descent of the Precision vs Recall plot relative to the original plot can be distinguished. Therefore, we consider the $\pm 0.5\%$ of noise as the limit of the

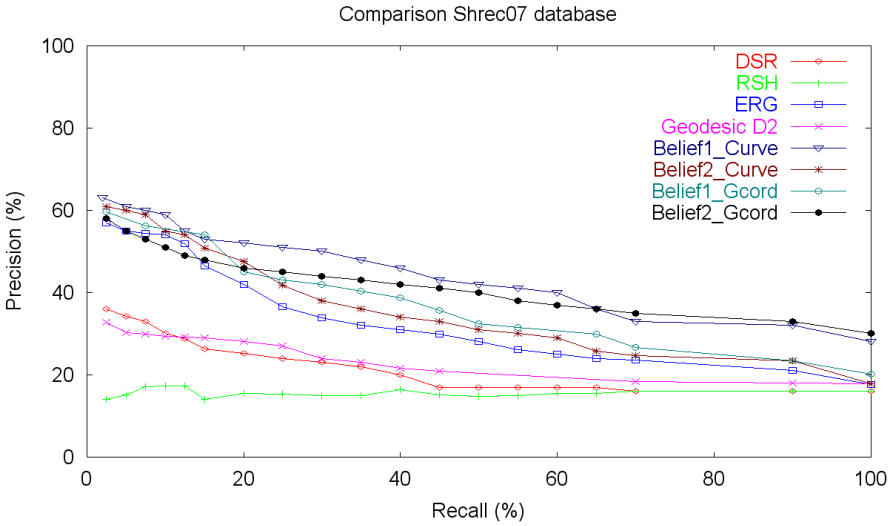


Figure 4.8 – *Precision vs Recall plot on the SHREC07 dataset.*

Figure 4.9 – *Surface noise on a TOSCA 3D-model: Original,* ±0.2% *noise and* ±0.5% *noise (from left to right).*

belief1_curve method in terms of noise robustness. Analyzing results shown in Figure 4.11, we find that the belief1_Gcord method is more robust under the noise addition than the belief1_curve method. This is due to the shape descriptor used in each method. Shapes of curve can highly change with noise addition which affects the robustness of the method.

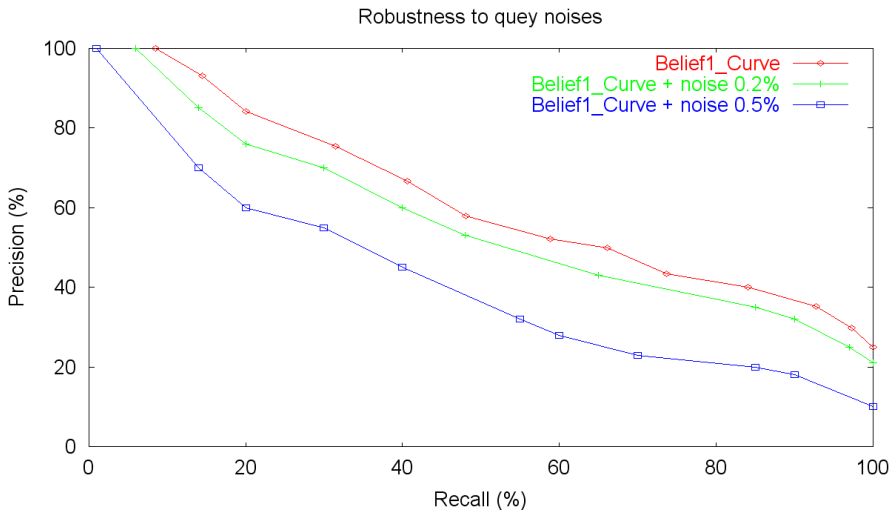


Figure 4.10 – *The Precision vs Recall plot of the belief1_curve method compared with the Precision vs Recall plot with noise addition on the TOSCA dataset.*

Robustness to feature points localization

In order to assess the robustness of our method under localization of feature points, we apply the following transformation for the query set. We select a random vertex located in a geodesic radius (1% and 2% of the geodesic diameter of the 3D-object) centered in the original feature points.

In Figure 4.12 an example of the feature points localization with noise addition is shown. The original feature point is colored in green. The perturbation consists of a random selection of one vertex among vertices colored in red. Figure 4.13 shows that the performance of the belief1_curve is robust with respect to 1% of perturbation. However 2% perturbation plot shows the limit of the method in term of feature point perturbation.

Figure 4.13 shows that the performance of the belief1_Gcord is robust with respect to 1% and 2% of perturbation. Here also, we can conclude that the belief1_Gcord method is more robust than the be-

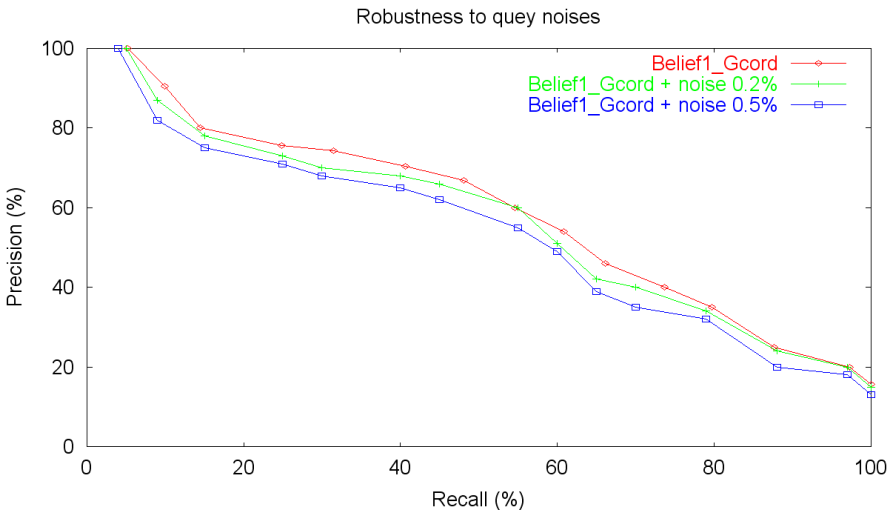


Figure 4.11 – *The Precision vs Recall plot of the belief1_Gcord method compared with the Precision vs Recall plot with noise addition on the TOSCA dataset.*

lief1_curve method with feature point perturbation and this is also due to the shape descriptor used in each method.

Conclusion

In this chapter, we have presented our framework for shape similarity computing based on the evidence theory. In the first section, we gave a theoretical overview of belief function theory. In the second section, we have presented two 3D-shape matching methods for global 3D search and retrieval applications. The belief functions has been used in this matching framework to compute a global similarity metric based on the 3D-patch matching. The results on the TOSCA and the SHREC07 datasets show the effectiveness of our methods. They also show the robustness to rigid and non rigid transformations, surface noise and partially similar models.

In the next chapter, we will present a 3D-shape classification method based on the belief function theory.

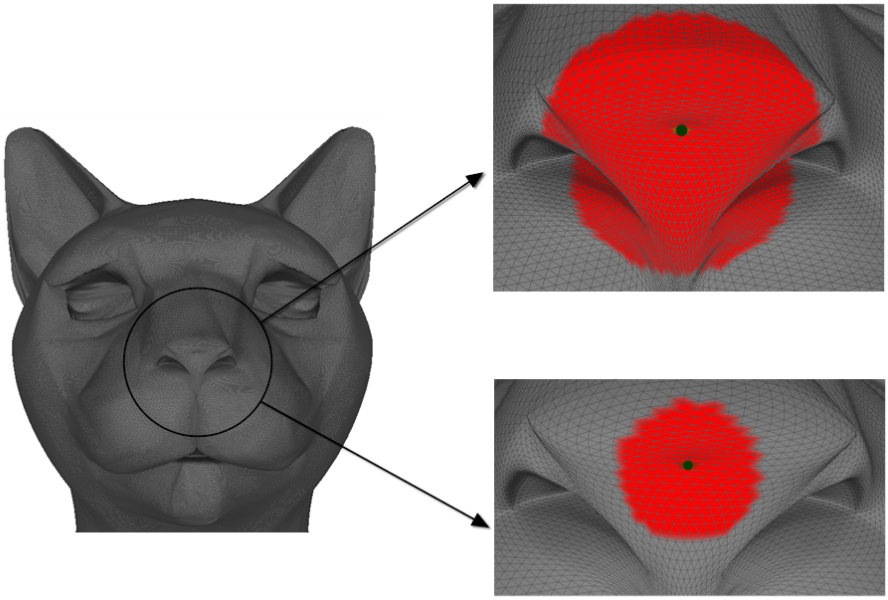


Figure 4.12 – *An example of the feature point location with perturbation (1% bottom and 2% top).*

French conclusion

Dans ce chapitre, nous avons présenté une nouvelle méthode pour le calcul de similarité entre les formes d'objets 3D. Etant donné un ensemble de patches extraits d'un objet 3D, nous considérons que chaque patch peut fournir une certaine information sur la forme globale de l'objet. L'ensemble d'informations fournis par les différents patch de l'objet sont représentées et combinées dans un cadre crédibiliste. Une décision finale sur la forme de l'objet est déduite à partir des informations combinées.

Les résultats obtenus sur deux différentes bases d'objets 3D (TOSCA et SHREC07) montrent l'efficacité de nos méthodes. Ils montrent également la robustesse de notre méthode aux différentes transformations qu'un objet 3D peut subir. Dans le prochain chapitre, nous présenterons

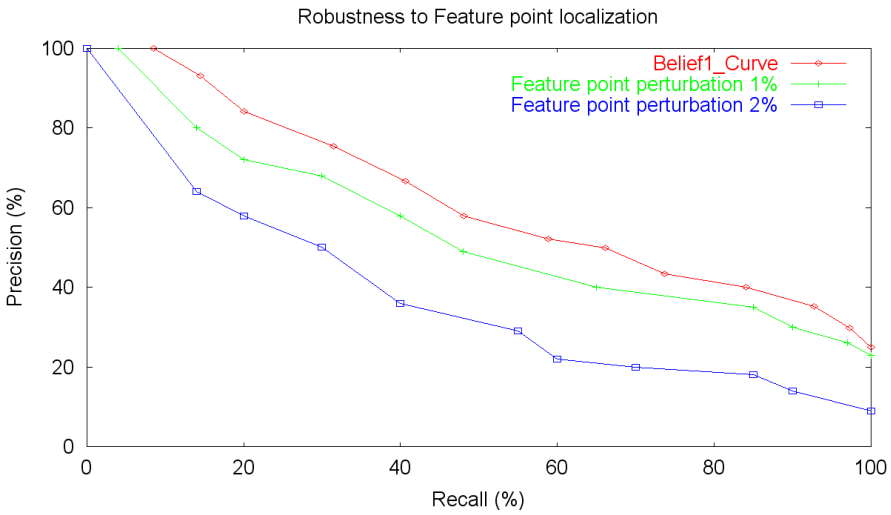


Figure 4.13 – *The Precision vs Recall plot of the belief1_curve method compared with the Precision vs Recall plot with feature points location perturbation on the TOSCA dataset.*

une méthode de classification 3D basé sur la théorie des fonctions de croyance.

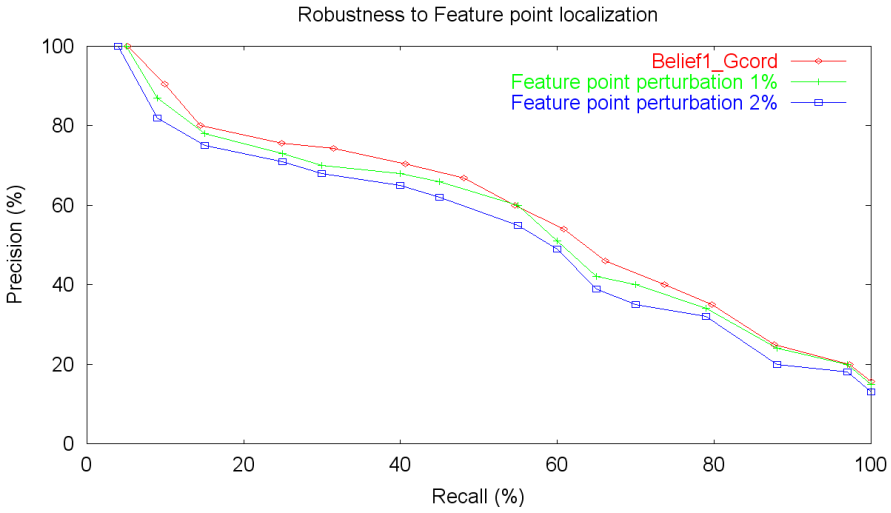


Figure 4.14 – *The Precision vs Recall plot of the belief1_Gcord method compared with the Precision vs Recall plot with feature points location perturbation on the TOSCA dataset.*

5 Belief function based system for 3D-object classification

Contents

This chapter presents a belief based method for 3D-object classification. The method consists of two stages. The training stage, where

3D-objects in the same category are processed and a set of representative parts is constructed, and the labeling stage, where unknown objects are categorized.

5.1 Introduction

This work addresses the categorization problem with a patch-based approach. It consists of capturing a compact model of an object category by building a set of representative patches from the different objects in that category. For this end, patches from the objects in the same category are extracted. We emphasize here on the fact that our patches are the local features of the objects that can be represented by the proposed descriptors (Curves and Geodesic cords). Once we extracted patches from all the objects in the same category, we construct a set containing representative patches. This set is used to represent all object patches in that category. A straightforward way to build this set is to use vector quantization techniques. Here, we use a variation of the evidential k-nearest neighbors algorithm [ZD98]. The centroids of the resulting clusters serve as representative patches for our category. This process is iterated for all the categories in the training set. The labeling of unknown 3D-objects is achieved by labeling their associated patches. Here, we assume that each patch can help to predict the category of the whole object. More specifically, each patch of the object to be labeled is considered as an item of evidence supporting certain hypotheses concerning the category membership of that object. Based on this evidence, the object patches are compared with category representative patches and basic belief masses are assigned to each category. As a result of considering each object patch in turn, we obtain a set of *Basic Belief Assignments* (BBAs) that can be combined using the Dempster's rule of combination to form a resulting BBA synthesizing a final belief regarding the category of the whole object. Another issue is detailed in this section, when labeling an unknown 3D-object, one can be faced with the problem of handling the unclassifiable object (reject). Here we show that we are able to handle this issue using belief theory based data association

method. Using a specific modeling of belief functions, this is done by detecting and managing a portion of a conflict, which originates from the non-exhaustivity of the frame of discernment.

French introduction

Nous consacrons ce chapitre pour la catégorisation d'objets 3D. La catégorisation est divisée en deux phases.

La première phase consiste à construire un modèle compact d'une catégorie d'objet 3D par la construction d'un ensemble de patches représentatifs. Ces patches représentatifs servent à décrire les objets 3D appartenant à la même catégorie.

La deuxième phase consiste à étiqueter des objets 3D inconnus. L'étiquetage est réalisé en fonction de la relation entre les patches représentatifs précédemment construits et les patches extraits aux objets inconnus. Lors de l'étiquetage d'un objet 3D, on peut être confronté au problème de la classification des objets inclassables (le cas où l'objet à étiqueté n'appartient à aucune catégorie de la base d'apprentissage). Nous montrons que nous sommes en mesure de résoudre ce problème en utilisant la théorie des fonctions croyance.

5.2 Representative patch construction

Here, we attempt to find representative patches of each category. For this end, we developed an evidential clustering method based on the Transferable Belief Model concept. Let us consider a category C in the training set. All objects in C are processed (patches are extracted from each object) and the sets of patches are extracted. Let $P^C = \{P_1, ..., P_N\}$ be the collection of all object patches in C. We assume that these patches can be classified into M classes $W = \{W_1, ..., W_M\}$ (for the choice of M

see Algorithm 3) and each patch P_i will be assumed to possess a class label indicating with certainty its membership to one class in W. Let P_s be an incoming patch to be classified. Classifying P_s means assigning it to one class in W. Using the vocabulary of the evidential theory, W can be called the *frame of discernment* of the problem.

Let us denote by Θ_s the set of the k-nearest neighbors of P_s in P^C, according to some distance measure D (here, D represents the distance between two descriptors presented in chapter 3). Let $P_k \in \Theta_s$ a 3D-patch classed in W_l. The pair (P_k, W_l) can be regarded as a piece of evidence that increases our belief that P_s also belongs to W_l. However, this piece of evidence does not provide by itself 100% certainty. In the evidential formalism, this can be expressed by saying that only some patches of our belief are committed to W_l. Since the fact that $P_k \in W_l$ does not point to any other particular hypothesis, the rest of our belief can not be distributed to anything else than W, the whole frame of discernment. This item of evidence can therefore be represented by a *Basic Belief Assignment* (BBA) m_k verifying:

$$\begin{aligned} m_k(W_l) &= \chi e^{(-\gamma_l D^2(P_s,P_k))} \\ m_k(W) &= 1 - m_k(W_l) \\ m_k(A) &= 0 \ \forall A \in 2^W \setminus \{W, \{W_l\}\} \end{aligned} \tag{5.1}$$

With χ a parameter such that $0 \leq \chi \leq 1$. γ_l is obtained by an optimization procedure proposed by Zouhal and Denoeux [ZD98]. We set $\chi = 0.9$. For each of the k-nearest neighbors of P_s, a BBA depending on both its class label and its distance to P_s can therefore be defined. In order to make a decision regarding the class assignment of P_s, these BBAs can be combined using Dempster's rule [Sha76] into one BBA m_s. As a result, P_s will take the label of the class maximizing the pignistic probability induced by m_s. Algorithm 3 summarizes this method.

Algorithm 3: 3D-patch clustering algorithm

Input:

Each patch in P^C is considered as one class. W is initialized to be $W = \{W_1..., W_N\}$ and N is the number of patches in P^C, so that $M = N$

Begin

REPEAT

1. *Set a random order to patches in P^C;*

FORALL *P_i in P^C with respect to the order*

2. *Compute m_k (for each of the k-nearest neighbors of P_i);*

3. *Compute m_i by combining m_k using Dempster's rule of combinaison;*

4. *Compute BetP the pignistic probability induced by m_i;*

5. *Change the label of P_i in W according to BetP;*

ENDFOR

6. *Analyze W and reduce the number of classes;*

UNTIL *obtaining a stable partition in W;*

End

Once the clustering process is achieved, we compute the centroid of each cluster. The centroid is a patch whose parameter values are the mean of the parameter values of all the patches in the cluster. Centroids here are called *representative patches* and denoted by R.

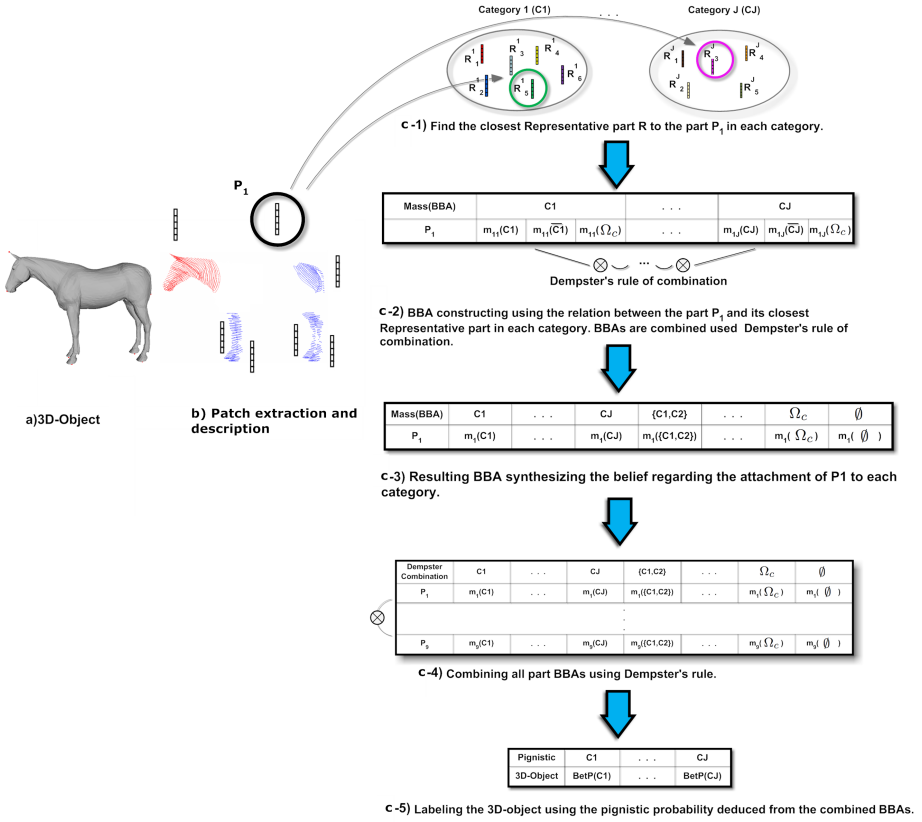


Figure 5.1 – *The labeling system architecture: the input of the system is a 3D-object to be labeled. The output is the probabilities of belonging to each category .*

5.3 Labeling a new 3D-object

In this section, we focus on the labeling of 3D objects. Figure 5.1 shows the various steps of this process. First, giving an object O to be labeled (Figure 5.1 step a), the algorithm begins by partitioning this object (Figure 5.1 step b). Second, an invariant descriptor is associated to each extracted patch of that object (Figure 5.1 step b). The partitioning and the description of these patches are done in the same way as in the training process. The labeling of the object O is achieved based on its patches. Here, we assume that each patch can help to predict the category of the whole object. In the context of belief functions, we can say that each patch represents an evidence source which provides an information regarding the category of the object. By considering all patches, we obtain a set of evidence sources that can be combined to produce a final decision concerning the category of the object.

5.3.1 Evidence extracting from patches

Recalling here that, after the training process, each category contains a set of representative patches. Given a patch extracted from the object to be labeled (for example P_1 in Figure 5.1 step b), we select in each category the representative patch that is the closest to P_1 (Figure 5.1 step c-1). Then, from the selected representative patch a mass function that quantifies the degree of belief given to the assumption "P_1 matches with a particular category C_j" is derived (Figure 5.1 step c-2). As a result of considering each category in turn we obtain a set of BBAs that can be combined using the Dempster's rule of combination to form a resulting BBA. This BBA syntheses a final belief regarding the relation between the patch and the categories (Figure 5.1 step c-3).

More formally, let us denote by $\{P_i\}_{1\leq i\leq I}$ the set of I patches composing the object O. From the training set, we enumerate J categories

$\Omega_c = \{C_j\}_{1 \leq j \leq J}$. A category C_j contains a set of representative patches. Let $R^j_{P_i}$ be the closest representative patch to the patch P_i in the category C_j. Each pair $(P_i, R^j_{P_i})$ with $1 \leq j \leq J$ constitutes a distinct item of evidence regarding the category membership of P_i. If P_i is closed to $R^j_{P_i}$ then one will be inclined to believe that both patches belong to the same category. On the contrary if their dissimilarity is very large then we consider that P_i may belongs to $\overline{C_j}$ the complement of C_j in Ω_c. Consequently this item of evidence may be postulated to induce a basic belief assignment BBA m_{ij} over Ω_c defined by:

$$\begin{aligned} m_{ij}(C_j) &= \mu \cdot S(P_i, C_j) \\ m_{ij}(\overline{C_j}) &= \mu \cdot \big(1 - S(P_i, C_j)\big) \\ m_{ij}(\Omega_c) &= 1 - \mu \end{aligned} \tag{5.2}$$

$S(P_i, C_j) = e^{(-D(P_i, R^j_{P_i}))}$ is a function of the distance between the patch P_i and its closest representative patch $R^j_{P_i}$ in the category C_j. μ is a discounting coefficient [Sha76] associated with the category C_j. In practice, we set $\mu = 0.9$ for all categories.

As a result of considering each category we obtain J BBAs as shown in Figure 5.1 step d-2. These masses are combined using the Dempster's rule of combination to form a resulting BBA m_i synthesizing a final belief regarding the attachment of P_i to each category. Figure 5.1 step d-3 shows the resulting BBA m_i.

5.3.2 3D-object labeling

In order to get a final decision about the category of the unknown 3D-object, all masses m_i $i \in [1..I]$ are combined using the Dempster's rule of combination (Figure 5.1 step c-4).

A decision can be made regarding the category membership of the 3D-object by examining the pignistic probability deduced from the re-

sulting mass m (Figure 5.1 step c-5). The labeling process is summarized in Algorithm 4.

Algorithm 4: : Labeling a new 3D-object

Input: *Given a training set of J categories* $\Omega_c = \{C_j\}_{1 \leq j \leq J}$. *Each category is represented by a set of representative patches.. Given an 3D-object O to be labeled.*

Begin

1. Partition O into I 3D-patches $\{P_i\}_{1 \leq i \leq I}$;

FORALL P_i in $\{P_i\}_{1 \leq i \leq I}$

FORALL C_j in Ω_c

2. Find the closest representative patch $R^j_{P_i}$ *to the patch* P_i;

3. Compute m_{ij} *(according to eq.5.2);*

ENDFOR

4. Compute m_i *by combining* m_{ij} *using Dempster's rule;*

ENDFOR

5. Compute m by combining all m_i *using Dempster's rule;*

6. Compute the pignistic probability induced by m for all categories;

7. Label O according to the highest pignistic probability;

End

5.4 Reject option

Introducing a reject option is very useful, yet a difficult problem in data classification. Instead of Bayesian classifiers where the reject is modeled empirically by comparing the posteriori probability with a threshold (T) [VFJZ01], the reject option in the belief theory is modeled in natural way. It can be deducted from the conflict on each mass distribution in Figure 5.1 step c-3. The idea consists to divide the conflict into two components: a conflict due to the non-exhaustivity of the frame of discernment repre-

C_1	$\emptyset$	C_1	C_1
$\overline{C_1}$	C_2	$\emptyset$	C_2
Ω_c	C_2	C_1	Ω_c
	C_2	$\overline{C_2}$	Ω_c

Table 5.1 – *Representation of the Dempster's rule of combination*

sented by the reject and an unknown conflict. In order to illustrate this idea, let us consider a two-element frame of discernment $\Omega_c = \{C_1, C_2\}$. Ω_c can also be represented by $\{C_1, \overline{C_1}\}$ or $\{C_2, \overline{C_2}\}$. Given two BBA m_1 and m_2 which are defined on Ω_c by: $m_1 = m_1(C_1), m_1(\overline{C_1}), m_1(\Omega_c)$: a 3D-object belongs or does not belong to the category C_1 or we are in a situation of almost complete ignorance concerning the category of that patch. $m_2 = m_2(C_2), m_2(\overline{C_2}), m_2(\Omega_c)$: a 3D-object belongs or does not belong to the category C_2 or we are in a situation of almost complete ignorance concerning the category of that patch.

Here, m_1 and m_2 are considered as two independent sources of information to be combined in order to decide with which category, the object is associated. The evidence combination of these two belief using Dempster's rule of combination can be represented by Table 5.1. The last row and the first column of this table are named by the subsets of Ω_c. Each of the squares in the table correspond to the intersection of the subset of each source of information m_1 and m_2. The value of BBA taken for the resulting subset is obtained by the multiplication of the BBA values of the subsets constituted Ω_c. From this table, we can see that the conflict (the mass of the empty set) is represented by two grids. Its value is given by:

$$m_{12}(\emptyset) = m_1(C_1).m_2(C_2) + m_1(\overline{C_1}).m_2(\overline{C_2}) \tag{5.3}$$

The first portion of the conflict $m_1(C_1).m_2(C_2)$ is created because the two sources of information m_1 and m_2 related respectively to cate-

gory C_1 and C_2 confirmed that a 3D-object corresponds to the two categories at the same time. In contrast, the second portion of the conflict $m_1(\overline{C_1}).m_2(\overline{C_2})$ is created because the two sources of information confirm that the 3D-object does not correspond to any category and thus the frame of discernment is not exhaustive. Hence, in our view, the separation between the first and the second portion of the conflict must be done because they do not have the same origin. We define a reject when sources of information confirm that the 3D-object does not correspond to any category. The reject is added to the frame of discernment as a new element and its belief degree is given by: $m_{12}(reject) = m_1(\overline{C_1}).m_2(\overline{C_2})$. More specifically, in our case and through multiple sources of information in the Table in Figure 5.1 step (d-2), the mass value of the reject is given by:

$$m_i(Reject) = \prod_{j=1}^{J}(m_{ij}(\overline{C_j})) \tag{5.4}$$

Finally, according to the pignistic probability deduced from the mass function computed in section 5.3.2, a decision can be taken about the reject of the 3D-object.

5.5 Experiments and Results

In this section, we present the experimental results of the proposed 3D-classification methods. We present results from three experiments. In the first expirement, we evaluate the performance of the belief based classifier and explore the impact of the choice of descriptors on classifier accuracy. The accuracy is computed as the percentage of object models which are correctly classified. We then compare the performance of the belief based classifier with the Bayesian classifier [HKDH04] on the same problem. In the second experiment we describe results with com-

parison to some state-of-the-art retrieval methods. The last experiment shows the contribution of reject modeling option of our framework.

The experiments were conducted on two different datasets. The first dataset is the SHREC07 database. As mentioned in chapter 4, the database contains 400 3D-objects classified into 20 classes. It is a challenging dataset, not only because of the large number of classes, but also because it contains shapes with highly variable poses and non-rigid or isometric transformations. Figure 5.2 reviews some examples from this dataset. Each object in the figure represents one class. The second dataset is composed of shapes from the TOSCA and the Sumner datasets. The TOSCA dataset has been proposed by Bronstein et al. [BBK07] for non-rigid shape correspondence measures. The Sumner dataset has been proposed by Sumner and Popovic [SP04] for deformable shape correspondence. The total set size is 380 shapes. Figure 5.3 shows some examples from this dataset. Each object in the figure represents one class.

Figure 5.2 – *SHREC07 dataset snapshot. Each object corresponds to one category.*

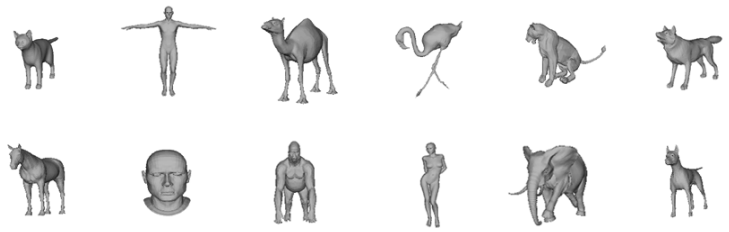


Figure 5.3 – *TOSCA-Sumner dataset snapshot. Each object corresponds to one category.*

5.5.1 Framework performance

From a qualitative point of view, Figures 5.4 and 5.5 give a good overview of the efficiency of the framework on the SHREC07 dataset. Figure 5.4 presents a confusion matrix. Rows in this matrix correspond to query parts extracted from a human 3D-object, and columns correspond to the different categories shown in Figure 5.2 (Ordered from left to right and from top to bottom). The lightness of each element $(i; j)$ is proportional to the magnitude of the similarity between the part i and its closest representative one in the category j. Lighter elements represent better matches, while hot elements indicate worse matches. One can notice in this visualization that the parts of the human object tends to match with the 12th object category which corresponds to the human one in Figure 5.2. This result confirms our assumption that 3D-objects in the same category have the same parts.

Figure 5.5 shows another confusion matrix. In this matrix rows correspond to 3D-object queries and columns correspond to the categories shown in Figure 5.2. The lightness of the diagonal squares of the matrix proves the effectiveness of our classifier.

More quantitatively, Table 5.2 and Table 5.3 show the classification results of our framework using different descriptors. On the SHREC07 dataset (Table 5.2), Geodesic cords descriptors yielded an accuracy of

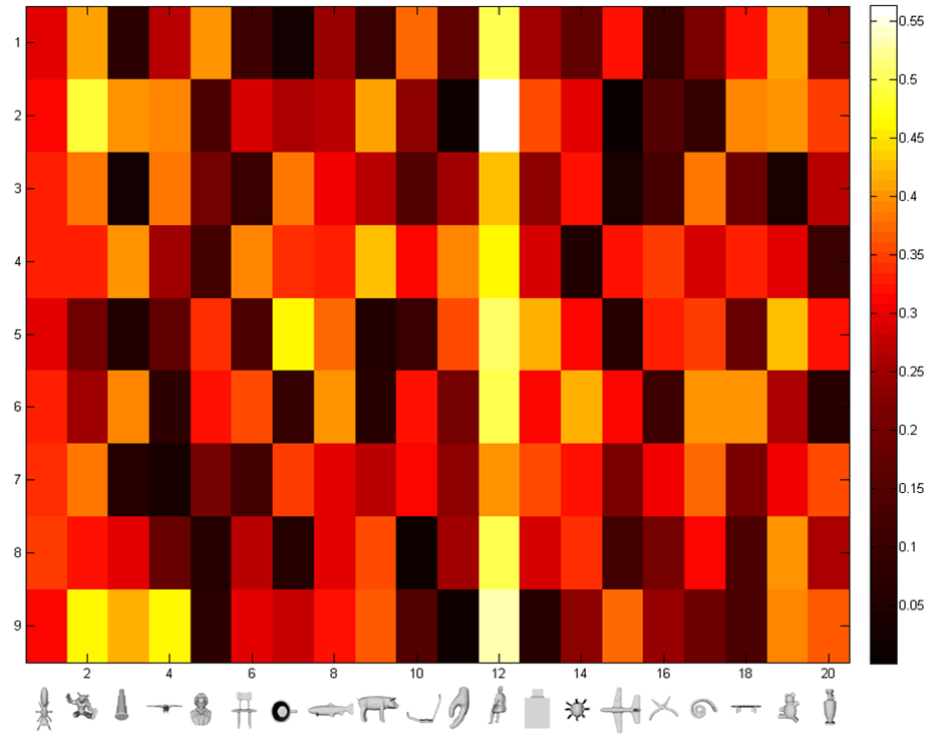


Figure 5.4 – *The confusion matrix for a human object part categorization. Rows are query parts. Columns are object categories. (Using the Curve feature descriptor on the SHREC07 dataset)*

around 75%, while the Curve descriptor leads to a much higher accuracy of around 87.35%. These results show that the Curve descriptor is more suited for local shape description. A combination of the Curve descriptor and Geodesic cord feature yields a better accuracy than Curve descriptor feature alone 89.3%(the combination is based on the mean distance). On the Tosca-Sumner dataset (Table 5.3), the combination of the Curve descriptor and Geodesic cord feature also gives the highest accuracy rate 95%.

5.5.2 Belief classifier versus Bayesian classifier

Table 5.4 and Table 5.5 show a comparison accuracy between the belief classifier and the Bayesian classifier on respectively the SHREC07

Test Data	Dataset Size	Curve	Geodesic cord	Curve-Geodesic cord
Training Set	100	90	76.8	92.4
Test Set	300	84.7	74.5	86.2
Entire Dataset	400	87.35	75.65	89.3

Table 5.2 – *Classification accuracies with feature descriptor changing on the SHREC07 dataset.*

Test Data	Dataset Size	Curve	Geodesic cords	Curve-Geodesic cords
Training Set	90	94.2	84.2	95.1
Test Set	290	92.3	83.8	94.9
Entire Dataset	380	93.25	84	95

Table 5.3 – *Classification accuracies with feature descriptor changing on the Tosca-Sumner dataset.*

Test Data	Dataset Size	Our methods Accuracy (%)	Bayesian classifier Accuracy (%)
Training Set	100	92.4	65.3
Test Set	300	86.2	62.3
Entire Dataset	400	89.3	63.8

Table 5.4 – *Classification result comparison with the Bayesian classifier on the SHREC07. (Using the Curve descriptor-Geodesic cord feature descriptor).*

Test Data	Dataset Size	Our methods Accuracy (%)	Bayesian classifier Accuracy (%)
Training Set	90	95.1	75.6
Test Set	290	94.9	72.3
Entire Dataset	380	95	73.95

Table 5.5 – *Classification result comparison with the Bayesian classifier on the Tosca-Sumner. (Using the Curve descriptor-Geodesic cord feature descriptor)*

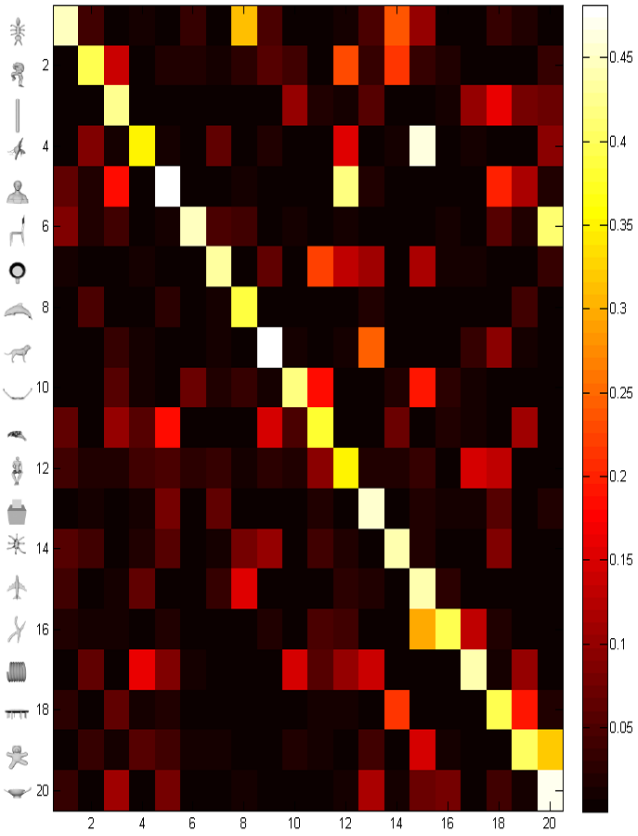


Figure 5.5 – *The confusion matrix for some 3D-object categorization. Rows are query object. Columns are object categories. (Using the Curve feature descriptor on the SHREC07 dataset)*

and the Tosca-Sumner datasets. On the SHREC07 dataset, our classifier showed an accuracy of 92.4% and 86.2% on the training set and an independent test set, respectively. That is to say 89.3% accuracy over the entire dataset. Using a Bayesian classifier we report only 63.8% accuracy. On the Tosca-Sumner dataset, results confirm the contribution of the use of the belief framework instead of the Bayesian one. The belief classifier reports 95% while the Bayesian one reports only 73.95%.

5.5.3 Comparison with related work

In this experiment, we compare our method effectiveness to methods proposed by Biasotti et al. [BGM*06].

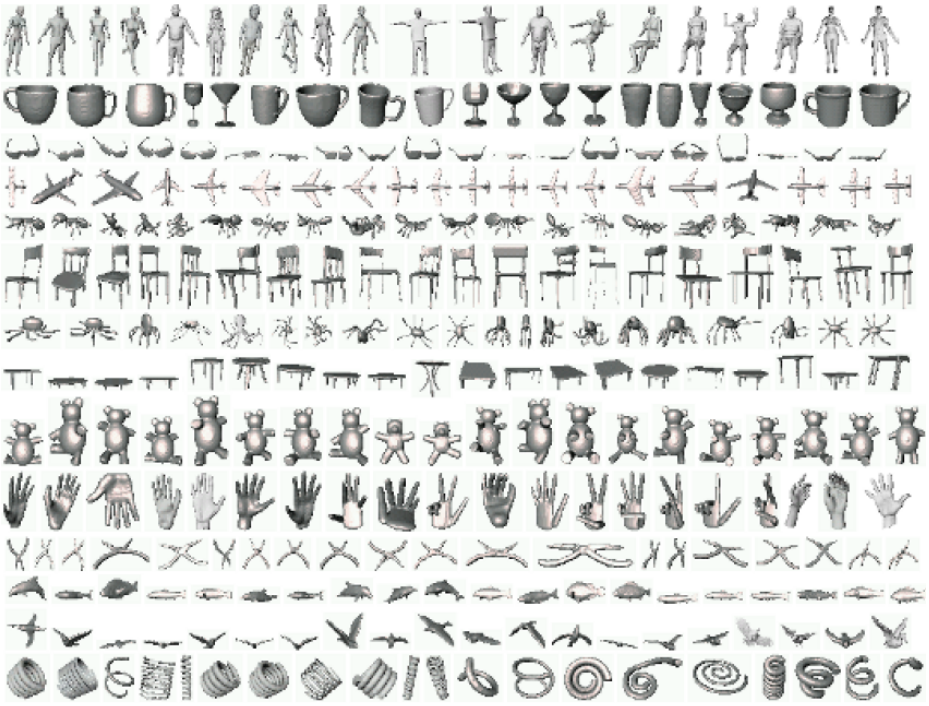


Figure 5.6 – *The subset of the SHREC07 dataset used in the Biasotti et al. framework.*

In this work, the authors compared the performance of five similarity measures on four different shape descriptors in classifying 3D objects. The four different shape descriptors used in their paper are:

- The spherical harmonics (SH) [KF02] which is a volume-based descriptor.
- The light-field descriptor (LF) [GSCO07] which is an image-based descriptor and two topological matching methods.
- The Multi-resolution Reeb graph (MRG) [MYTT01].
- the Extended Reeb graph (ERG) [BM05].

The five similarity measures are:

- The Minimum Distance Classifier (MinDC) which coincides with the Nearest Neighbor classifier.

- Maximum Distance Classifier (MaxDC) which classifies a query by taking into account the most dissimilar descriptor belonging to the class.
- The Average Distance Classifier (AvgDC) which is defined as the average distances between the query and the members of the class.
- The Centroid Distance Classifier (CDC) where the query is classified according to its dissimilarity with a representative member of a class.
- the Atypicity Distance Classifier (ADC) which evokes the notion of typicity, to represent how much a descriptor is typical of the class it belongs to, with respect to the elements in the other classes.

In order to demonstrate the effectiveness of our method compared with the Biasotti et al's classifiers, we tested our method on the same dataset used by Biasotti et al. This dataset is a subset of the SHREC07 dataset composed of 280 3D-objects classified into 14 classes (see Figure 5.6). The results of the experiment are shown in Table 6.1. Each entry is related to the performance of a given shape descriptor (enumerated in the second row) for a given classifier (reported in the first column of the table). The classification accuracy of our method is given in the last row. While Biasotti et al. concluded that the MinDC (nearest neighbor) similarity measure performed the best for all four different shape descriptors in their work, one can notice that our classifier shows the highest classification rate 93.4%. Moreover, the nearest neighbor based approaches require to compare each object to be classified to all objects in the dataset, which seems to be impractical with huge databases where our method is preferred, while it requires matching of only the representative parts. Please note that using a PC with a 3 Ghz Core 2 Duo processor with 3 GB memory, and a Matlab implementation of our algo-

	State of the art methods (Rate in %)			
	SH	LF	MRG	ERG
MinDC	89%	88 %	88 %	83 %
MaxDC	33%	38 %	41 %	38 %
AvgDC	66%	73 %	74 %	58 %
CDC	64%	68 %	76 %	60 %
ADC	63%	68 %	73 %	58 %
Our method (Rate in %)				
93.4%				

Table 5.6 – *Classification result comparison with related work.*

Classification Algorithm	Training Set Size	Test Set Size	Reject Rate (%)	Accuracy (%)
Our algorithm with reject option	100	400	6.6	92.3
Bayesian algorithm with reject (T=0.1)	100	400	3.1	65.4
Bayesian algorithm with reject (T=0.15)	100	400	7.3	69.2
Our algorithm without reject option	100	400	0	89.3
Bayesian algorithm without reject	100	400	0	63.8

Table 5.7 – *Classification results with Reject option on the SHREC07 dataset.*

rithms, the running time of the labeling process depends on the quality of the meshes and their number of vertices. The full processing time of a query (from the SHREC07 or the Tosca-Sumner datasets) varies from 2 to 25 seconds.

5.5.4 Reject option contribution

Table 5.7 shows the accuracies for 3D-classification with and without reject option on the SHREC07 dataset. One can notice that the classification accuracy improved from 89.3% to 92.3%. For Bayesian classifier, the 3D-object whose maximum a posteriori probability is below the threshold (T) in Table 5.7 are rejected. When T=0.15, we notice that the

Bayesian classifier rejects much more than our classifier. However our accuracy is still higher.

Conclusion

In this chapter, we have presented a patch-based method for categorizing 3D-objects using a new evidential classifier. The categorization process is completely automated and consists of two different stages. The training stage which lies on the category model building is based on the belief function theory and it goes into two steps: 1) 3D-patch extraction and 2) representative parts construction. The second stage is the labeling, in which belief functions have been also used. In the labeling process, we have introduced a reject option, which can be used to handle the labeling of unknown 3D-objects. The classifier has been evaluated on two databases of 400 and 380 3D-models. Our system achieved a classification accuracy over 89.3% and 95%, respectively on the two datasets. The reject option has also been evaluated and the experimental results obtained on the SHREC07 dataset show that this option efficiently improves the classification accuracy from 89.3% to 92.3%.

In the next chapter, we will present another method, for shape similarity computing, which will be applied to 3D-retrieval and classification. The method, we propose, is based on the bag of Feature techniques (BoF). This technique, which is a popular approach in areas of computer vision and pattern recognition, have recently gained great popularity in shape analysis community.

French conclusion

Dans ce chapitre, nous avons présenté une méthode de catégorisation d'objets 3D en utilisant la théorie des fonctions de croyance. Le pro-

cessus de catégorisation est entièrement automatisé et se compose de deux étapes. L'étape d'apprentissage qui consiste à construire un modèle compact d'une catégorie d'objets.

L'étape d'étiquetage, au cours de quelle, nous avons introduit une option de rejet. Le classifier a été évalué sur deux bases de données de 400 et 380 objets respectivement. Notre système atteint une précision de classification de 89,3% et 95%, respectivement sur les deux ensembles d'objets. L'option de rejet a été également évaluée et les résultats obtenus sur la base SHREC07 montrent que cette option améliore la précision de la classification de 89,3% à 92,3%.

Dans le prochain chapitre, nous allons présenter une autre méthode, pour la mesure de similarité entre les formes d'objets 3D, qui sera appliqué à la recherche et la classification. La méthode que nous proposons est basé sur les techniques de *sac de mots*.

6 A bag of feature technique for 3D-object retrieval and classification

Contents

This chapter presents a method for 3D-shape matching using Bag-of-Feature techniques. The method starts by selecting and then

describing a set of points from the 3D-object. Based on vector quantization, we cluster the set of descriptors to form a shape vocabulary. Then, each point selected in the object is associated to a cluster (word) in that vocabulary. The method is applied to 3D-object retrieval and classification.

6.1 Introduction

Bag of Features (BoF), which is a popular approach in areas of computer vision and pattern recognition, have recently gained great popularity in shape analysis community. Liu et al. [LZQ06] 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 quantizing Spin images of the object. Ohbuchi et al. [OOFB08] introduced a view-based method using salient local features. They represented 3D-objects as word histograms derived from the vector quantization of salient local descriptors extracted on the depth-buffer views captured uniformly around the objects. Ovsjanikov et al. [OBBG09] 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. [TCF09] has used the bags of features 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.

First, a theoretical study showing the building of bag of feature is presented. Then, we show its application in the 3D-shape retrieval and classification tasks.

French introduction

Les techniques *sacs de mots* sont des outils très populaires dans les domaines de la vision la reconnaissance des formes. Ces techniques

ont récemment gagné une grande popularité dans la communauté de l'analyse de forme [LZQ06, OOFB08, TCF09].

D'une manière générale, on considère que le monde des formes 3D peut être décrit au moyen d'un dictionnaire (de «mots»). Un objet particulier est alors représenté par l'histogramme des occurrences des mots le composant. Il s'agit donc d'un vecteur de la même taille que le dictionnaire, dont la composante *i* indique le nombre d'occurrences du i-ème mot du dictionnaire dans le document. La constitution du dictionnaire est généralement effectuée par un algorithme de quantification.

Dans ce chapitre nous présentons premièrement, une étude théorique montrant la construction des sacs de mots. Ensuite, nous montrons son application dans la recherche et la classification d'objets 3D.

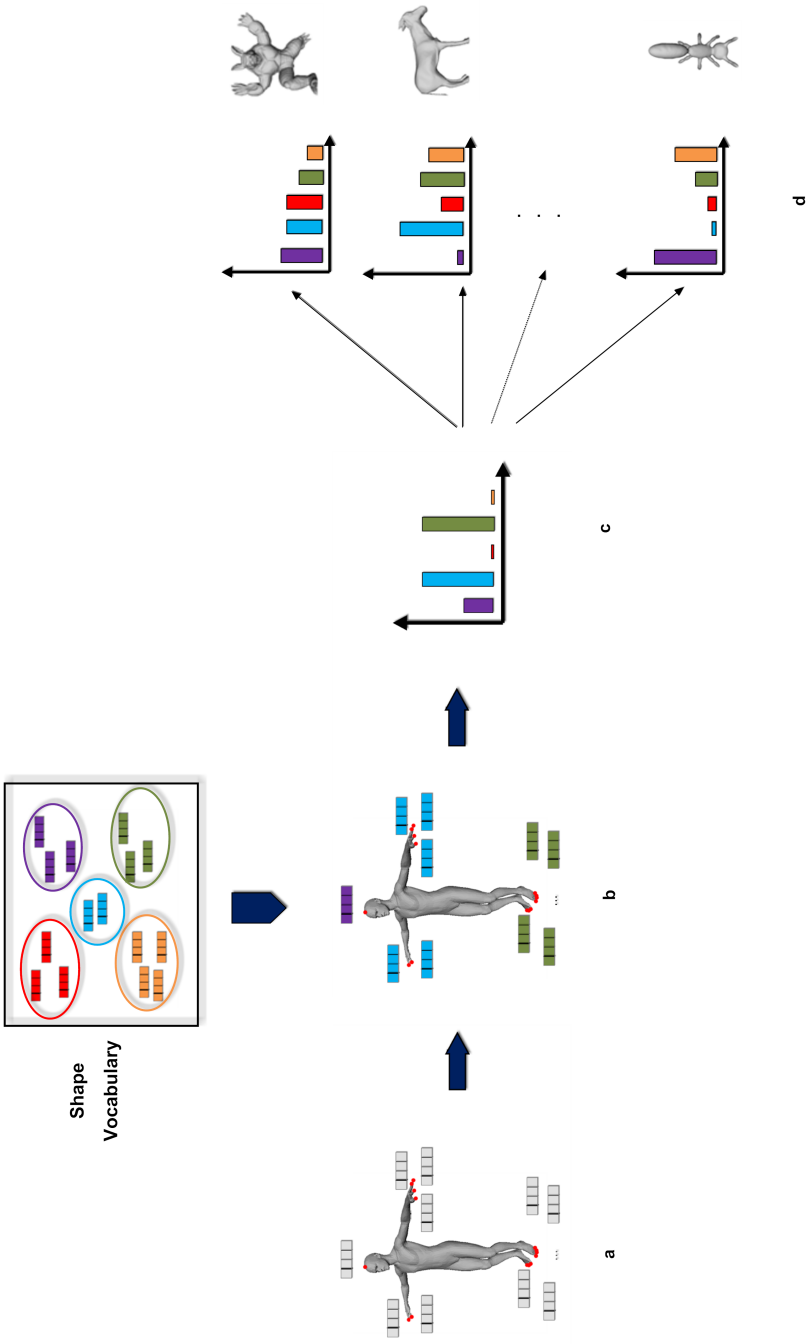


Figure 6.1 – *An illustration of our method. (a) Detection and description of 3D-object patches, (b) Assigning patch descriptors to a set of predetermined clusters (a vocabulary) with a vector quantization algorithm, (c) Constructing a bag of keyshapes and (d) Matching between two objects*

6.2 Bag of Feature techniques

The method of BoF is largely inspired by the Bag of Words concept which has been used in text retrieval and classification for quite some time. Even though there are countless variations of algorithms emerging under the label Bag of Features and it is hard to capture the actual BoF algorithm, there is a common concept which is shared by all of theses methods. We present this concept in the following sections.

6.2.1 Bag of Feature Building

The general steps for building a BoF representation for 3D-objects are depicted in Figure 6.1.

1. Detection and description of 3D-object patches in this approach (Figure 6.1(a)), we use the algorithms developed in chapter 3 for the feature extraction and description process.

2. Assigning patch descriptors to a set of predetermined clusters (a vocabulary) with a vector quantization algorithm (Figure 6.1(b)).

3. Constructing a weighted vector of keyshapes, which counts the number of patches assigned to each cluster (Figure 6.1(c)).

4. Matching between two objects, treating their bag of keyshapes as the feature vector, and thus determine their dissimilarity (Figure 6.1(d))

Ideally these steps are designed to maximize matching accuracy while minimizing computational effort. Thus, the descriptors extracted in the first step should be invariant to variations that are irrelevant to the retrieval task (deformable shapes) but rich enough to carry enough information to be discriminative between dissimilar objects. The vocabulary used in the second step should be large enough to distinguish

relevant changes in 3D-object patches, but not too large as to distinguish irrelevant variations such as noise. We refer to the quantized feature vectors (cluster centers) as "*keyshape*" by analogy with "*keywords*" in text retrieval. However, in our case "words" do not necessarily have a repeatable meaning such as "*legs*", or "*head*", nor is there an obvious best choice of vocabulary. Instead, our goal is to use a vocabulary that allows good matching performance. We now discuss the choices made for each step in more details.

6.2.2 Shape vocabulary construction

The vocabulary is obtained by quantification of the set of descriptors extracted in the training stage. The vocabulary is used to construct discriminant representatives, with which any 3D-object can be described. One extreme of this approach would be to compare each query descriptor to all descriptors in the indexing stage: with the huge number of descriptors involved, this seems impractical. Another extreme would be to try to identify a small number of large "clusters" which sufficiently discriminate between different shape classes. Moreover, the representation space is neither densely nor uniformly populated. Some descriptors may never appear in 3D-objects while others can be very frequent. The first consequence of this remark is that the vocabulary must be adapted to the encountered 3D-objects. That is to say, it must reflect local descriptors present in the 3D-objects. The most common method to build the shape vocabulary is to arrange descriptors encountered in the training stage into a finite number of clusters using a clustering algorithm. The number of clusters is the vocabulary size. For this end, we chose to use the simplest square-error partitioning method: K-means [DHS00]. This algorithm proceeds by iterated assignments of points to their closest cluster centers and re-computation of the cluster centers. Two diffi-

culties 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. For example, Pelleg et al [PM00] use cluster splitting to do it, where the splitting decision is done by computing the Bayesian Information Criterion. However, in the present case we do not really know anything about the density or the compactness of our clusters. Moreover, we are not even interested in a "correct clustering" in the sense of feature distributions, but rather in accurate categorization. We therefore run k-means several times with different number of desired representative vectors (K) and different sets of initial cluster centers. We select the final clustering giving the lowest empirical risk [Vap98].

6.3 Application to 3D-object Retrieval

In this section, we present the use of BoF for the 3D-object retrieval context.

6.3.1 3D-Object indexing using text retrieval methods

In text retrieval methods, each document is represented by a vector of word frequencies. The word frequency in the given document is simply the number of times a given word appears in that document. It is usual to normalize the frequency vector by the sum of occurrences of all terms of the considered document. Moreover, in text retrieval methods, word frequencies are usually weighted. Here we present the standard weighting that is employed, and then the shape analogy of document retrieval to 3D-object retrieval. The standard weighting is known as *term frequency - inverse document frequency*, (*tf-idf*), and is computed as follows. Suppose there is a vocabulary of k words, then each document

is represented by a k-vector $V_d = (t_1, ..., t_i, ..., t_k)^T$

$$t_i = \frac{n_{id}}{n_d} \times log\frac{N}{n_i}$$

where n_{id} is the number of occurrences of word i in document d, n_d is the total number of words in the document d, n_i is the number of occurrences of term i in the whole database and N is the number of documents in the whole database. The weighting is a product of two terms: the word frequency $\frac{n_{id}}{n_d}$, and the inverse document frequency $log\frac{N}{n_i}$ The intuition is that word frequency weights words occurring often in a particular document, and thus describe it well, whilst the inverse document frequency scales down the weight of words that appear often in the database. At the retrieval stage documents are ranked by their normalized scalar product (cosine of angle) between the query vector V_q and all document vectors V_d in the database. In our case the query vector is given by the keyshape contained in the object query, and the objects in the database are ranked according to the similarity of their weighted vectors to this query vector.

6.3.2 Experiments

In this section, we describe a set of experiments whose purpose is to validate the BoF based method proposed in this chapter. Firstly, we have studied the influence of the vocabulary size in the retrieval accuracy. Then, we evaluate the method in the two databases presented in chapter 4. The robustness to shape deformations is discussed in the sequel. The algorithms that we have described in the previous sections have been implemented using MATLAB software. The framework encloses an off-line feature extraction algorithm and a vector quantization algorithm, and an on-line retrieval process. We have tested the performance of the method using the two descriptors we proposed in this thesis; the curve and the Geodesic cord.

In the following we will call:

- BoF_Curve, BoF based method using curve descriptor.
- BoF_Gcord, BoF based method using Geodesic cord descriptor.

Influence of the vocabulary size

As mentioned earlier, in order to determine the vocabulary size used in this method, we run K-means several times with different number of desired representative vectors (K) and different sets of initial cluster centers. We select the final clustering giving the lowest empirical risk in matching. Here, we have measured the performance of the method with respect to the vocabulary size and we have computed the overall error rates as a function of the number of clusters K.

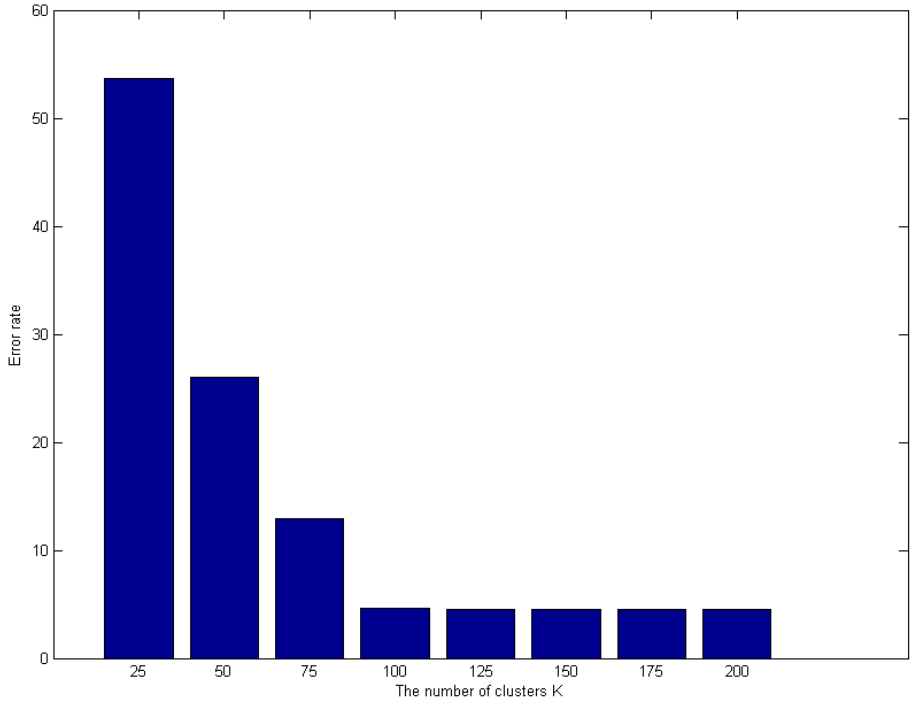


Figure 6.2 – *The lowest overall error rate found for different choices of k.*

In Figure 6.2 we present the overall error rates. We compute the percentage of success for the first (PF) and the second (PS) retrieved

items, i.e. the probability of the first and second elements in the return vector to be relevant to the given query, and average them over the whole set of queries. Each point in Figure 6.2 is the best of 10 random trials of k-means. The error rate only improves slightly as we move from k = 100 to k = 200. We therefore assert that k = 100 presents a good trade-off between accuracy and speed[1].

Comparison with related work

In this section, we present the experimental results of the proposed BoF method compared with some state-of-the-art shape-matching algorithms and the proposed belief function based methods. The evaluation criteria used in this section are the Precision vs Recall plots. We have used the same databases used in chapter 4: the TOSCA data set for non-rigid shapes and the SHREC07 shape benchmark for partially similar models.

Results on the TOSCA dataset Figure 6.3 shows the Precision-recall curves comparing the BoF based method to method presented in chapter 4. We find that the BoF method slightly outperforms the belief function based method on the TOSCA dataset.

Results on the SHREC07 dataset Quantitatively, we compare the Precision vs Recall plot of our approach with other methods competing the SHREC07 contest. Such a plot is the average of the 30 Precision vs Recall plots corresponding to the 30 models of the query-set. Figure 6.4 shows the these plots. We find here that the belief based methods outperform the BoF methods. The belief based method is more suited to partial

[1]The full processing time of a query takes less than 12 seconds to get predicted labels on the whole database using the nearest neighbor object with k = 100 using a PC with a 3 Ghz Core 2 Duo processor with 3 GB memory.

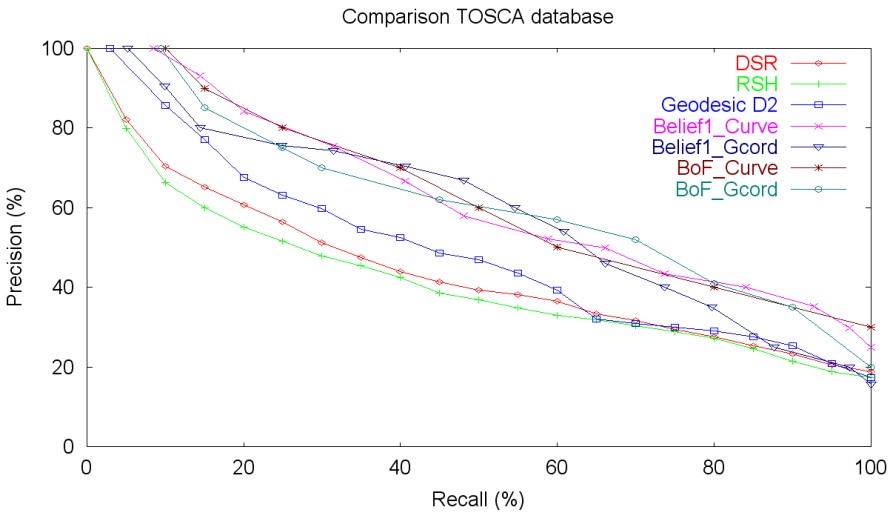


Figure 6.3 – *Precision vs Recall plots comparing the BoF based method to the RSH, the D2, the DSR algorithm and the belief based method presented in chapter 4 on the TOSCA dataset.*

shape matching. This is due to the significance of parts that is taken into account in the matching process when comparing two 3D-object using the belief function based method.

6.4 Application to 3D-object Classification

This section presents a BoF method for 3D-object classification. The classification is based on conventional machine learning approaches.

6.4.1 Categorization

Once descriptors have been allocated to clusters to form feature vectors, we scale down the problem of generic categorization to that of multi-class supervised learning, with as many classes as defined shape categories. The categorizer performs two separate steps in order to predict the classes of unlabeled 3D-object: training and testing. During training, labeled data are sent to the classifier and used to adapt a statistical

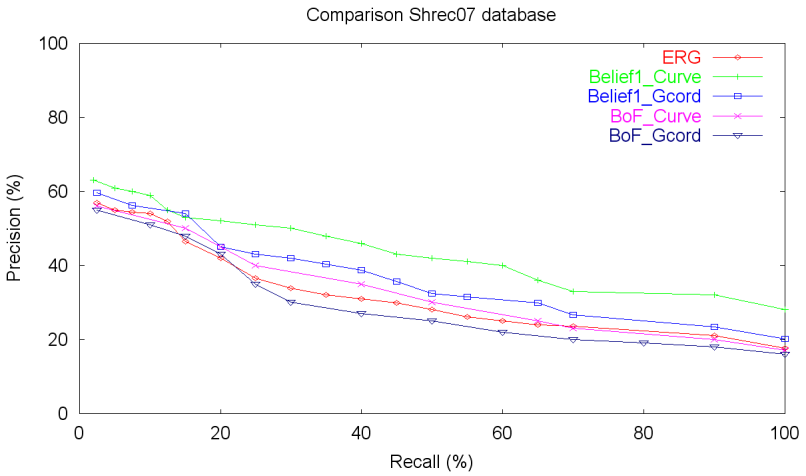


Figure 6.4 – *Precision vs Recall plot on the SHREC07 dataset.*

decision procedure for distinguishing categories. Among many available classifiers, we compared the Naïve Bayes classifier for its simplicity and its speed, and the Support Vector Machine since is it often known to produce state-of-the-art results in high-dimensional problems.

Categorization by Naïve Bayes

The Naïve Bayes Classifier [Lew98] is a probabilistic classifier based on the Bayesian theorem. To demonstrate the concept of shape categorization using Naïve Bayes classifier, let us assume we have a set of labeled 3D-object $O = \{O_i\}$ and an vocabulary $V = \{v_t\}$ of representative keyshapes. Each descriptor extracted from a 3D-object is labeled with the keyshape to which it lies closest in feature space. We count the number $N(t,i)$ of times keyshape v_t occurs in object O_i. To categorize a new 3D-object, we apply Bayes's rule and take the largest a posteriori score as the prediction:

$$P(C_j/O_i) \propto P(O_i/C_j)P(C_j) = P(C_j)\prod_{t=1}^{|V|} P(v_t/C_j)^{N(t,i)}. \qquad (6.1)$$

It is evident in this formula that Naïve Bayes requires estimates of the class conditional probabilities of keyshape v_t given category C_j. In

order to avoid probabilities of zero, these estimates are computed with Laplace smoothing:

$$P(v_t/C_j) = \frac{1 + \sum_{O_i \in C_j} N(t,i)}{|V| + \sum_{s=1}^{|V|} \sum_{O_i \in C_j} N(s,i)}. \tag{6.2}$$

Categorization by SVM

The SVM classifier is a classification method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels with maximal margin [Vap98]. In order to apply the SVM to multi-class problems we take the one-against-all approach. Given an m class problem, we train m SVM's, each distinguishes object from some category i from objects from all the other $m-1$ categories j not equal to i. Given an object to be classified, we assign it to the class with the largest SVM output.

6.4.2 Experiments

In this section, we give results from three experiments. In the first experiment, we test the effect of the number of clusters on the classifier accuracy and analyze the effectiveness of the Naïve Bayes classifier. In the second experiment, we analyze the performance of the SVM on the same problem. These experiments were conducted on the SHREC07 dataset (earlier presented in chapter 4). In the last experiment we present results on the fourteen class dataset employed in [BGM*06].

The descriptor used in this section is the Geodesic cord descriptor.

We used the overall error rate performance measures to evaluate our multi-class classifiers.

The overall error rate:

$$R = 1 - \frac{\sum_{j=1}^{N_C} M_{jj}}{\sum_{j=1}^{N_C} |C_j|} \tag{6.3}$$

Each performance metric was evaluated with 10-fold cross validation.

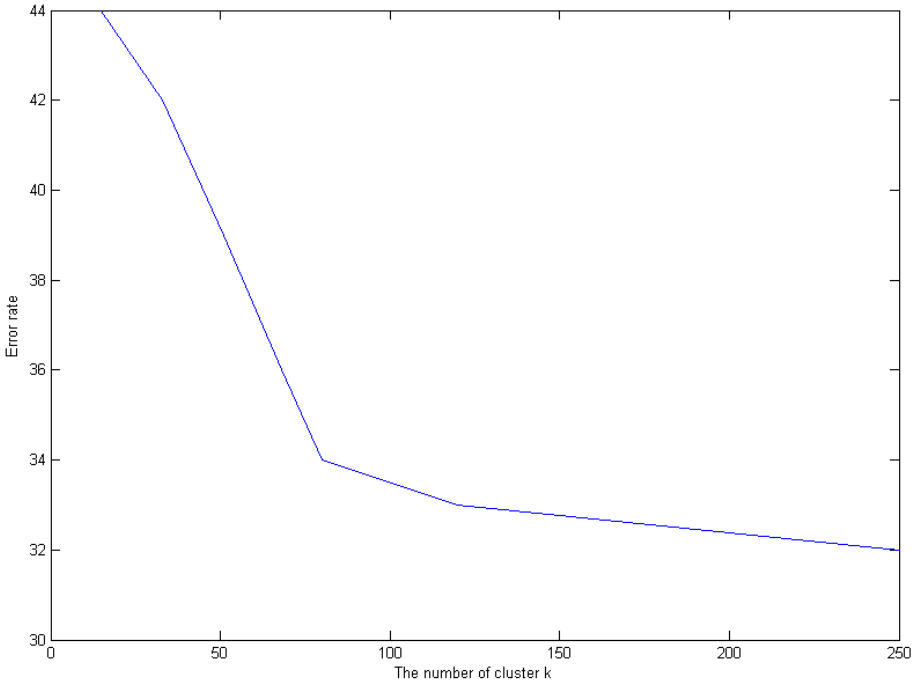


Figure 6.5 – *The lowest overall error rate found for different choices of k.*

Naïve Bayes Results

In Figure 6.5 we present the overall error rates using Naïve Bayes as a function of the number of clusters k. Each point in Figure 6.5 is the best of 10 random trials of k-means. The error rate only improves slightly as we move from k = 100 to k = 250. We therefore assert that k = 100 presents a good trade-off between accuracy and speed[1].

[1]It takes less than 10 seconds to get predicted labels on the whole database using the Naïve Bayes classifier with k = 100 using a PC with a 3 Ghz Core 2 Duo processor with 3 GB memory.

SVM Results

As expected the SVM performance surpass the performance of Naïve Bayes classifier, reducing the overall error rate from 33 to 24%.

In training this SVM we used the same best vocabulary with k=100 as for Naïve Bayes. We compared linear, quadratic and cubic SVM's and found that linear method gave the best performance. The penalty parameter Cst of the SVM classifier was determined for each SVM and values of around $Cst = 0.006$ typically gave the best results.

From a qualitative point of view, Figure 6.6 shows the different behavior of the two classifiers on a set of twenty objects selected from the database. Rows in Figure 6.6 correspond to 3D-objects (the objects correspond to the last column in Figure 5.2), and columns correspond to the different categories shown in the rows of Figure 5.2 (in the same order). In this visualization, the lightness of each element $(i; j)$ is proportional to the magnitude of the probability of the belong of the object i to the category j. Lighter elements represent better matches, while hot elements indicate worse matches. The lightness of the diagonal squares of the matrix proves the effectiveness of the classifier.

One can notice in this visualization that the SVM classifier (Figure 6.6b) shows more robust results compared with the Naïve Bayes classifier (Figure 6.6a).

Comparison with related work

In this experiment we compare our method effectiveness to methods proposed in Biosotti et al. framework [BGM*06] earlier presented in chapter 5 section 5.5.3. The results of the experiment are shown in Table 6.1. Each entry is related to the performance of a given shape descriptor (enumerated in the first row). The performance is evaluated in terms of classification rate (i.e. the percentage of object models which are

	Methods				
	SH	LF	MRG	ERG	our SVM classifier
Rate	89%	88 %	88 %	83 %	91%
Mean rank	2%	2 %	2 %	2 %	1.62%

Table 6.1 – *Classification result comparison with related work.*

correctly classified). Notice that these rates are computed when the position of the correct class with respect to the rank identified by the classifier is set to be 2 as in [BGM*06]. Our classifier show the highest classification rate 91%.

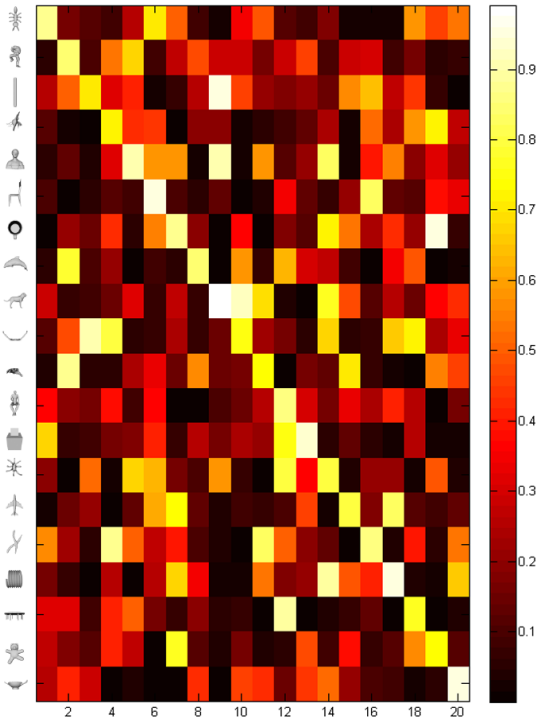
Conclusion

In this chapter, we have presented a method for matching 3D-objects based on the BoF techniques. We have tested this method on two tasks. First, we have presented results for retrieving 3D-objects from 3 datasets the shreco7, the Tosca and the Sumner datasets. These results show the effectiveness of our approach and clearly demonstrate that the method is robust to non-rigid and deformable shapes. Then, we have applied our method in the classification task. For that, two well-known classifier: the Naïve Bayes and the SVM have been learned and tested on the shreco7 dataset. The results show the effectiveness of the SVM classifier compared with the Bayesian one.

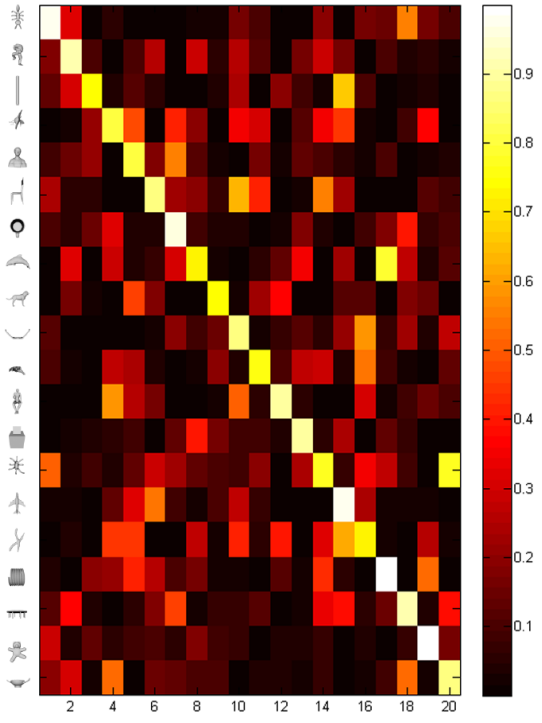
French conclusion

Dans ce chapitre, nous avons présenté une méthode de mise en correspondance d'objets 3D basée sur la représentation par sacs de mots. Nous avons expérimenté cette méthode dans le cadre de la recherche et la classification d'objets.

Les résultats obtenus sur les deux différentes bases d'objets 3D (TOSCA et SHREC07) montrent l'efficacité et la robustesse de notre méthode vis à vis aux différentes transformations qu'un objet 3D peut subir.



(a)



(b)

Figure 6.6 – *Example of some 3D-objects classification. (a) Naïve Bayes classifier results. (b) SVM classifier results*

7 Conclusion

In this thesis, we have developed novel approaches for 3D object matching. The approaches are robust to isometric transformation as well as non rigid ones. First, we have proposed two 3D-shape descriptors, namely *Curve descriptor* and *Geodesic cord descriptors*. These descriptors which reflect the local geometric characteristics of given 3D-objects, are computed on patches of the objects. To separate the different patches of a given object, we proposed the use of feature point extraction algorithm.

The proposed descriptors are based on the intrinsic propriety of the shapes of 3D-patches. It is in turn invariant to the rigid as well as non-rigid transformations of the surface.

The curve descriptor consists to represent a 3D-patch by an indexed collection of closed level curves in $\mathbb{R}^3$ extracted around each feature point. Then, tools from shape analysis of curves are applied to analyze and to compare curves. We used an extension of the Riemannian framework proposed by Joshi et al. [JKSJ07] to: (i) compute distances between curves to quantify differences in their shapes, (ii) find optimal deformations between curves, and (iii) define and compute average of a given set of curves.

The Geodesic cord descriptor represents the shape signature for a 3D-patch as a probability distribution sampled from a shape function

measuring intrinsic properties of the 3D-patch. The distribution is sampled from an intrinsic distance function on the 3D-surface. The distance function used here is based on the geodesic distances between the feature point and all the points on the 3D-patch surface.

In this thesis, we also proposed to use the belief functions, as fusion technique, to define a global distance between 3D-objects. We have experimented this technique in the retrieval and classification tasks. Our approach is based on the use of information extracted from the different patches (descriptors) of the 3D-object. Each patch, provides an information source regarding the shape of that object. A combination of these information is necessary in order to increase the recognition rate of the object. The information fusion can be seen as a problem of merging classification results of each patch (or fusion of classifiers).

We have also proposed the use of Bag of Feature techniques in 3D-object retrieval and classification. We have tested this method on two tasks. First, we have presented results for retrieving 3D-objects from 3 datasets the SHREC07, the Tosca and the Sumner datasets. These results show the effectiveness of our approach and clearly demonstrate that the method is robust to non-rigid and deformable shapes. Then, we have applied our method in the classification task. For that, two well-known classifier: the Naïve Bayes and the SVM have been learned and tested on the SHREC07 dataset.

7.1 Open problems and directions

In this thesis, we presented new solutions based on curve and geodesic cord representation for 3D-shape description. We proposed two methods, the belief functions and bag of features method, for 3D-object matching. We assessed their validity with regard to the state-of-the-art techniques.

Towards shape analysis of parameterized surfaces We proposed to analyze shapes of 3D-surfaces using shapes of curves extracted around a set of feature points. 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. [KKDS10] where a novel Riemannian framework for shape analysis of parameterized surfaces is introduced. We are confidant in the fact that such an approach can be integrated and automated in our framework.

Towards combining belief function and BoF methods In the context of shape similarity computing, we introduced two different methods: the belief function based method and the BoF based one. We believe that a combination of both methods could improve the matching results. Specific strategies of combination can be applied. We can integrate the belief function in the BoF process by modeling the keyshapes by mass functions. We believe that the combination should be deeply studied to significantly extend current approaches.

Towards spatial information in the BoF approach 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 the BoF matching results. Computer vision techniques for graph matching [RLAB10] can be applied to preserve the spatial relation between features in a given 3D-object.

French conclusion

Dans cette thèse de doctorat, nous avons développé une nouvelle approche pour la mise en correspondance des objets 3D en présence des

transformations non-rigides et des modèles partiellement similaires. L'approche que nous proposons est composée de deux phases. Une première phase pour la description d'objets et une deuxième phase de mesure de similarité.

Pour décrire un objet 3D, nous avons choisi une méthode basée sur des descripteurs locaux. La méthode consiste à extraire d'un objet 3D un ensemble de points caractéristiques pour lesquels deux descripteurs locaux sont calculés. Le premier descripteur *Geodesic cord descriptor* étudie la distribution des distances géodésiques entre un point caractéristique et l'ensemble des points de la surface de l'objet 3D. Le deuxième descripteur *Curve based descriptor* permet de représenter la surface 3D de l'objet par un ensemble de courbes. La forme de ces courbes est analysée à l'aide d'outils issus de la géométrie Riemannienne.

Pour mesurer la similarité entre les objets 3D, nous avons utilisé deux techniques différentes dont l'une est basée sur les fonctions de croyance et l'autre est basée sur les *sac-de-mots*. Les fonctions de croyance offrent un cadre naturel pour la représentation et la combinaison d'informations issues de différentes sources. Ce cadre nous a permis de calculer une similarité globale entre les objets 3D en combinant les différentes informations fournies par les descripteurs. La technique basée *sac-de-mots* consiste à quantifier les descripteurs locaux calculés sur les points caractéristiques et permettent de représenter les objets 3D sous la forme d'histogrammes des fréquences d'apparition des mots visuels. Cette représentation permet le calcul efficace de similarités entre objets 3D.

Afin de valider notre approche nous l'avons adaptée à deux applications différentes à savoir la recherche et la classification d'objets 3D. Les résultats obtenus sur différent benchmarks montrent une efficacité et une pertinence comparés avec les autres méthodes de l'état-de-l'art.

Bibliography

[AAV04] Adán M., Adán A., Vázquez A. S.: Clustering using wcc models. In *Structural, Syntactic, and Statistical Pattern Recognition*, Fred A., Caelli T., Duin R. P. W., Campilho A., Ridder D. d., (Eds.), vol. 3138 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2004, pp. 921–929. (Cited page 25.)

[ADBP03] Assfalg J., Del Bimbo A., Pala P.: Retrieval of 3d objects using curvature maps and weighted walkthroughs. In *Proceedings of the 12th International Conference on Image Analysis and Processing* (Washington, DC, USA, 2003), IEEE Computer Society, pp. 348–. (Cited page 25.)

[AEKH09] Andrei Z., Edmond B., Kiran V., Horaud R.: Surface feature detection and description with applications to mesh matching. In *International Conference on Computer Vision and Pattern Recognition, CVPR'09, June, 2009* (Miami, Etats-Unis, June 2009), IEEE, pp. 373–380. (Cited page 23.)

[BB10] Bronstein A. M., Bronstein M. M.: Spatially-sensitive affine-invariant image descriptors. In *Proceedings of the 11th European conference on Computer vision: Part II* (Berlin, Heidelberg, 2010), ECCV'10, Springer-Verlag, pp. 197–208. (Cited page 23.)

[BBBK08a] Bronstein A. M., Bronstein M. M., Bruckstein A., Kimmel R.: Analysis of two-dimensional non-rigid shapes. *Int. J. Comput. Vision 78* (June 2008), 67–88. (Cited page 33.)

[BBBK08b] Bronstein A. M., Bronstein M. M., Bruckstein A., Kimmel R.: Partial similarity of objects, or how to compare a centaur to a horse. *International Journal of Computer Vision* (July 2008). (Cited pages xi, 32, 33, 34, 71, and 72.)

[BBGO11] Bronstein A. M., Bronstein M. M., Guibas L. J., Ovsjanikov M.: Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Trans. Graph. 30* (February 2011). (Cited pages 24 and 28.)

[BBK07] Bronstein A. M., Bronstein M. M., Kimmel R.: Efficient computation of isometry-invariant distances between surfaces. *IEEE Trans. VCG 13/5* (2007), 902–913. (Cited pages 75, 76, and 102.)

[BDBP09] Berretti S., Del Bimbo A., Pala P.: 3d mesh decomposition using reeb graphs. *Image Vision Comput. 27* (September 2009). (Cited page 22.)

[BGM*06] Biasotti S., Giorgi D., Marini S., Spagnuolo M., Falcidieno B.: A comparison framework for 3d object classification methods. In *Multimedia content representation, classification and security* (2006), vol. 4105, pp. 314–321. (Cited pages 106, 126, 128, and 129.)

[BH84] Berthold K., Horn P.: Extended gaussian images. *Proceedings of the IEEE 72*, 2 (1984), 1671–1686. (Cited page 26.)

[BM05] Biasotti S., Marini S.: 3d object comparison based on shape descriptors. *International Journal of Computer Applications in Technology 23* (2005), 57–69. (Cited page 107.)

[BMM*03] Biasotti S., Marini S., Mortara M., Patane G., Spagnuolo M., Falcidieno B.: 3d shape matching through topological structures. vol. 2886. 2003, pp. 194–203. (Cited page 22.)

[Can99] Canterakis N.: 3d zernike moments and zernike affine invariants for 3d image analysis and recognition. In *In 11th Scandinavian Conf. on Image Analysis* (1999), pp. 85–93. (Cited pages 3 and 7.)

[CCFM08] Castellani U., Cristani M., Fantoni S., Murino V.: Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Computer Graphics Forum 27*, 2 (2008), 643–652. (Cited page 23.)

[CMEH*03] Cole McLaughlin K., Edelsbrunner H., Harer J., Natarajan V., Pascucci V.: Loops in reeb graphs of 2-manifolds. In *ACM Symposium on Computational Geometry* (2003), pp. 344–350. (Cited page 40.)

[CTSO03] Chen D.-Y., Tian X.-P., Shen Y.-T., Ouhyoung M.: On Visual Similarity Based 3D Model Retrieval, 2003. (Cited pages x and 19.)

[DBD08] Dugelay J.-L., Baskurt A., Daoudi M.: *3D Object Processing: Compression, Indexing and Watermarking*. 2008. (Cited pages 3 and 7.)

[DD01] Dietmar S., Dejan V.-V.: 3d model retrieval with spherical harmonics and moments. In *DAGM* (2001), Springer-Verlag, pp. 392–397. (Cited pages 27 and 76.)

[Dem67] Dempster A. P.: Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics 38* (1967), 325–339. (Cited page 65.)

[DHS00] Duda O., Hart P.-E., Stork D.-G.: Pattern classification. In *John Wiley Sons* (2000). (Cited page 119.)

[FADV07] Filali Ansary T., Daoudi M., Vandeborre J.-P.: A bayesian 3D search engine using adaptive views clustering. *IEEE Transactions on Multimedia 9*, 1 (January 2007), 78–88. (Cited pages x, 20, and 32.)

[GBAL09] Gebal K., Baerentzen J.-A., Aanaes H., Larsen R.: Shape analysis using the auto diffusion function. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), SGP '09, Eurographics Association, pp. 1405–1413. (Cited page 24.)

[GJ90] Garey M.-R., Johnson D.-S.: *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990. (Cited page 21.)

[Glo09] Glomb P.: Detection of interest points on 3d data: Extending the harris operator. In *Computer Recognition Systems 3*, Kurzynski M., Wozniak M., (Eds.), vol. 57 of *Advances in Soft Computing*. Springer Berlin / Heidelberg, 2009, pp. 103–111. (Cited page 23.)

[GSCO07] Gal R., Shamir A., Cohen-Or D.: Pose oblivious shape signature. *IEEE Transactions on Visualization and Com-*

puter Graphics 13, 2 (March-April 2007), 261–271. (Cited page 107.)

[HKDH04] Huber D., Kapuria A., Donamukkala R., Hebert M.: Parts-based 3d object classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2004). (Cited page 101.)

[HRKM03] Healy D.-M., Rockmore D.-N., Kostelec P.-J., Moore S.: FFTs for the 2-Sphere-Improvements and Variations. *Journal of Fourier Analysis and Applications 9*, 4 (July 2003), 341–385. (Cited page 27.)

[HS88] Harris C., Stephens M.: A Combined Corner and Edge Detection. In *Proceedings of The Fourth Alvey Vision Conference* (1988), pp. 147–151. (Cited page 23.)

[IB10] Ivan S., Benjamin B.: A robust 3d interest points detector based on harris operator. In *Eurographics 2010 Workshop on 3D Object Retrieval (3DOR'10)* (2010), Eurographics Association, pp. 7–14. (Cited page 23.)

[JH99] Johnson A., Hebert M.: Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE transactions on Pattern Analysis and Machine Intelligence 21* (1999). (Cited pages x, 27, and 28.)

[JKSJ07] Joshi S., Klassen E., Srivastava A., Jermyn I.: Removing shape-preserving transformations in square-root elastic (sre) framework for shape analysis of curves. In *EMMCVPR* (2007), pp. 387–398. (Cited pages 4, 8, 44, 45, and 133.)

[KF02] Kazhdan M., Funkhouser T.: Harmonic 3d shape matching. In *ACM SIGGRAPH 2002 conference abstracts and applications* (New York, NY, USA, 2002), SIGGRAPH '02, ACM, pp. 191–191. (Cited pages x, 26, 27, and 107.)

[KI93] Kang S.-B., Ikeuchi K.: The complex egi: a new representation for 3-d pose determination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 15*, 7 (July 1993), 707 –721. (Cited page 26.)

[KKDS10] Kurtek S., Klassen E., Ding Z., Srivastava A.: A novel riemannian framework for shape analysis of 3d objects. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (june 2010), pp. 1625 –1632. (Cited page 135.)

[KLT05] Katz S., Leifman G., Tal A.: Mesh segmentation using feature point and core extraction. *The Visual Computer 25* (2005), 865–875. (Cited page 39.)

[KS06] Klassen E., Srivastava A.: Geodesics between 3d closed curves using path-straightening. In *European Conference on Computer Vision (ECCV)* (2006), pp. 95–106. (Cited page 47.)

[KvD79] Koenderink J.-J., van Doorn A. J.: The internal representation of solid shape with respect to vision. 211–216. (Cited pages 20 and 24.)

[Lau98] Laurent Y.: Computable elastic distance between shapes. *SIAM Journal of Applied Mathematics 58*, 2 (1998), 565–586. (Cited page 44.)

[Lew98] Lewis D.: Naïve bayes at forty: The independence assumption in information retrieval. *ECML* (1998). (Cited page 125.)

[LHGQ06] Li X., He Y., Gu X., Qin H.: Curves-on-surface: A general shape comparison framework. In *International Conference on Shape Modeling and Applications (SMI)* (Los Alamitos, CA, USA, 2006). (Cited page 42.)

[LZQ06] Liu Y., Zha H., Qin H.: Shape topics: A compact representation and new algorithms for 3d partial shape retrieval. In *Computer Society Conference on Computer Vision and Pattern Recognition* (2006). (Cited pages 34, 115, and 116.)

[MAS01] Michael E., Ayellet T., Sigal A.: Content based retrieval of vrml objects - an iterative and interactive approach. pp. 97–108. (Cited pages 3 and 7.)

[MM06] Michor P.-W., Mumford D.: Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc. 8* (2006), 1–48. (Cited page 44.)

[MP02] Mortara M., Patanè G.: Affine-invariant skeleton of 3d shapes. In *IEEE Shape Modeling International* (2002), pp. 245–252. (Cited page 39.)

[MPB07] Marini S., Paraboschi L., Biasotti S.: Shape retrieval contest 2007: Partial matching track. *SHREC (in conjunction with IEEE Shape Modeling International)* ((2007)), 13–16. (Cited pages 75 and 76.)

[MYTT01] Masaki H., Yoshihisa S., Taku K., Tosiyasu K.: Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH* (2001), pp. 203–212. (Cited pages 22 and 107.)

[OBBG09] Ovsjanikov M., Bronstein A. M., Bronstein M. M., Guibas L.-J.: Shapegoogle: a computer vision approach

for invariant shape retrieval. In *Workshop on Nonrigid Shape Analysis and Deformable Image Alignment (NORDIA)* (2009). (Cited pages 28, 34, and 115.)

[OFCD02] Osada R., Funkhouser T., Chazelle B., Dobkin D.: Shape distributions. *ACM Transactions on Graphics (TOG) 21(4)* (2002), 807–832. (Cited pages x, 16, 38, 51, and 78.)

[OOFB08] Ohbuchi R., Osada K., Furuya T., Banno T.: Salient local visual features for shape-based 3d model retrieval. In *International Conference on Shape Modeling and Applications (SMI)* (2008). (Cited pages 34, 115, and 116.)

[PM00] Pelleg D., Moore A.: X-means: Extending k-means with efficient estimation of the number of clusters. In *International Conference on Machine Learning* (2000). (Cited page 120.)

[PR99] Paquet E., Rioux M.: Nefertiti: a query by content system for three-dimensional model and image databases management. *Image and Vision Computing 17* (1999), 157–166. (Cited pages 15 and 51.)

[RCB05] Ricard J., Coeurjolly D., Baskurt A.: Generalizations of angular radial transform for 2d and 3d shape retrieval. *Pattern Recogn. Lett. 26* (October 2005). (Cited pages 3 and 7.)

[Ree46] Reeb G.: Sur les points singuliers d'une forme de pfaff complètement intégrable ou d'une fonction numérique. *In Comptes Rendus, Académie des Sciences de Paris 222* (1946), 847–849. (Cited page 21.)

[RLAB10] Revaud J., Lavoué G., Ariki Y., Baskurt A.: Scale-invariant proximity graph for fast probabilistic object

recognition. In *Conference on Image and Video Retrieval* (2010). (Cited page 135.)

[Rus07] Rustamov R. M.: Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the fifth Eurographics symposium on Geometry processing* (2007). (Cited page 28.)

[Sha76] Shafer G.: A mathematical theory of evidence. *Princeton University Press* (1976). (Cited pages 65, 93, and 98.)

[SK94] Smets P., Kennes R.: The transferable belief model. *Artificial Intelligence 66* (1994), 191–234. (Cited pages 66, 68, 69, and 70.)

[SKK91] Shinagawa Y., Kunii T.-L., Kergosien Y.-L.: Surface coding based on morse theory. *IEEE Comput. Graph. Appl. 11* (September 1991), 66–78. (Cited page 22.)

[SKK01] Sebastian T.-B., Klein P.-N., Kimia B.-B.: Recognition of shapes by editing shock graphs. In *In IEEE International Conference on Computer Vision* (2001), pp. 755–762. (Cited page 21.)

[SOG09] Sun J., Ovsjanikov M., Guibas L.: A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum 28*, 5 (2009), 1383–1392. (Cited pages xi, 24, and 29.)

[SP00] Shimshoni I., Ponce J.: Probabilistic 3d object recognition. *Int. J. Comput. Vision 36* (January 2000), 51–70. (Cited page 31.)

[SP04] Sumner R., Popovic J.: Deformation transfer for triangle meshes. In *International Conference on Computer Graphics and Interactive Techniques* (2004). (Cited page 102.)

[TCF09] Toldo R., Castellani U., Fusiello A.: 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 34, 115, and 116.)

[Tun05] Tung T.: *Rapport de thèse: Indexation 3D de bases de données d'objets par graphes de Reeb améliorés*. 2005. (Cited pages x and 22.)

[TV04] Tangelder J.-W.-H., Veltkamp R.-C.: A survey of content based 3d shape retrieval methods. In *In Shape Modeling International* (2004), pp. 145–156. (Cited page 21.)

[TVD09] Tierny J., Vandeborre J.-P., Daoudi M.: Partial 3D shape retrieval by reeb pattern unfolding. *Computer Graphics Forum - Eurographics Association 28* (March 2009), 41–55. (Cited pages xi, 22, 37, 39, and 40.)

[Vap98] Vapnik V.: Statistical learning theory. *Wiley* (1998). (Cited pages 120 and 126.)

[VFJZ01] Vailaya A., Figueiredo M.-A.-T., Jain A.-K., Zhang H.-J.: Image classification for content-based indexing. *IEEE Transactions on Image Processing 10* (2001), 117–130. (Cited page 99.)

[Vra03] Vranic D.-V.: An improvement of rotation invariant 3d-shape based on functions on concentric spheres. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International*

Conference on (2003), vol. 3, pp. III – 757–60 vol.2. (Cited page 27.)

[Vra04] Vranic D.: *3D Model Retrieval*. PhD thesis, Universitat Leipzig, May 2004. (Cited page 78.)

[YC98] Yi J. H., Chelberg D. M.: Model-based 3d object recognition using bayesian indexing. *Computer Vision and Image Understanding 69*, 1 (1998), 87 – 105. (Cited page 30.)

[YM05] Yezzi A.-J., Mennucci A.: Conformal metrics and true "gradient flows" for curves. In *IEEE International Conference on Computer Vision* (2005), vol. 1, pp. 913–919. (Cited page 44.)

[ZD98] Zouhal L.-M., Denoeux T.: An evidence-theoretic k-nn rule with parameter optimization. *IEEE Transactions on Systems 28* (May 1998), 263–271. (Cited pages 91 and 93.)

[ZP02] Zaharia T., Preteux F.: Indexation de maillages 3d par descripteurs de forme. In *In RFIA* (2002), pp. 48–57. (Cited pages 17 and 24.)

[ZP04] Zaharia T., Preteux F.: Descripteurs de forme : Etude comparée des approches 3d et 2d/3d. In *In RFIA* (2004). (Cited page 18.)

[ZWS95] Zhang K., Wang J., Shasha D.: On the editing distance between undirected acyclic graphs and related problems. 1995, pp. 395–407. (Cited page 21.)

Titre Classification et recherche d'objets 3D

Résumé Une nouvelle approche pour la mise en correspondance des objets 3D en présence des transformations non-rigides et des modèles partiellement similaires est proposée dans le cadre de cette thèse. L'approche est composée de deux phases. Une première phase pour la description d'objets et une deuxième phase de mesure de similarité. Pour décrire un objet 3D, nous avons choisi une méthode basée sur des descripteurs locaux. La méthode consiste à extraire d'un objet 3D un ensemble de points caractéristiques pour lesquels deux descripteurs locaux sont calculés. Le premier descripteur *Geodesic cord descriptor* représente la distribution des distances géodésiques entre un point caractéristique et l'ensemble des points de la surface de l'objet 3D. Le deuxième descripteur *Curve based descriptor* permet de représenter la surface 3D de l'objet par un ensemble de courbes. La forme de ces courbes est analysée à l'aide d'outils issus de la géométrie Riemannienne. Pour mesurer la similarité entre les objets 3D, nous avons utilisé deux techniques différentes dont l'une est basée sur les fonctions de croyance et l'autre est basée sur les *sac-de-mots*. Afin de valider notre approche nous l'avons adaptée à deux applications différentes à savoir la recherche et la classification d'objets 3D. Les résultats obtenus sur différent benchmarks montrent une efficacité et une pertinence comparés avec les autres méthodes de l'état-de-l'art.

Mots-clés Objets 3D, classification, indexation, fonction de croyances, sac à mots.

Title Contributions to 3D-shape matching, retrieval and classification

Abstract Three dimensional object representations have become an integral part of modern computer graphic applications such as computer-aided design, game development and audio-visual production. At the Meanwhile, the 3D data has also become extremely common in fields such as computer vision, computation geometry, molecular biology and medicine. This is due to the rapid evolution of graphics hardware and software development, particularly the availability of low cost 3D scanners which has greatly facilitated 3D model acquisition, creation and manipulation. Content-based search is a necessary solution for structuring, managing these multimedia data, and browsing within these data collections. In this context, we are looking for a system that can automatically retrieve the 3D-models visually similar to a requested 3D-object. Existing solutions for 3D-shape retrieval and classification suffer from high variability towards shape-preserving transformations like affine or isometric transformations (non-rigid transformations). In this context, the aim of my research is to develop a system that can automatically retrieve quickly and with precision 3D models visually similar to a 3D-object query. The system has to be robust to non-rigid transformation that a shape can undergo.During my PhD thesis:We have developed a novel approach to match 3D objects in the presence of non-rigid transformation and partially similar models. We have proposed to use a new representation of 3D-surfaces using 3D curves extracted around feature points. Tools from shape analysis of curves are applied to analyze and to compare curves of two 3D-surfaces. We have used the belief functions, as fusion technique, to define a global distance between 3D-objects. We have also experimented this technique in the retrieval and classification tasks. We have proposed the use of Bag of Feature techniques in 3D-object retrieval and classification.

Keywords 3D-shape, classification, retrieval, belief functions, bag of features.