



HAL
open science

Reeb Graph Modeling of 3-D Animated Meshes and its Applications to Shape Recognition and Dynamic Compression

Meha Hachani

► **To cite this version:**

Meha Hachani. Reeb Graph Modeling of 3-D Animated Meshes and its Applications to Shape Recognition and Dynamic Compression. Networking and Internet Architecture [cs.NI]. Université Montpellier; École nationale d'ingénieurs de Tunis (Tunisie), 2015. English. NNT: 2015MONTTS152 . tel-02062224

HAL Id: tel-02062224

<https://theses.hal.science/tel-02062224>

Submitted on 8 Mar 2019

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

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

Université Tunis El Manar



المدرسة الوطنية للمهندسين بتونس
école nationale d'ingénieurs de Tunis

مخبر أنظمة الاتصالات
6'COM
Laboratoire des Systèmes de Communications

THÈSE

Présentée pour obtenir le titre de

DOCTEUR

DE L'ÉCOLE NATIONALE D'INGÉNIEURS DE TUNIS

SPÉCIALITÉ : TÉLÉCOMMUNICATIONS

Par

Meha Hachani

Le 19 décembre 2015

Reeb Graph Modeling of 3-D Animated Meshes and its Applications to Shape Recognition and Dynamic Compression

Pr. Taoufik AGUILI (Sys'Com-ENIT)

Président

Pr. Amel BEN AZZA (Sup'Com)

Rapporteur

Pr. Mohamed DAOUDI (Télècom-Lille1)

Rapporteur

Pr. Gilles GESQUIERE (UL-LYON2)

Examineur

Pr. Azza OULED ZAID (Sys'Com –ENIT)

Directeur de thèse

Pr. William PUECH (LIRMM- UM)

Co-directeur de thèse

B.P. 37 le Belvédère 1002 Tunis Tunisie

Tél. : 216 71 874 700 / 71 875 475 :الهاتف

Email : Enit@ enit.rnu.tn

تونس 1002 البلفدار 37 ص ب

الفاكس : 216 71 872 729 Fax :

:البريد الإلكتروني

Acknowledgements

The three-year thesis are achieved only with the efforts deployed by my advisors. Now I've had the opportunity to express the deepest appreciation to my two thesis supervisors for their support, encouragement and more specifically for having made me discover scientific research. During these three years, Pr. Azza Ouled Zaid (Professor at ENIT/Tunis) has been a scientific research guide. Thanks to her I have learned to properly present my research articles. I thank him for his support and the attention that she carried me. I would like to thank also Pr. William Puech (Professor at UM/ Montpellier) for his outstanding human qualities and its encouragements. I thank him also for his help and support to integrate and fit into the lab team during my internships at LIRMM laboratory.

Especial thanks to the reviewers of my manuscript for having accepted this significant task (over a short period) Pr. Mohamed Daoudi (Professor at Telecom lille1) who accepted to review my manuscript despite his time consuming responsibilities and Pr. Amel Ben Azza (Professor at SupCom/ Tunis) who also kindly accepted to sacrifice his precious time for reviewing my manuscript. I would like to thank the committee member Pr. Gilles Gesquière (Professor at Lyon 2 University) has agreed to participate in this jury as an examiner. My sincerest thanks go to Pr. Taoufik Aguilu (SysCom resarch laboratory Director) who made me the honor of being president of the committee. I'm sincerely grateful for all the commitee members.

Fiannaly, I would like to thank my research colleagues at SysCom/ Tunis laboratory and LIRMM/ Montpellier for all the technical exchanges, scientific and their sympathy. Especial thanks to my team member ICAR at LIRMM laboratory for all the good memories that I have shared with them.

Résumé en Français

Le développement fulgurant de réseaux informatiques, a entraîné l'apparition de diverses applications multimédia qui emploient des données 3D dans des multiples contextes. Si la majorité des travaux de recherche sur ces données s'est appuyées sur les modèles statiques, c'est à présent vers Les modèles dynamiques de maillages qu'il faut se tourner. Les séquences de maillages variant au cours de temps représentent un nouvel axe de recherche où leur analyse joue un rôle incontournable, tel que la compression, l'indexation ou encore l'extraction des squelettes.

Les formes dynamiques 3D sont généralement représentées par une séquence de maillages 3D avec une connectivité constante et une information temporelle fournie par une géométrie variable dans le temps. Cette représentation est soumise à une grande variété d'opérations de traitement telles que l'indexation, la segmentation et la compression. Cependant, le maillage triangulaire est une représentation extrinsèque, sensible face aux différentes transformations affines et isométriques. Par conséquent, il a besoin d'un descripteur structurel intrinsèque avant d'être traité par l'une des opérations de traitement mentionnées ci-dessus. Pour relever ces défis, nous nous concentrons sur la modélisation topologique intrinsèque basée sur les graphes de Reeb. Un graphe de Reeb est une représentation graphique, de type squelette, décrivant la structure topologique du modèle 3D. Leurs constructions reposent sur la théorie de Morse, qui définit une fonction continue sur la surface fermée de l'objet. Cette fonction continue permet la segmentation de la surface de l'objet en régions, chaque région est représentée par un nœud. Les nœuds dont les régions associées sont connexes sont liés par une arête. Il existe différentes fonctions continues qui peuvent être utilisées pour la construction du graphe de Reeb des maillages triangulaires.

Représentation par graph de Reeb basée sur la diffusion de la chaleur

Dans le cadre de notre travail, notre principale contribution consiste à définir une nouvelle fonction continue basée sur les propriétés de diffusion de la chaleur. Ce dernier est calculé comme la distance de diffusion d'un point de la surface aux points localisés aux extrémités du modèle 3D qui représentent l'extremum locales de l'objet (*points caractéristiques*) qui sont détectés en utilisant la notion de propagation de la chaleur. La restriction du noyau de la chaleur au domaine temporel rend la fonction proposée intrinsèque et stable contre les perturbations.

Les résultats expérimentaux obtenus sur des modèles 3D dynamiques ont démontré la robustesse et l'efficacité de la fonction scalaire proposée. Cette approche de construction de graph de Reeb peut être extrêmement utile comme descripteur de forme locale pour la reconnaissance de forme 3D. Il peut également être introduit dans un système de compression dynamique basée sur la segmentation. Dans ce contexte, nous exploitons les graphes de Reeb dans deux applications largement

utilisées qui sont la reconnaissance des formes et la compression dynamique 3D.

Application à la reconnaissance de forme 3D

Dans une deuxième partie, nous avons proposé d'exploiter la méthode de construction de graphe de Reeb dans un système de reconnaissance de formes 3D non rigides. L'objectif consiste à segmenter le graphe de Reeb en cartes de Reeb définies comme cartes de topologie contrôlée. Chaque carte de Reeb est projetée vers le domaine planaire canonique qui peut être soit un disque unitaire ou un anneau unitaire selon le type de la carte. Ce dépliage dans le domaine planaire canonique introduit des distorsions d'aire et d'angle. En se basant sur une estimation de distorsion, l'extraction de vecteur caractéristique est effectuée. Nous calculons pour chaque carte un couple de signatures, qui sera utilisé par la suite pour faire l'appariement entre les cartes de Reeb. Pour évaluer l'efficacité de la fonction scalaire utilisée et les signatures proposées, nous avons testé cette méthode sur la base de données la plus connue SHREC 2012 contenant 1200 modèles 3D répartis en 60 classes. Les performances de notre technique ont été évaluées par le calcul de cinq scores : First Tiers, Second Tiers, Les k-meilleurs scores, la mesure E et le gain cumulé. La courbe précision/rappel a montré la capacité de la méthode à retrouver les classes d'objets, il s'agit d'un calcul statistique sur la base de données. Pour effectuer une comparaison fidèle avec d'autres méthodes de l'état de l'art, nous avons testé notre technique de reconnaissance de forme sur plusieurs bases de données tels que : SHREC 2010, SHREC 2011 et MCGill. D'après l'étude expérimentale sur ces bases de données, il a été montré que notre technique donne des résultats satisfaisants du point de vue compromis efficacité et rapidité par rapport aux techniques de l'état de l'art.

Applications à la compression dynamique basée sur la segmentation

Dans une troisième partie, nous avons proposé de concevoir une technique de segmentation, des maillages dynamiques 3D. Cette technique de segmentation est basée sur la même notion de théorie de Morse et de graphe de Reeb. L'idée principale est de détecter les nœuds critiques, en appliquant une analyse topologique des fonctions lisses définies sur la surface de maillage 3D. Le processus de segmentation est effectué en fonction des valeurs de la fonction scalaire proposée dans la première partie. Le principe consiste à dériver une segmentation purement topologique qui vise à partitionner le maillage en des régions rigides tout en estimant le mouvement de chaque région au cours du temps. Pour obtenir une bonne répartition des sommets situés sur les frontières des régions, nous avons proposé d'ajouter une étape de raffinement basée sur l'information de la courbure. Chaque limite de région est associée à une valeur de la fonction qui correspond à un point critique. La valeur optimale de la fonction scalaire doit déterminer une limite qui correspond à un profil de profondeur de concavité sur la surface de l'objet. Il devrait être

proche de la valeur critique de cette fonction scalaire qui correspond au point critique le plus proche. L'objectif visé est de trouver la valeur optimale de cette fonction qui détermine le profil des limites. Cela revient à résoudre un problème d'optimisation qui consiste à minimiser la fonction de concavité. Les résultats expérimentaux effectués sur des maillages 3D dynamiques montrent l'efficacité de notre technique en termes de précision et stabilité contre diverses perturbations y compris les changements topologiques.

La technique de segmentation développée est exploitée dans un système de compression sans perte des maillages dynamiques 3D. Il s'agit de partitionner la première trame de la séquence, considérée comme trame de référence. Chaque région est modélisée par une transformée affine et leurs poids d'animation associés. En combinant linéairement les transformées affines des différentes régions avec les poids d'animation appropriés, nous obtenons le champ de mouvement sur l'ensemble du maillage. Le vecteur partition, associant à chaque sommet l'index de la région auquel il appartient, est compressé par un codeur arithmétique. Les deux ensembles des transformées affines et des poids d'animation sont quantifiés uniformément et compressés par un codeur arithmétique. La première trame de la séquence est compressée en appliquant un codeur de maillage statique. Nous avons proposé de coder les erreurs de prédiction, calculées exclusivement à partir de la première trame de l'animation, en appliquant directement une méthode de compression sans perte des valeurs prédites à virgules flottantes.

Nous avons évalués le système de compression basée sur la segmentation, en effectuant une comparaison avec d'autres méthodes très connues de l'état de l'art. D'après l'étude expérimentale, nous remarquons que notre technique donne des résultats satisfaisants du point de vue compromis débit/distorsion par rapport aux techniques de l'état de l'art.

La suite du travail se concentre sur l'optimisation de notre système de compression en ajoutant une stratégie d'allocation binaire. Afin d'améliorer les performances de notre codeur, la quantification de l'erreur de prédiction temporelle est optimisée en minimisant l'erreur de reconstruction. Ce processus est effectué sur les données de l'erreur de prédiction, qui est divisé en 3 sous-bandes correspondant aux erreurs de prédiction des 3 coordonnées x , y et z . Le taux de distorsion introduit est déterminé en calculant le pas de quantification, pour chaque sous-bande, afin d'atteindre le débit binaire cible. L'évaluation des performances a démontré l'amélioration du compromis débit/distorsion en utilisant le processus d'allocation binaire. L'étude expérimentale a montré que notre approche conduit à des résultats satisfaisants par rapport à l'état de l'art.

Contents

1	Introduction	7
1.1	Field applications of 3D shapes	7
1.2	Objectives and contributions	9
1.3	Outline	10
2	3D shapes modeling	13
2.1	Introduction	13
2.2	Field applications of 3D shapes	13
2.3	Creation of 3D shapes	14
2.4	3D shape modeling	14
2.4.1	Linear Representations (Polygonal Meshes)	14
2.4.2	Surface representations	21
2.4.3	Volume representations	24
2.4.4	Discrete models	25
2.4.5	Fractal models	25
2.4.6	Constructive models	26
2.5	Conclusion	26
3	Background knowledge on 3D shape intrinsic modeling	27
3.1	Introduction	27
3.2	Geometry modeling	27
3.2.1	Spectral and Laplacian based modeling	28
3.2.2	Conformal geometry based modeling	32
3.2.3	Riemannian geometry based modeling	35
3.3	Topology modeling	38
3.3.1	Curve skeletons	38
3.3.2	Segmentation	41
3.3.3	Differential topology based modeling (Reeb Graph)	44
3.4	Conclusion	47
4	Reeb Graph extraction based on Heat Diffusion	49
4.1	Introduction	49
4.2	Heat diffusion	50
4.2.1	Heat kernel	50
4.2.2	Laplace-Beltrami operator	52
4.2.3	Diffusion distance	53
4.3	Survey on Reeb graph extraction	54
4.4	Proposed method	56
4.4.1	Feature points extraction	56
4.4.2	Reeb graph construction	58

4.5	Experimental results	62
4.5.1	Parameter setting	63
4.5.2	Accuracy assessment	63
4.5.3	Robustness evaluation	67
4.5.4	discussion	67
4.6	Conclusion	68
5	Application to 3D pattern recognition	69
5.1	Introduction	69
5.2	Previous work	70
5.3	3D shape retrieval system	72
5.3.1	Signatures computation	72
5.3.2	Global shape similarity calculation	75
5.4	Experimental results	76
5.4.1	Experimental setup	76
5.4.2	Efficacy evaluation	79
5.4.3	Robustness assessment	79
5.4.4	Comparison with previous methods	80
5.4.5	Computation times	87
5.5	Conclusion	88
6	Application to Segmentation-based compression scheme	91
6.1	Introduction	91
6.2	Survey on 3D dynamic compression	92
6.3	Proposed 3D segmentation-based compression scheme	95
6.3.1	Proposed Segmentation approach	96
6.3.2	Compression scheme	98
6.3.3	Rate control	99
6.4	Experimental results	101
6.4.1	Evaluation criteria	101
6.4.2	Performance results	103
6.5	Conclusion	109
7	Conclusion	111
7.1	Summary of contribution	111
7.2	Open problems and Perspectives	113
.1	Appendix A	115
	Bibliography	119

List of Figures

1.1	The various fields of applications where 3D objects occupies an indispensable role.	8
1.2	Time-varying geometry on the cat sequence.	8
2.1	Triangular mesh illustrating the topological information.	15
2.2	From left to right: an irregular mesh, semi-regular mesh and regular mesh.	16
2.3	17
2.4	A sphere (a) is of <i>genus</i> 0, a torus (b) is of <i>genus</i> 1 and a 2-torus is of <i>genus</i> 2.	18
2.5	Naive representation of a triangular mesh.	18
2.6	Some examples of dynamic 3D meshes.	20
2.7	A parametric surface example.	22
2.8	Example of an implicit surface.	23
2.9	Illustration of the hierarchical aspect of the subdivision.	23
3.1	The Fielder vector gives a natural ordering of the nodes of a graph. The displayed contours show that it naturally follows the shape of the dragon.	30
3.2	Geometric spectrum of simplified Bunny mesh (100 vertices).	31
3.3	Conformal mapping of an original real female face (a) to a square (b). A checker board texture (c) mapped back to the face preserving the right angles on (d) from [Gu 2003].	34
3.4	Conformal parameterization of an original brain mesh (a) in a rectangular conformal parameter domain (b). In (c) the brain is textured using the parameterization obtained from (b). (d) and (e) show the conformal factor and the mean curvature of the brain surface respectively, tired from [Lam 2014].	34
3.5	The top row shows the five input poses of the armadillo shape. The bottom row (left) shows the 2D planar triangulation obtained by adding two refinement steps. The bottom row (right) shows the curve drawn in the exploration phase.	38
3.6	A linear 2-D graph obtained after applied the axe median transformation with the central axis (a). (b) shows the graph of the sensitivity to artifacts.	39
3.7	Main stages of Tierny <i>et al.</i> [Tierny 2006a] proposed enhanced topological skeleton approach. (a) Feature points extraction, (b) Mapping function definition, (c) Reeb graph construction, (d) Constriction approximation and (e) Enhanced skeleton construction.	40

3.8	Example of application: mesh deformation. (a) Enhancing 3D mesh topological skeleton, (b) Its application to deformation.	41
3.9	Example of mesh attributes used for segmentation. (a): Minimum curvature, (b): Average geodesic distance and (c) Shape diameter function.	42
3.10	The Reeb graph of a 3D torus object using the height function.	45
3.11	Example of different scalar function distribution in armadillo object. (a) Height function, (b) Using barycenter function and (c) Using geodesic function.	46
4.1	Feature points detection. Fig (a) shows the two farthest points v_1 and v_2 and Fig (b) shows the two sets of local minima of $f(v, v_1)$ and $f(v, v_2)$ corresponding to v_1 and v_2 respectively	57
4.2	The stability of the scalar function over time on the horse model, red to blue colors express the increasing values of the scalar function.	60
4.3	Feature points extraction in the reference frame of the different sequences.	61
4.4	Kinematic Reeb Graph of different 3D mesh sequences.	61
4.5	Visual comparison between Lavoué <i>et al.</i> [Benhabiles 2012] algorithm (b), Tierny <i>et al.</i> [Tierny 2008a] algorithm (c) and our method (a).	62
4.6	Kinematic Reeb Graph of women 3D sequence with variable connectivity.	64
4.7	Robustness of the feature vertices detection against various transformations.	65
4.8	Robustness of the Reeb graph construction against various transformations.	66
5.1	Unfolding signature computation relative to the area distortion for a Disk-like chart (a) and Annulus-like chart (b) from [Tierny 2007]	74
5.2	Example of stretching signatures for altered versions of primitive charts.	75
5.3	Chart similarity matchings between a query model and the top 4 retrieved objects.	76
5.4	Precision-recall curves for: (A) the whole collection selected from SHREC 2012 data set and (B) each category.	81
5.5	Precision-recall curves of the tested methods for the SHREC 2010 database.	84
5.6	Examples of query objects from the SHREC 2007 query-set and the top-7 retrieved models.	85
5.7	Average Normalized Discounted Cumulated Gain (NDCG) vectors for Our method and Reeb pattern unfolding RPU [Tierny 2009] on the SHREC 2007 data-set.	87
5.8	Precision-recall curves, of our method and the EMD-PPPT algorithm [Agathos 2009] for the McGill database.	88

6.1	Block diagram of the proposed coding scheme.	95
6.2	Match region boundaries with deep surface concavities.	96
6.3	Rate/distortion characteristic using Lagrangian optimization.	102
6.4	Segmentation results on four selected frames, extracted from (a-b-c-d) Dance, and (e-f-g-h) Snake sequences.	103
6.5	Rate/distortion performances for Cow (a), Chicken (b) and Dance (c) sequences.	106
6.6	Rate/distortion performances for Chicken (a), Snake (b) and Dance(c) sequences.	107
6.7	RMSE as a function of the frame index for cow and snake sequences.	108
6.8	Key-frames extracted from the Cow sequence, (a) frame decoded at $2bpvf$, (b) frame decoded at $3.5bpvf$, (c) frame decoded at $4.7bpvf$, (d) frame decoded at $5.5bpvf$, (d) frame decoded at $6.8bpvf$	108
1	Probability density functions of the predicted coefficient coordinates for the Cow model: (a) X-coordinates, (b) Y-coordinates, (c) Z-coordinates. The blue and red curves represent the real distribution and the approximated one respectively.	117

List of Tables

2.1	A Comparison between 3D shape modeling techniques.	25
4.1	Properties and computation time of the tested dynamic meshes.	63
4.2	Computation times for the tested dynamic meshes.	64
5.1	Retrieval performance of our method evaluated using five standard measures on the whole collection selected from SHREC 2012 data set and for each category.	80
5.2	(I) Screen shots of the null model (a) after different transformations: random noise (b), isometry (c), affine transformation (d), partiality (e), sampling (f), scale (g), topology (h). (II) Similarity results of the matching experiments.	82
5.3	Comparison of similarity estimation scores on the SHREC 2011 data set obtained by our method and the method proposed in [Barra 2013].	83
5.4	Similarity estimation scores on the SHREC 2010 data set.	84
5.5	Retrieval performance for the McGill database.	86
5.6	Average execution times searching 3D models in McGill, SHREC 2010 and SHREC 2012 data sets.	88
6.1	Properties of the tested dynamic meshes.	102
6.2	Evaluation of the mean square error of the motion compensation $E(\Pi)$.	104
6.3	Average execution times for Cow, Dance and Chicken models.	109
1	Chi-Square distribution table.	116
2	χ^2 test for Cow model, with $k \in 6, 16, 26$	116

Introduction

Contents

1.1	Field applications of 3D shapes	7
1.2	Objectives and contributions	9
1.3	Outline	10

PICTURE concept did not starts only with the advent of first computer, camera or scanner. The language of the image is reproduced since ancient times, with the beginning of this life. Where humans, in a long time ago, communicate among themselves via sign language and graphics. Up to now archaeologists are trying to decode their manuscripts to learn the secrets of the various people lives. But with the invention of the computer, the scanner or any image capture equipments, it has become necessary to look at ways to analyze and process this kind of digital data.

1.1 Field applications of 3D shapes

In the last decade, the technological progress in telecommunication, hardware design and multimedia, allows access to an ever finer three-dimensional (3-D) modeling of the world. Nowadays, this kind of 3D contents is commonly used in several domain applications (see Fig.1.1) including digital entertainment and scientific simulation. The critical challenges with 3D models lie in their visualization, rendering, protection or transmission over channels with limited bandwidth and storage on media with low capacity.

In order to ensure interoperability exchanges and the interpretation of these particular data, 3D objects must be represented according to standard formats. There exists many 3-D representations such as implicit surface, NURBS or voxel. But the most widely used representation of 3D shapes is the triangular surface mesh. This representation, consisting of vertices, edges and faces, is very widespread due to its simplicity. It contains geometrical information representing vertex coordinates in 3D space and topological information describing the incidence and adjacency relationship between vertices. In addition to its algebraic simplicity and high usability, 3D mesh representation is considered as an effective low-level model. Indeed, any kind of 3D models can be easily converted to 3D mesh representation.



Figure 1.1: The various fields of applications where 3D objects occupies an indispensable role.

While most researchers have focused on the field of 3D objects, now it is necessary to turn to 3D time domain (3D+t). 3D dynamic meshes are becoming a media of increasing importance. A 3D dynamic shape is usually represented by a sequence of 3D meshes with constant connectivity and temporal information provided by time-varying geometry, only the vertex positions changes over time (see Fig. 1.2).

Similar to pixel grid representation, this 3D content is subject to various processing operations such as indexation, segmentation or compression. However, surface mesh is an extrinsic shape representation. Therefore, it suffers from important variability under different sampling strategies and canonical shape-non-altering surface transformations, such as affine or isometric transformations. Consequently it needs an intrinsic structural descriptor before being processed by one of the aforementioned processing operations.

To meet these challenges, in this thesis, we focus to the intrinsic topological modeling based on Reeb graph and we intend to extend this principle for dynamic models.

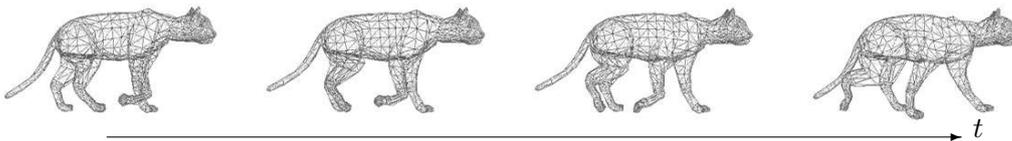


Figure 1.2: Time-varying geometry on the cat sequence.

1.2 Objectives and contributions

The research topic of this thesis work is the topological modeling based on Reeb graphs. Specifically, we focus on 3D shapes represented by triangulated surfaces. Our objective is to propose a new approach, of Reeb graph construction, which exploits the temporal information. The main contribution consists in defining a new continuous function based on the heat diffusion properties. The latter is computed from the discrete representation of the shape to obtain a topological structure.

The restriction of the heat kernel to temporal domain makes the proposed function intrinsic and stable against transformation. Due to the presence of neighborhood information in the heat kernel, the proposed Reeb Graph construction approach can be extremely useful as local shape descriptor for non-rigid shape retrieval. It can also be introduced into a segmentation-based dynamic compression scheme in order to infer the functional parts of a 3D shape by decomposing it into parts of uniform motion. In this context, we apply the concept of Reeb graph in two widely used applications which are pattern recognition and compression.

Application to pattern recognition

Reeb graph has been known as an interesting candidate for 3D shape intrinsic structural representation. we propose a 3D non rigid shape recognition approach. The main contribution consists in defining a new scalar function to construct the Reeb graph. This function is computed based on the diffusion distance. For matching purpose, the constructed Reeb graph is segmented into Reeb charts, which are associated with a couple of geometrical signatures. The matching between two Reeb charts is performed based on the distances between their corresponding signatures. As a result, the global similarity is estimated based on the minimum distance between Reeb chart pairs.

Application to segmentation-based dynamic Compression

Skeletonisation and segmentation tasks are closely related. Mesh segmentation can be formulated as graph clustering. First we propose an implicit segmentation method which consists in partitioning mesh sequences, with constant connectivity, based on the Reeb graph construction method. Regions are separated according to the values of the proposed continuous function while adding a refinement step based on curvature and boundary information.

Intrinsic mesh surface segmentation has been studied in the field of computer vision, especially for compression and simplification purposes. Therefore we present a segmentation-based compression scheme for animated sequences of meshes with constant connectivity. The proposed method exploits the temporal coherence of the geometry component by using the heat diffusion properties

during the segmentation process. The motion of the resulting regions is accurately described by 3D affine transforms. These transforms are computed at the first frame to match the subsequent ones. In order to improve the performance of our coding scheme, the quantization of temporal prediction errors is optimized by using a bit allocation procedure. The objective aimed at is to control the compression rate while minimizing the reconstruction error.

1.3 Outline

The remainder of this manuscript is laid out as follows :

Chapter 2 presents a classification of different 3D object representations focusing on triangular 3D surface models. the fields and the area applications of 3D object are represented first. Then it reviews the different representations of three-dimensional objects regrouped in three main categories: surface models, volume models and linear models, specifically polygonal meshes while focusing on 3D shapes represented by surface meshes.

Chapter 3 introduces the different modeling 3D meshes specifying the benefits of using the topological modeling based on Reeb graphs. The final part defines the differential topology modeling by introducing the morse theory notion and giving a survey on Reeb Graph extraction methods.

Chapter 4 proposes a new Reeb graph construction approach which exploits the temporal information. The main contribution consists in defining a new scalar function. In this chapter, we introduce the heat diffusion principle, adapted to Riemannian manifolds, which is the core of the proposed scalar function. Then we describe our Reeb Graph construction method in detail. Finally we investigate the performance of our approach in terms of accuracy and robustness.

Chapter 5 presents a 3D non rigid shape recognition approach that uses the Reeb graph representation as local shape descriptor. We start by providing a brief overview of the most relevant work in the field of 3D pattern recognition. Then, we describe the proposed approach in detail. Finally, we evaluate the performance of our system by conducting a fair comparison with previous approaches from the state-of-the-art.

Chapter 6 proposes another application of Reeb graph representation in the context of dynamic mesh partitioning. First, we provide an overview of the various

existing work in the field of segmentation and compression of 3D mesh sequences. Second, we describe the proposed segmentation-based dynamic compression scheme. In order to examine the effectiveness of our compression system, we report the compression results and compare them to other 3D dynamic coding techniques from the state-of-the-art.

Finally, Chapter 7 concludes this manuscript. It provides a summary of contributions and presents directions of future work and open problems.

3D shapes modeling

Contents

2.1	Introduction	13
2.2	Field applications of 3D shapes	13
2.3	Creation of 3D shapes	14
2.4	3D shape modeling	14
2.4.1	Linear Representations (Polygonal Meshes)	14
2.4.2	Surface representations	21
2.4.3	Volume representations	24
2.4.4	Discrete models	25
2.4.5	Fractal models	25
2.4.6	Constructive models	26
2.5	Conclusion	26

2.1 Introduction

3D objects are commonly used in several domain applications, in this chapter we highlight the fields where 3D modeling is considered as an important issue. In addition to the area applications of these data, we are also interested to their hardware and software generation which is addressed in the second part of this chapter. After being created, 3D objects are modeled according to standard formats, in order to ensure their interoperability exchanges and their interpretation. In the last part of this chapter, we review the modeling of three-dimensional objects. We distinguish three main categories: surface models, volume models and linear models, specifically polygonal meshes. We focus in this thesis on 3D shapes represented by surface meshes.

2.2 Field applications of 3D shapes

The recent technological progresses in the fields of telecommunication, computer graphics and multimedia allow access to an ever finer three Dimensional modeling of the world. 3D shape modeling occupies a very important place in the computer graphics world. It is used in areas as diverse as medicine, video

games, computer-aided design... It is very interesting to 3D objects whenever we want to make virtual tours of museums or to model real and/or virtual 3D scenes. Today with the technological advances in the medical field, 3D objects are integrated in computer-aided diagnosis through CT (Computerized Tomography) and MRI (Magnetic Resonance Imaging) scans. Furthermore, they are used in computer-assisted surgeries. Furthermore, it is worth mentioning that 3D objects play an unavoidable role in geographic information systems such as astronomy, geology, and mapping.

Obviously, we cannot evoke all of the application areas. However, it is important to cite the studies and simulations of physical phenomena around us. This area is based on the numerical simulation by using finite element analysis methods and by solving differential equations. By this way, it is possible to study the propagation of electromagnetic waves through the human body, and consequently evaluate their dangerousness.

In what follows, before we turn to the modeling of 3D objects, we briefly back on methods of creating the underlying 3D models.

2.3 Creation of 3D shapes

There are specialized 3D CAD (Computer-Aided Design) software (AutoCAD, Autodesk Maya, Autodesk 3ds autodesk inventor ...) and geometric modelers that are generally used to obtain a geometric and topological representation of virtual 3D object or scene.

The representation of a real 3D object can be obtained by using special hardware devices called *range scanners*. The scanning devices can produce data (range images or point clouds) which is very dense without necessarily reflecting the curvature of the object. Indeed, these devices produce a highly redundancy data, especially on smooth areas of the mesh, which is difficult to process. To alleviate this problem it is possible to reduce the redundancy through simplification methods.

After the data acquisition phase, several modeling types can be used to represent these 3D-data in order to ensure their interoperability exchanges and interpretation. Among these 3D modeling types, we can cite: surface representation, volume representation, and linear representation.

2.4 3D shape modeling

2.4.1 Linear Representations (Polygonal Meshes)

Linear models are widely used thank's to their simplicity. They are characterized by a very understood modeling ability, which allows them to represent any complex

topology object. Among the linear representation, we distinguish polygonal meshes, which are represented by a set of vertices connected by edges forming facets. The most commonly used geometric forms to represent these facets are triangles (3-D triangular meshes) that will be used in the context of our studies. 3D meshes belong into the surface modeling class that provides roughly a representation of an object, which is very complex and adapted to the shape design. This kind of representation is defined by a geometric information represented by the vertex coordinates in 3D space and topological information describing the incidence and adjacency relationships between vertices, edges and faces. The topological information includes the degree of a face which means the number of edges which it compose (in the case of triangular mesh, the degree of faces is equal to 3) and the vertex valence, which is the number of its incident edges. These two pieces of information are explained in Fig. 2.1.

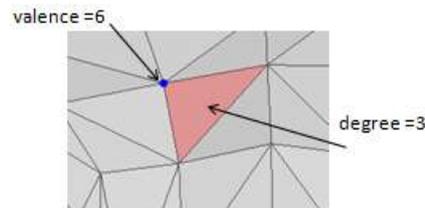


Figure 2.1: Triangular mesh illustrating the topological information.

2.4.1.1 Topological properties of 3D surfaces

The key concept when studying the topological properties of surfaces, is the notion of homeomorphic topological spaces. Properties of figures unchanged by homeomorphisms are called topological properties, or topological invariants.

Homeomorphism: Two 3D topological surfaces S and S' are homeomorphic only if there is a continuous bijection φ between the two surfaces $\varphi: S \rightarrow S'$ such as the inverse function φ^{-1} is also continuous. Intuitively, S and S' are called homeomorphic if the surface S can be stretched and bent without breaking to fit the shape of S' . The notion of homeomorphism allows defining equivalence classes in the surface spaces. In particular, it allows introducing the varieties, defined as follows: A triangle mesh can be 2-manifold if it satisfies the following properties:

- Property local disk: if there is on each point of the surface a neighborhood homeomorphic to an open disk or an open semi-disk.
- Property scheduling edges: the adjacent edges of each vertex must be arranged in a circular fashion.
- Neighborhood Property face: each edge of the mesh must have exactly two adjacent faces if it is an inside edge to the mesh and only one face if it is an

boundary edge.

To test whether a mesh is a manifold or not, it is important to introduce the concepts of regular vertex and regular edge.

- **Regular vertex**: all its neighbors can be rearranged to define a unique path.
- **Regular edge**: it is shared by a maximum of two triangles.

According to the aforementioned definitions, we can demonstrate the following property: a triangular mesh is manifold only if all its vertices and edges are regular. A mesh is called non-manifold if it has at least one edge connected with at least three sides, so it will be impossible to differentiate the inside and the outside without ambiguity.

Neighborhood regularity, depends on the valence of the vertices, is also a very important property for triangular meshes. It depends on the valence of the vertices, As illustrated in Fig.2.2, we distinguish three mesh structures:

- Irregular mesh: all vertices have different valence values due to the the lack of consistency in how to connect the vertices
- Regular mesh: all vertices have the same valence.
- Semi-regular mesh: a small number of vertices are irregular and the remains have the same valence.

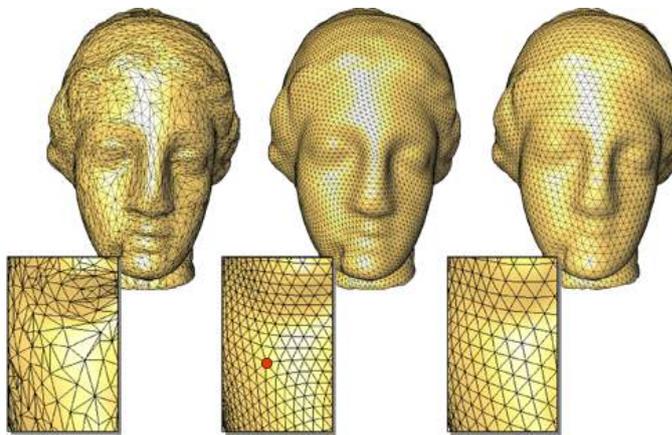


Figure 2.2: From left to right: an irregular mesh, semi-regular mesh and regular mesh.

It is also possible to distinguish other types of meshes such as:

- Conform mesh: it has all geometric elements of non zero areas and the intersection of two geometric elements of the mesh is either empty or reduced to a vertex or an entire edge. Connecting the middle of a ridge and a summit will, for example prohibited.

- Multi-resolution mesh: it offers several levels of information and good support for progressive rendering, scalable compression, and data transmission. The aim is to represent the surface at different levels of detail. The decomposition process of the original mesh into intermediate meshes is reversible. From the coarser mesh it is possible to reconstruct all levels of approximations until reaching the fine one (*coarse to fine*). Or, conversely, simplifying a fine mesh to obtain a coarser approximation (*fine to coarse*). Fig. 2.3 (a) and (b) show the simplification and the reconstruction stages. It is important to note that the hierarchical decomposition techniques depend on the mesh connectivity constraints, while the simplification approaches are applicable on any mesh connectivity.

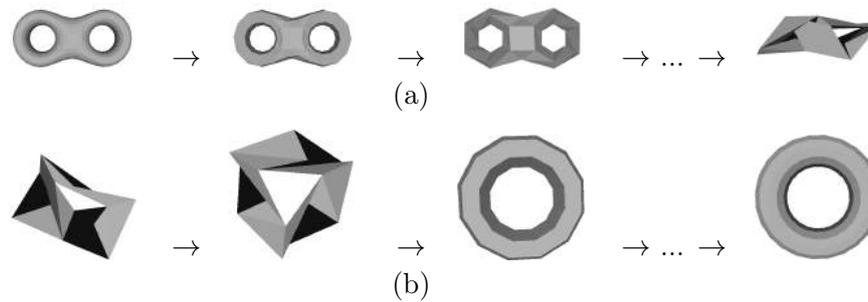


Figure 2.3:

Some triangular meshes respect the *Delaunay criterion*. In this case, the circumscribed circles of triangles forming the mesh are do not contain any vertex.

Euler's characteristic: Let M be a manifold mesh, oriented and without board, composed of F triangles, E edges and V vertices. Let G be the *genus* of the mesh M , which corresponds to the maximum number of closed curves without common points that can be drawn inside this surface without disconnecting it. The Euler's formula [Coxeter 1989] is given by:

$$\chi = V - E + F. \quad (2.1)$$

This Euler's characteristic χ is related to the *genus* G of the surface. Indeed, the *genus* is a global topological feature that allows to determine equivalence classes in the varieties of space. It reflects more or less its topological complexity and is intuitively equal to the number of handles in the shape (see Fig.2.4). More specifically, the *genus* G of a 3-D object can be expressed by the following equation:

$$G = \frac{2c - b - \chi}{2}, \quad (2.2)$$

where c is the number of connected components and b is the number of edges of the surface. In practice, a small number of meshes satisfies the regularity property.

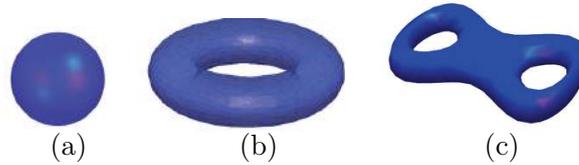


Figure 2.4: A sphere (a) is of *genus* 0, a torus (b) is of *genus* 1 and a 2-torus is of *genus* 2.

However, under certain assumptions, we can demonstrate that the average valence of the vertices is 6. This result is a direct consequence of the Euler's characteristic.

The orientation of a face is defined according to the cyclic order of vertices and the right-hand rule. There are two possibilities: the orientations of two adjacent faces are *compatible* if there exist two vertices shared across commands in both sides. So the complete mesh is called *orientable* if we can find a combination of orientations in all sides such that each pair of adjacent faces in the mesh is compatible.

2.4.1.2 Standard formats of representation

Various standard formats use the naive representation of polygonal meshes. Most of these file formats are represented in an ASCII form such as the Virtual Reality Modeling Language (VRML), the 3D Object File Format OFF, the Wavefront Object format OBJ, the Stanford University PoLYgon format PLY, The storage strategies of these file formats are very similar. The geometry is generally represented by an indexed list of vertex coordinates and the connectivity is composed of a list of faces, where each face is represented by the indices of its vertices. The global file consists of the geometrical information followed by the topological one as shown in Fig.2.5 The principle is to encode the mesh geometry by using a matrix G

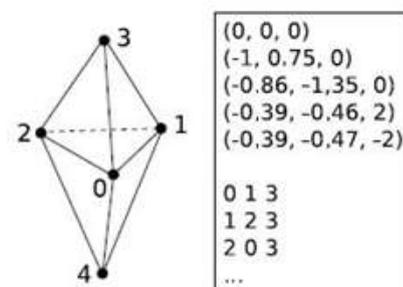


Figure 2.5: Naive representation of a triangular mesh.

with V rows and 3 columns, with V being the number of vertices:

$$G = \begin{pmatrix} X_1^x & X_1^y & X_1^z \\ X_2^x & X_2^y & X_2^z \\ X_3^x & X_3^y & X_3^z \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ X_V^x & X_V^y & X_V^z \end{pmatrix} \quad (2.3)$$

where X_l^x , X_l^y and X_l^z are the cartesian coordinates of the vertex indexed by l in the surface mesh M . The mesh connectivity is also represented by a matrix denoted by Γ of size $F \times 3$ (where F is the number of faces).

$$\Gamma = \begin{pmatrix} v_1^1 & v_1^2 & v_1^3 \\ v_2^1 & v_2^2 & v_2^3 \\ v_3^1 & v_3^2 & v_3^3 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ v_V^1 & v_V^2 & v_V^3 \end{pmatrix} \quad (2.4)$$

where v_i^1 , v_i^2 and v_i^3 are the integer indices of three vertices forming the i^{th} triangle of M .

2.4.1.3 From 3D to 3D+t domain

Technological progress in the field of multimedia and computer vision has led to the exploitation of the time factor t to process 3D objects. While the majority of research in this area was based on 3D objects, now, it is necessary to turn to 3D time domain (3D+t). Indeed, dynamic3D shapes are becoming a media of increasing importance used mainly in the field of video games, movies, computer-aided design, and medical imaging. This kind of data is usually represented by key-frame sequences of 3D triangular meshes sharing the same connectivity and temporal information provided by time-varying geometry. Only the vertices position changes over time. As for static models, dynamic models can be formalized mathematically as follows: Let's designate by $(M_t)_{t \in \{1, \dots, T\}}$ a sequence of 3D meshes (where T is the number of frames). Under the hypothesis of a fixed connectivity, by considering Γ (given by eq. 5.3), the mesh geometry at time t is represented by a matrix G_t of dimension $3 \times V$ (where V is the number of vertices) defined by:

$$G_t = \begin{pmatrix} X_{1,x}^t & X_{1,y}^t & X_{1,z}^t \\ X_{2,x}^t & X_{2,y}^t & X_{2,z}^t \\ X_{3,x}^t & X_{3,y}^t & X_{3,z}^t \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ X_{V,x}^t & X_{V,y}^t & X_{V,z}^t \end{pmatrix} \quad (2.5)$$

where $X_{l,x}^t$, $X_{l,y}^t$ and $X_{l,z}^t$ are the cartesian coordinates of the vertex indexed by l at time t M_t . Fig.2.6 shows some key-frame sequences of 3D triangular meshes. Animate a 3D object consists to describe the motion and/or the deformation that it

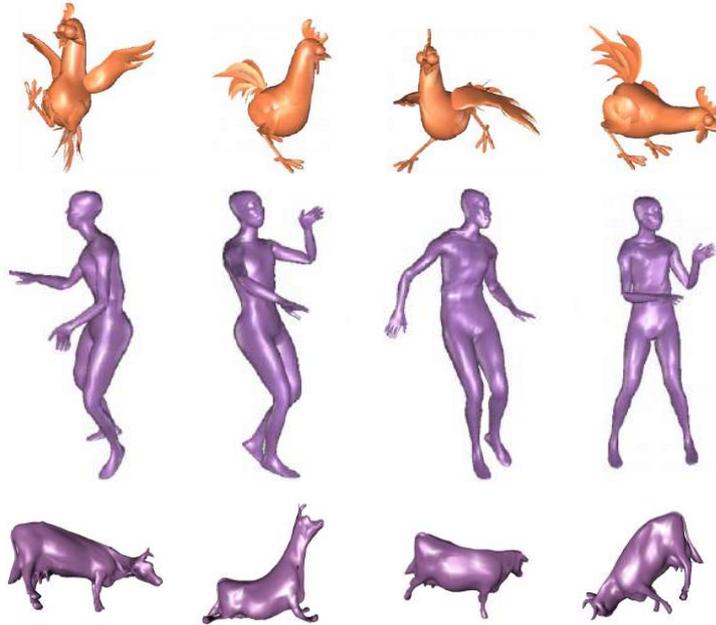


Figure 2.6: Some examples of dynamic 3D meshes.

undergoes during a specified time period. Most often this amount of data, needed to generate a dynamic 3D object represented by key-frame sequences, describes the time evolution of a 3D surface (i.e change of the vertex positions, normals, colors ...). The first approach that has been adopted to generate animated content specifies the properties of the 3D object as a function of time. Obviously, such approach (heavy and non-intuitive) is not usable in practice, even in the case of simple 3D models. To simplify the task of animated content generation, the majority of animation techniques proposed to describe the animation operator according to the motion patterns and / or deformation.

In general, the creators of 3D animated objects can be classified into two main categories: animation using descriptive models and procedural animation.

The first category is based on an explicit representation of the animation that describes, for each key frame, the motion field parameters or associated deformation. This type of animated 3D objects creation allows designers to accurately control the progress of the animation. However, it requires a significant volume of user interaction for the specification of key-frames. On the other hand, the second category is primarily based on a set of physical, mathematical or behavioral laws. It generates dynamically and automatically realistic animations and high quality while taking into account the interaction with user or changes in the environment. The disadvantage is that the control of the time flow of the animation is limited. To store these animated models, there are various standards of representation formats such as:

- The standard VRML *Virtual Reality Modeling Language* (WRL file extensions), developed by the Web3D Consortium is a description language for interactive 3D virtual universe. It represents a 3D scene as a hierarchical tree whose nodes describe objects or scene properties (3D meshes, basic shapes, sounds, light sources, colors ...).
- The standard X3D *eXtensible 3D* extends the VRML standard by introducing new features and a description format. This representation allows to describe the animated humanoid, physical interactions between solids, and particle systems necessary for modeling elements such as fire, smoke, snow ...
- The H-Anim standard is another description language for character animations articulated human model. H-Anim representation allows modeling the anatomical skeleton of a 3D articulated character by a hierarchical tree structure.

2.4.2 Surface representations

Surface models are composed of k -simplices which may be the vertices (0-simplex), the edges (1-simplex) or triangles (2-simplices). The polygonal mesh belongs to this type of modeling, the object is represented by several polygonal elements and the surface will be built by assembling its elements. This representation model is classified into three types of surfaces: parametric, implicit, and subdivision surfaces.

2.4.2.1 Parametric surfaces

The parametric representation is characterized by the definition of each surface point by a equation with two parameters η and μ represents the application of a region of the plane (η, μ) in three dimensional space. Fig. 2.7 shown an example of a parametric surface.

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

It is preferable that the functions f_x , f_y and f_z are polynomial functions to obtain an accurate approximation of the surface and a simple geometric interpretation of their coefficients. This type of surfaces is obviously used to interpolate or approach a set of points. These points are usually organized in a matrix form. These para-

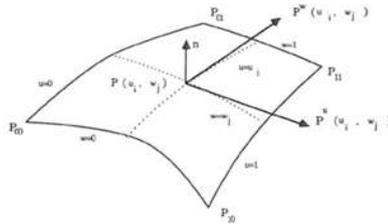


Figure 2.7: A parametric surface example.

metric models include a large family of surfaces. We can distinguish primarily the sub family of curves and Bezier surfaces (area B-Splines / NURBS) that are characterized by a set of points called control points forming a grid. The disadvantage lies in the movement of a control point which affects the entire object.

2.4.2.2 Implicit surfaces

Contrary to parametric models that explain the point coordinates, the implicit formalism is defined according to a particular mathematical form [Bloomenthal 1997]. It consists at representing a surface as a set of points in space checking a property which is generally related to the value taken at these points.

An implicit surface S is defined as the set of zeros of a function f in \mathbb{R}^3 in \mathbb{R} . The set of points $P = (x, y, z)$ of the implicit surface S defined by f is one that satisfies the following equation:

$$f(x, y, z) = 0. \quad (2.6)$$

From this formulation and using the sign of the function we can directly concluded the information about the relationship between all points in three-dimensional space:

- if $f(x, y, z) < 0$; p will be on the outside of the object to be modeled.
- if $f(x, y, z) > 0$; p will be inside the object to be modeled.
- if $f(x, y, z) = 0$; p will be on the surface of the object to be modeled.

The advantage is to separate the space into two components: inside and outside the area. So one can easily determine the position of a point relative to the boundary

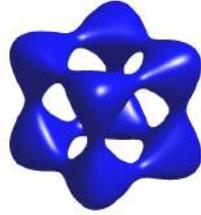


Figure 2.8: Example of an implicit surface.

surface. This representation model can be considered as the surface model or volume model because otherwise described the volume defined by the surface. This type of representation allows modeling of rounded shapes. It is best suited for medical imaging, physical processes [Terzopoulos 1987], human modeling and modeling smooth objects [Turk 1999].

Implicit surfaces are divided into two categories: algebraic surface that is mathematically defined as the set of roots of a polynomial function more or less degree of complex (2, 3 or 4). Non algebraic surface that serves to model an object by a set of particles. Fig. 2.8 shows an example of implicit algebraic surface defined by the equation: $x^4 - 5x^2 + y^4 - 5y^2 + z^4 + 5z^2 + 11.8 = 0$.

2.4.2.3 Subdivision surfaces

A subdivision surface [D. Zorin 2000] is a smooth surface defined as the limit of a sequence of refinements, applied to a control mesh. Fig.2.9 describes the hierarchical aspect of the subdivision. These refinements include modifying connectivity and

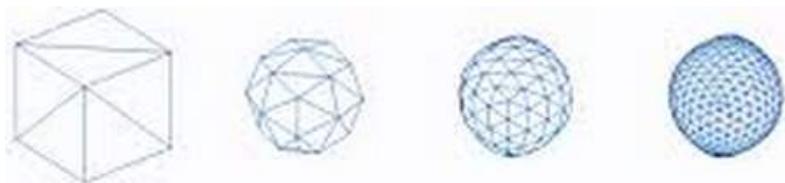


Figure 2.9: Illustration of the hierarchical aspect of the subdivision.

geometry by adding, moving vertices to obtain a mesh that tends toward a smooth boundary.

In general, a subdivision scheme is described by:

- A topological component: all subdivision schemes are changing the initial mesh connectivity, then we can distinguish two types of schemes: primal patterns that retain the old highs, and dual patterns that suppress.

- A geometric component: the vertices position change can be interpreted as a smoothing of the original mesh. One can also distinguish two types of schemes: the interpolating patterns that keep the position of initial vertices and non-interpolating schemes that change their positions by moving them.

We distinguish several subdivision schemes differentiated according to the type of polygons treated and the type of operation performed subdivision. We quote as well the nature of schemes approximating where the control points are not located on the boundary surface. It is difficult to estimate the resulting surface. Meanwhile, we find the nature of interpolating subdivision schemes in which all control points lie on the boundary surface, since the movement includes only the newly inserted vertices.

2.4.3 Volume representations

3D volume representations are particularly suited for medical imaging (3D Volume Representation of tumor through the use of Magnetic Resonant Imaging) ref(3D Volume Representation of brain tumor using image processing). 3D volumetric medical images are usually analyzed as a sequence of 2D image slices [Shen 2008] due to concerns over the exponential increase in computational cost in 3D. These kind of representations allow structural modeling objects by one or several primitives of volume nature, generally ordered in graph form. There are different types of primitives such as cylinders, superquadrics the hyperquadrics and other implicit polynomial. Thus we can classify these models into two groups:

- Quantitative models having a great modeling power.
- Qualitative models for symbolic modeling.

2.4.3.1 Superquadrics

The superquadric model is an extension of quadric, this primitive has the capability to admit an implicit and a parametric forms, the most commonly used are the super-ellipsoid. The high description capability is one of the advantages of this model despite the small number of parameters. These models are well suited to the field of medical imaging. They provide an efficient modeling, in both space and time, of certain organs such as the heart.

2.4.3.2 Hyperquadrics

The hyperquadric model is a general case of superquadric, but it only makes an implicit representation compared to superquadric model. It differs from superquadric model by the non symmetry of its representation and its descriptive power. Despite the high description power of hyperquadric model, its use remains marginal compared to that of superquadric. As a result, the complexity and the lack of parametric formulations of hyperquadric primitives make them less reliable. Similarly to

superquadrics, these primitives are used mainly for the reconstruction and modeling of 3D objects in the medical field.

Table 2.1: A Comparison between 3D shape modeling techniques.

Model	advantages	disadvantages
3D Mesh	algebraic simplicity high usability arbitrary topology	lack of continuity scale dependence
Parametric surfaces	high continuity mathematically defined compactness local control	no arbitrary topology complex handling
Implicit surfaces	compactness	complex handling descriptive ability limited to organic forms complexity sampling
Subdivision surfaces	high continuity arbitrary topology algebraic simplicity compactness multiresolution local control	no defined mathematically
Volume Representations	compactness	limited descriptive ability
Discrete Model	high usability	scale dependence size memory
fractal Model		descriptive ability restricted to natural objects
Constructive Model		complex rendering

2.4.4 Discrete models

By using a discrete model, an object is represented by the set of spatial cells occupied by the volume of the object in space. This representation is obtained using a three-dimensional array consisting of fixed-size cubes called voxels. The discrete models are very simple however, they are very expensive in terms of memory, and they are often used in the medical field.

2.4.5 Fractal models

The objective is to represent a curve or an irregular shaped surface by an iterative method. This kind of representation was used for 2D image compression and has

been extended to 3D object compression. It is used only to represent natural objects such as mountains and clouds... It can represent repeated patterns many times. These are usually recursive functions using an initial pattern and a replacement pattern. For surfaces, the objective is to divide each segment in half, from an initial triangle, and change the height of the midpoint of each segment randomly

2.4.6 Constructive models

These models are widely used in computer-aided design (CAD) applications. They represent an object by a tree called *build tree* whose leaves are the objects and the non-terminal nodes are considered as operators.

2.5 Conclusion

Table 5.1 summarizes the main advantages and drawbacks of each model representation categories described in this chapter. For a larger survey of 3D surface representations, the interested reader should refer to additional reference on polygonal meshes and their applications in geometry processing [Botsch 2007]. In the context of our work, we focus on the 3D triangular surface meshes, which have a fairly wide descriptive power allows them to manipulate in a simple way the objects of arbitrary topology. However this representation is extrinsic, it suffers from high sensibility against affine and isometric transformations. Therefore to overcome this problem it seems necessary to look for defining computational intrinsic modeling which will be addressed in the next chapter.

Background knowledge on 3D shape intrinsic modeling

Contents

3.1	Introduction	27
3.2	Geometry modeling	27
3.2.1	Spectral and Laplacian based modeling	28
3.2.2	Conformal geometry based modeling	32
3.2.3	Riemannian geometry based modeling	35
3.3	Topology modeling	38
3.3.1	Curve skeletons	38
3.3.2	Segmentation	41
3.3.3	Differential topology based modeling (Reeb Graph)	44
3.4	Conclusion	47

3.1 Introduction

In the previous chapter, we mentioned that the 3D triangular meshes are frequently used to represent 3D objects, thanks to their algebraic simplicity and high usability. However, their only downside lies in the fact that a 3D triangular mesh is an extrinsic modeling, and any applied topological, affine or isometric transformation may affect this representation. For this reason, we need to go through an intrinsic modeling before processing this kind of 3D data. In this chapter, we review the intrinsic modeling of three-dimensional objects. In particular we distinguish two categories: geometry and topology based modeling. For each category, we describe three representative classes of approaches. Finally, we give some theoretical preliminaries and existing work about Reeb graph based modeling which is the core of our research.

3.2 Geometry modeling

The surface geometry is often referred to as its shape. It is primarily defined by the set of its intrinsic characteristics varying under smooth transformations. In the

following, we present three classes of geometry based modeling methods for surface mesh intrinsic description.

3.2.1 Spectral and Laplacian based modeling

Before beginning this section, let us answer the question: what defines spectral modeling? If we suppose a closed system of basic equations and introduce into this system a finite expansion of dependent variables by means of functions such as *Fourier*. Thus we obtain, for these function coefficients, series of coupled non linear differential equations, due to the orthogonality properties of the used spatial functions. By using the Fourier transform, the horizontal spatial dependence is removed. These function coefficients depend only on the time and the vertical coordinate. To solve the coupled non linear differential equations, a simple time-differencing and a vertical finite differencing are mostly applied.

Spectral modeling can be considered as spectral modeling synthesis, noted SMS, which is an acoustic modeling technique adapted to any signals including speech. It allows to replace the portions of the time-domain signal by their short-time Fourier transforms. This principle ensures that the sound representation is very similar to the perception of sound by the brain. This allows to reduce the calculation complexity based on perceptual modeling, and more fundamental data structures perception. Thank's to the short-time Fourier transforms, the famous MP3 audio compression format can reach an order of magnitude information reduction with little or no loss. That is also due to the fact that it prioritizes the conserved data in each spectral frame based on psychoacoustic principles.

In the case of manifolds, various existing work used the spectral transform by putting the given surface into one-to-one correspondence with a simpler domain [Zhou 2004], or to segment the surface into a set of simpler domains [Lee 1998, Pauly 2001]. Therefore, it is possible to define a frequency space in these simpler domains. Authors in [Sokrine 2005] proposed calculating *geometry aware* basis functions, defined as solutions of some least-squares problems.

3.2.1.1 Spectral mesh processing

Spectral mesh processing implies the use of eigenvalues, eigenvectors, or eigenspace projections from suitably defined mesh operators to perform appropriate tasks. The basic idea consists in constructing a matrix, based on the topological and/or geometrical information of the input mesh. This matrix representing a discrete linear operator can be considered as an incorporating pairwise incidence or adjacency relationships between vertices, edges and faces in the mesh. Once the matrix is constructed, an eigen-decomposition is then performed by computing the set of it eigenvalues and eigenvectors. Based on the resulting structures from the decomposition, which is used in a problem specific manner, the solution is obtained.

The primary motivation for proposing spectral mesh processing approaches is the pursuit of *Fourier analysis* in the manifold setting. Methods applied in the spectral domain, project the signal in a transformed space. They propose concepts specially adapted to the underlying irregular three-dimensional meshes. The objective is to infer intrinsic geometrical surface characteristics by computing its spectral transform.

Fourier analysis

In order to define the concept of the Fourier transform, we begin by introducing the case of a closed curve in the continuous setting. Supposing a square integrable periodic function notes $f : x \in [0, 1] \mapsto f(x)$, with f a function defined on a closed curve parameterized by normalized arc-length [Levy 2006]. This function, f , is decomposed into an infinite series of sinus and cosine of increasing frequencies:

$$f(x) = \sum_{k=0}^{\infty} \bar{f}_k H^k(x); \begin{cases} H^0 & = 1 \\ H^{2k+1} & = \cos(2k\pi x) \\ H^{2k+2} & = \sin(2k\pi x) \end{cases} \quad (3.1)$$

being \bar{f}_k the decomposition coefficients calculated according to equation 4.4, the set of these coefficients are called the Fourier Transform (FT) coefficients of the function f .

$$\bar{f}_k = \langle f, H^k \rangle = \int_0^1 f(x) H^k(x) dx, \quad (3.2)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product (i.e. the dot product for functions defined on in interval of $[0, 1]$).

The study of a periodic function by Fourier series has two components: analysis and synthesis. During the analysis, the Fourier coefficients are determined. The synthesis allows to reconstruct the function f using the resulting coefficients \bar{f}_k by applying the inverse Fourier Transform FT^{-1} .

Now let's generalizing these notions to arbitrary manifolds. We suppose the function H^k of the Fourier basis is the eigenfunctions of $\partial^2/\partial x^2$:

$$-\frac{\partial^2 H^{2k+1}(x)}{\partial x^2} = (2k\pi)^2 \cos(2k\pi x) = (2k\pi)^2 H^{2k+1}(x). \quad (3.3)$$

The eigenfunctions H^{2k+1} are associated with the eigenvalues $(2k\pi)^2$. To understand the geometric significance of the eigenfunction, in the next section, we study the discrete setting by considering the eigenfunctions as orthogonal non-distorting 1D parametrization of the shape. In the next sections, we focus on the Laplacian operator in the discrete and continuous settings and present its utility for 3D shape modeling.

Graph Laplacian: discrete setting

Among the early work in this field, we cited the original method proposed by Taubin [Taubin 1995b]. He demonstrated that the signal processing formalism could be applied correctly to geometry processing. The similarity between the eigenvectors of the graph Laplacian and the basis functions used in the discrete Fourier transform is the base of the proposed method in [Taubin 1995b]. The used Fourier function basis allows decomposing a given signal into a sum of sine waves of increasing frequencies.

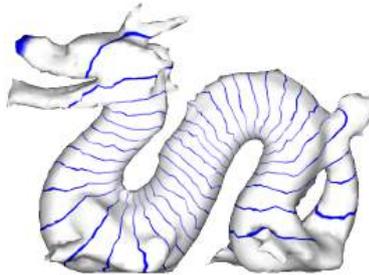


Figure 3.1: The Fielder vector gives a natural ordering of the nodes of a graph. The displayed contours show that it naturally follows the shape of the dragon.

Authors in [Isenburg 2009] have employed the spectral graph theory to calculate an ordering of mesh vertices in order to simplify the processing. Fig. 3.1 shows what it looks like for a snake-like mesh (it naturally follows the shape of the mesh)[Levy 2006]. The Graph Laplacian denoted $L = (a_{i,j})$ is a matrix defined as follow:

$$\begin{aligned} a_{i,j} &= w_{i,j} > 0 && \text{if } (i,j) \text{ is an edge} \\ a_{i,i} &= -\sum_j w_{i,j} \\ a_{i,j} &= 0 && \text{otherwise} \end{aligned} \tag{3.4}$$

being $w_{i,j}$ the weights associated with the graph edges. The interested reader should refer to [Lévy 2009] for more details and explanations.

Laplacian Beltrami : continuous setting

In the continuous setting, the laplacian operator called also laplace operator is extremely important in mechanics, electromagnetic, wave theory, and quantum mechanics. The laplacian operator is defined as the divergence of the gradient given by the following expression :

$$\Delta = \text{divgrad} = \nabla \cdot \nabla = \sum_i \frac{\partial^2}{\partial x_i^2}. \tag{3.5}$$

It is important to note that the eigenfunctions of the Laplace Beltrami (Manifold harmonics) define basic functions. However, the problem occurs in the calculation

of eigenvectors for large mesh size. Considering discrete meshes, many cotangent schemes have been proposed to estimate the Laplace-Beltrami operator [Meyer 2002, Reuter 2006, BelkiIn 2008] in order to overcome the current limits.

3.2.1.2 Applications

Spectral modeling in the case of 3D shape, consists in computing the eigenvalues and eigenvectors of a discrete Laplace operator. This eigen-decomposition is applied in various applications to achieve different tasks. Furthermore, a signal defined on a triangle mesh can be projected into the eigenvectors taken as a basis. The obtained coefficients of spectral transform can be analyzed or processed further. In this paragraph we present the applications which used the spectral transform or the eigenvectors of mesh Laplace. This kind of modeling occupies a very important position in various fields. Among these, Karni and Gaustman's work [Karni 2000] which consists in realizing a 3D shape compression scheme based on a spectral decomposition method. The main idea is to project the mesh geometry on the eigenvectors of the Laplacian matrix associated to the object. Thus, a spectrum (see Fig. 3.2) represented by geometrical coefficients (spectral coefficients) is then quantized and transmitted in ascending order of the frequency associated with each coefficient. This spectral analysis is considered as a generalization of the cosine transform on irregular surface meshes.

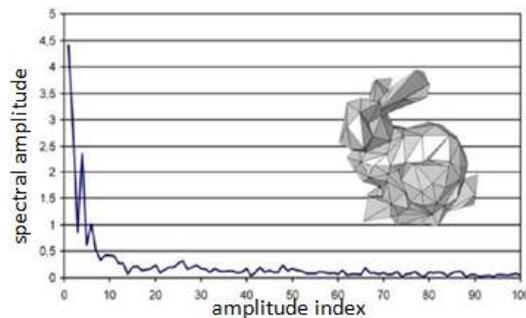


Figure 3.2: Geometric spectrum of simplified Bunny mesh (100 vertices).

In the literature, there are several watermarking techniques applied in the spectral domain to improve the robustness and imperceptibility tasks. To obtain a frequency representation of the mesh, they use the Laplacian matrix of size D ($N \times N$). The obtained N eigenvalues and N eigenvectors are standardized and sorted in ascending order according to their associated frequencies. The N spectral components are calculated respectively by projecting the cartesian coordinates (x, y, z) on the normalized and stored eigenvectors. Liu *et al.* [Lui 2013] have used the classical spectral analysis to insert the watermark into 3D meshes. Their method consists in devising the low frequency part of the spectrum in 5 mesh patches. A bit is then inserted in each patch by changing the relative relationship between a certain selected spectral amplitude and the average of the different

spectral amplitudes in the same patch.

In the context of mesh parameterizations and remeshing, approaches using the spectral domain have the interesting property of connecting local entities in a way that lets a global behavior emerge. A spectral technique presented in [Mullen 2004] consists in computing the first solution orthogonal to the trivial one, that is to say, the eigenvector associated with the first non-zero eigenvalue.

In the segmentation and clustering field, Huang *et al.* [Huang 2009] developed an hierarchical shape segmentation method using spectral analysis. The aim is to detect shape parts which would remain rigid over the deformations. Authors used an operator that encapsules shape geometry over the static setting. The objective aimed at defining a certain deformation energy and use the eigenvectors of the Hessian in order to characterize the space of possible deformations of an input mesh. The optimal computed partition is the one whose optimal articulated deformation, defined on the parts of the decomposition, conforms the best to the basis vectors of the space of typical deformations.

Spectral based modeling is also used as a local descriptor into 3D shape retrieval scheme. In [Sun 2009], authors present a concise and informative multi-scale signature based on heat kernel properties. The latter is calculated by restricting the well-known heat kernel to the temporal domain. The heat kernel does not admit an explicit function; it can be calculated using the Laplace-Beltrami operator. Authors in [Sun 2009], propose to use the cotangent scheme proposed in [BelkiIn 2008] to approximate the Laplace-Beltrami operator and calculate the set of eigenvalues and eigenfunctions.

3.2.2 Conformal geometry based modeling

Conformal geometry based modeling has been used in various applications of computer vision and graphics. In the literature, several studies have been made on conformal geometry mapping in surface parametrization analysis. This concept can be considered as an embedding procedure, which maps a 3D surface with disk topology to a planar domain D . In fact, conformal geometry theory supposes that each 3D shape with disk topology can be mapped to 2D domain using a global optimization [Wanf 2007].

Conformal map is a map which only scales the first fundamental forms preserving angles. It is one of mathematical tool in conformal geometry theory, which does not require to specify the boundary condition reverse unlike harmonics map. The latter are simple and easy to compute. However, the need to satisfy boundary condition makes them unreliable especially when the input data has occlusions. Therefore it is necessary to approximate the missing boundaries.

A conformal map is characterized by important properties:

- its connection to complex function theory.
- the surface S is determined by the mean curvature and the area stretching factor defined on the parameter domain.
- only two corresponding points may determine a conformal parametrization.
- conformal parametrization does not depend on the connectivity of surfaces, only the geometry is concerned.

3.2.2.1 Conformal geometry theoretical background

Looking for mapping the surface S to the planar domain D . Let's designate by U a conformal map from S to D ($d = U(s)$), and (u, v) are the coordinates on the planar domain D . A conformal mapping U satisfies the Cauchy-Riemann equations given by:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}. \quad (3.6)$$

$$d = u + iv \quad , \quad s = x + iy \quad (3.7)$$

The two Laplacian equations listed in equation eq 6.3, are obtained by differentiating those of eq6.2

$$\begin{aligned} \Delta u = 0 \quad , \quad \Delta v = 0 \\ \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad . \end{aligned} \quad (3.8)$$

In the case of discrete meshes, the existing work, on conformal parametrization, are based on: harmonic energy minimization, Laplacian operator linearization, angle based flattening method, Cauchy-Riemann,... Riemann's theorem demonstrates that it is possible to find for each surface S , homeomorphic to disc, a parametrization of the surface while satisfying eq.5.3. This conformal parametrization is determined using only two corresponding points on the surface S . Fig.3.3 illustrates a conformal map from a original real human face (a) to a square (b) while preserving the angles on the surface.

The Least Squares Conformal Map (LSCM) parametrization algorithm generates a discrete approximation of a conformal map by adding a constraint [Wanf 2007]. So that, 3D surface can be mapped to 2D domain using the LSCM method by considering the multiple correspondence as constraints.

3.2.2.2 Applications

Conformal geometric maps have been used in several applications dealing with surface characterization, thank's to their interesting properties. To compute conformal parametrization, Gul *et al.* [Gu 2003, Gu 2004] proposed using harmonic energy minimization with holomorphic 1-forms. These approaches

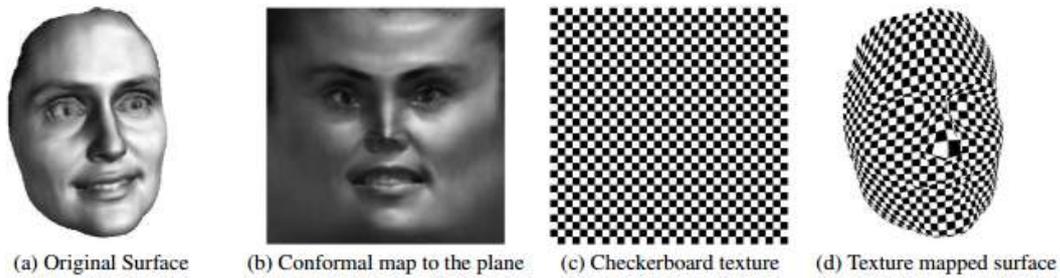


Figure 3.3: Conformal mapping of an original real female face (a) to a square (b). A checker board texture (c) mapped back to the face preserving the right angles on (d) from [Gu 2003].

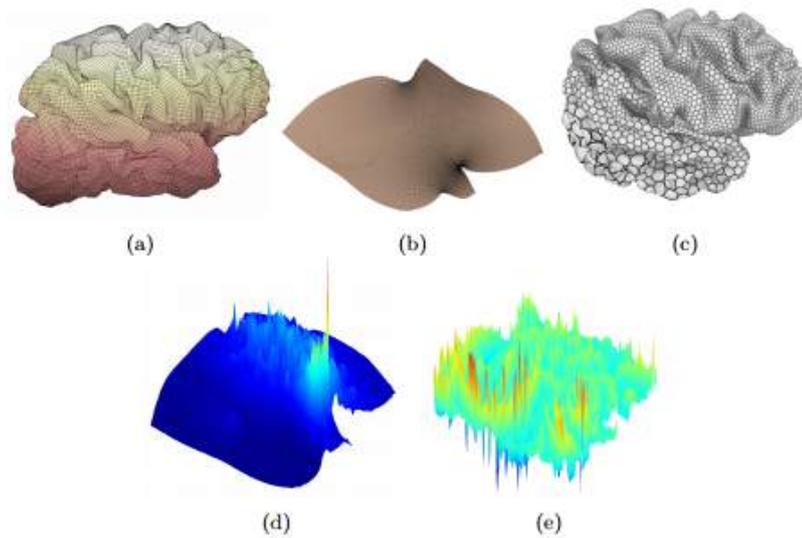


Figure 3.4: Conformal parameterization of an original brain mesh (a) in a rectangular conformal parameter domain (b). In (c) the brain is textured using the parameterization obtained from (b). (d) and (e) show the conformal factor and the mean curvature of the brain surface respectively, tired from [Lam 2014].

are used for brain human surface registration. In [Gu 2003, Gu 2004] a conformal parametrization approach is proposed based on the least-square conformal map.

In the field of 3D shape matching, various works use 2D image-matching solutions. Indeed, 3D surfaces have an inherent 2D structure since they are manifolds. As a result, the problem will be more tractable using conformal geometry theory. Based on conformal map, the authors in [Lam 2014] proposed a face and brain surface matching scheme. Wang *et al.* [Wanf 2007] presented a 3D face recognition

and 3D non rigid surface alignment based on harmonic maps, conformal maps and least-squared conformal maps.

3D surface classification is closely related to 3D shape retrieval fundamental problem. A surface can be classified based on different transformation groups as topological or Euclidean transformation groups. Gu and Yau [Gu 2003] proposed to classify surface by using conformal transformation groups. They assume that the conformal equivalent class is finer compared to the topological one. Hence, it is suitable for particle classification purposes.

In [Lam 2014], two algorithms have been proposed, according to the conformal geometry, for surface morphine and multi-scale representation. The main objective of these algorithms is to represent the 3D surface by its means curvature and conformal factor functions. The geometry is then determined according to Riemann surface theory using these two functions. These algorithms were tested on 3D human face models and MRI-derived brain surfaces. Fig. 3.4 presents the obtained results of the conformal parametrization with the means curvature and conformal factor computation.

3.2.3 Riemannian geometry based modeling

In the field of shape analysis, various work use Riemannian geometry modeling, which is definitively the most abstract geometry modeling paradigm discussed so far. In this context, objects are represented by a finite number of salient points or landmarks (point in an Euclidean space \mathbb{R}^2 or \mathbb{R}^3) called *shape space*, representing all the possible prolongments of the original input object. An equivalence is established regarding transformations that will not affect the shape such as rotation, translation and uniform scaling. The finite-dimensional Riemannian manifold (shape manifold) is represented by the obtained quotient space. Approaches that use Riemannian metric on the shape space lead to well defined statistics such as means and covariance on shape spaces.

In the next section we introduce the Riemannian manifold notion noted (M, g) . M designs the smooth manifold and the metric g defines a distance function which is usually used to compute angles and lengths of curves in the shape. This metric provides an inner product on each tangent space.

3.2.3.1 Riemannian manifold notion

In the case of differentiable manifold, the Riemannian paradigm is an interesting example of the *tensor field* notion. For a smooth manifold M , the commutation ring of smooth functions on M and the set of smooth vector fields on M are denoted by $C^\infty(M)$ and $C^\infty(TM)$, respectively. The smooth vector field sets $C^\infty(TM)$ forms

a module over $C^\infty(M)$:

$$C_0^\infty(TM) = C^\infty(M). \quad (3.9)$$

For each positive integer number n , let $C_n^\infty(TM)$, given by equation 6.9, be the n -fold tensor product of $C^\infty(TM)$ over $C^\infty(M)$.

$$C_n^\infty(TM) = C^\infty(TM) \otimes \dots \otimes C^\infty(TM). \quad (3.10)$$

In what follows, we present most important properties of the Riemannian manifold notion.

Definition 1. A smooth tensor field, noted by A , of a differentiable manifold M of type (n, s) is defined as a map $A : C_n^\infty(TM) \rightarrow C_s^\infty(TM)$. The map A is multilinear over $C^\infty(M)$ such that:

$$\begin{aligned} & A(X_1 \otimes \dots \otimes X_{k-1} \otimes (f.Y + g.Z) \otimes X_{k+1} \otimes \dots \otimes X_n) \\ = & f.A(X_1 \otimes \dots \otimes X_{k-1} \otimes Y \otimes X_{k+1} \otimes \dots \otimes X_n) \\ + & g.A(X_1 \otimes \dots \otimes X_{k-1} \otimes Z \otimes X_{k+1} \otimes \dots \otimes X_n) \end{aligned} \quad (3.11)$$

where $X, Y, Z \in C^\infty(TM)$, $f, g \in C^\infty(M)$ and $k = 1, \dots, n$. The most important property is that the $A(X_1 \otimes \dots \otimes X_n)$ value at a point $p \in M$ depends only on the vector fields values.

Definition 2. A Riemannian metric g on a smooth manifold M is a tensor field $g : C_2^\infty(TM) \rightarrow C_0^\infty(TM)$ such that for each point $p \in M$ the restriction g_p of g to $T_pM \otimes T_pM$ is an inner product on the tangent space T_pM .

$$g_p : (X_p, Y_p) \mapsto g(X, Y)(p). \quad (3.12)$$

The Riemannian manifold study is called Riemannian geometry and we note Riemannian manifold the pair (M, g) .

Definition 3. Two Riemannian manifold (M, g) and (N, h) . A map $\phi : (M, g) \rightarrow (N, h)$ is called **conformal** if there exists function $\lambda : M \rightarrow \mathbb{R}$ such that for each $X, Y \in C^\infty(TM)$:

$$e^{\lambda(p)} g_p(X_p, Y_p) = h_{\phi(p)}(d\phi_p(X_p), d\phi_p(Y_p)), \quad (3.13)$$

where $p \in M$. The exponential function e^λ is the **conformal factor** of the tensor field ϕ . If $(\lambda = 0)$ the conformal map is said to be **isometric**.

Definition 4. We denote by $I(M)$ the set of all isometries of a Riemannian manifold (M, g) . If $\phi, \psi \in I(M)$, so the composition $\psi \circ \phi$ and ϕ^{-1} are isometries. $(I(M), \circ)$ is called the **isometry group** of the Riemannian manifold (M, g)

Definition 5. The pair $(I(M), \circ)$ called isometry group of (M, g) is also said **transitive** if $\exists \phi_{pq} : M \rightarrow M$ for all $p, q \in M$ such that $\phi_{pq}(p) = q$, where ϕ_{pq} is an isometry. Consequently the Riemannian manifold (M, g) is called **Riemannian homogeneous space**.

3.2.3.2 Applications

As we have shown previously, working in a Riemannian manifolds gives nice properties, which solves many geometry processing tasks and various geometric modeling by endowing the set of closed orientable manifolds. Riemannian geometry based modeling has been applied in various applications include shape morphing, shape deformation, intuitive shape exploration and so many other applications, we discuss some of them, the most studied in the literature, in the next section.

In the context of matching 3D facial surface. Bronstein *et al.* [Bronstein 2006] study the matching and deformation points in an isometric way based on lengths of geodesic paths. In the same concept Charpiat *et al.* [Charpiat 2005] proposed a Riemannian framework using the Gromov-Hausdroff distance instead of the geodesic one. To address non rigid face matching, Lu *et al.* [Lu 2005, Lu 2006] have propose to use thin plate splines.

3D face surface comparison is another widely studied application. In [A. 2009], Srivastava *et al.* proposed to compute geodesics between pairs of faces by considering geodesic's lengths as the geometrical distances between the faces. In another hand, Samir *et al.* [Samir 2006] used the level curves of the height function to define and study shapes of facial curves.

Recently many research work have focused on deforming 3D facial surface. The main objective of these work is to chose a certain criterion. In order to study shapes of anatomical parts in the medical domain, Grenander's deformable template theory [Grenander 2010] consists in subdivided the set of non-rigid deformations into linear and nonlinear deformations. Kilian *et al.* [Kilian 2007] proposed a numerical approach to calculate geodesic conductivities between triangulated surfaces using Euclidean metrics. This approach does not address the invariance to arbitrary parameterizations of surfaces.

Shape exploration is another important application used Remanning geometry modeling. In [Kilian 2007], authors proposed a new framework to analysis shapes. The main objective consists in exploring the space of isometric deformations of an input object. After a preprocessing step, the input shape is mapped to a set of 2D point. A Delaunay triangulation of this point set is then constructed. Two scalar function based on the geodesic distance and the length of the curves obtained by linear interpolation, respectively, are assigned with each edge of this triangulation. Finally a refinement step is added only if the difference between the two scalar function is greater than a defined threshold.

Fig. 3.5, taken from [Kilian 2007], shows the planar triangulation obtained from some selected poses of the Armadillo shape after adding two refinement steps.

We also see in the same figure a curve drawn during the exploration phase.

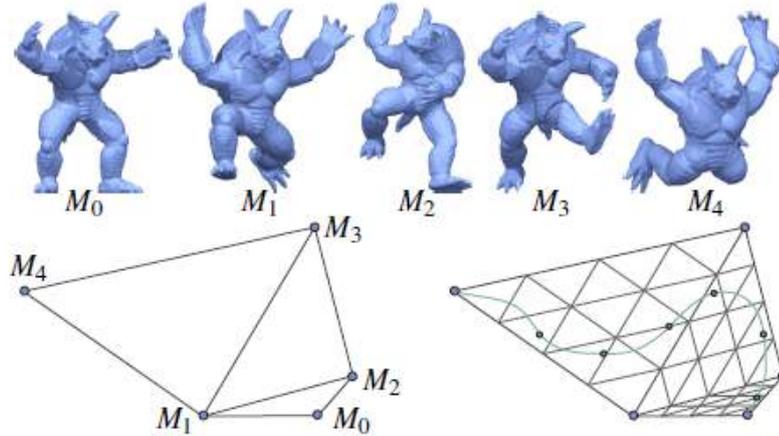


Figure 3.5: The top row shows the five input poses of the armadillo shape. The bottom row (left) shows the 2D planar triangulation obtained by adding two refinement steps. The bottom row (right) shows the curve drawn in the exploration phase.

3.3 Topology modeling

The surface topology is defined by the set of its characteristics, which are invariant to smoothing transformations. Thus, this intrinsic representation is very interesting for various shape modeling applications.

3.3.1 Curve skeletons

Curve skeleton extraction is fundamental shape feature tasks in 3D digital geometry processing and shape retrieval. This kind of representation provide an intrinsic structural shape description, which will be very useful in the field of 3D shape modeling applications.

This compact representation is a modern variant of shape skeletons based on the axe median transformation. In the past, it has been extensively studied in the case of 2D planar contours. Using the axe median transformation, makes this structural descriptor very sensitive against local distortions. In addition, the axe median transformation is very complex, in the case of 3D objects, which makes the skeleton extraction is very costly in terms of time.

The axe median of an object is defined by Blum in 1967 as the set of the centers of maximal disks included in the object [Blum 1967]. A disk is maximum if it is not included in any other record in the object. As is shown in Fig. 3.6, the

obtained result is a line graph in 2D. Each arc of the graph is the median, that is to say, it is in the middle of some part of the form and acts as an axis of symmetry.

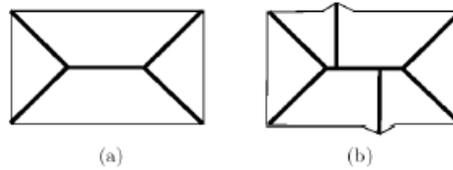


Figure 3.6: A linear 2-D graph obtained after applied the axe median transformation with the central axis (a). (b) shows the graph of the sensitivity to artifacts.

The skeleton model is initially defined as a collection of the center of the largest inscribed sphere. Commonly, curve-skeleton based methods can be roughly classified into three categories: semantic oriented segmentations, distance transform based (ridge detection) and general field functions based.

Authors in [Sun 2010] proposed a new geometric approach to extract skeleton based on feature points and core extraction by applying the multidimensional scaling (MDS) transformation. Using MDS allows to decompose the mesh into several prominent branch components and a core component to extract the skeleton.

In the literature, there is an important number of skeleton based topological approaches that study the properties of continuous functions computed over triangulated surface. The defined mapping function must be, all the time, invariant to shape deformations and different transformations. In the same context, Tierny *et al.* [Tierny 2006a] proposed a 3D mesh skeleton extraction approach based on geometrical analysis. The main purpose of this method consists in detecting feature points located on the shape extremities. Than, considering the set of extracted feature points, a mapping function based on the geodesic distance is defined on each vertex in the mesh. Thus allows to construct the Reeb graph of the mesh. Finally, a constriction approximation is applied in order to enable Reeb graph refinement into more enhanced topological skeleton. The different stages of this algorithm are illustrated in the Fig. 3.7.

While the majority of research in this area lies on 3D objects, now the trend is to turn to the 3D time domain (3D+t). 3D dynamic meshes are becoming a media of increasing importance. They constitute a fundamental and time consuming task in 3D animation systems. The issue of kinematic structural representation has rapidly gained the interest of the scientific community in recent years. Few existing work in the literature have been proposed to extract kinematic skeletons for 3D dynamic meshes [Aguiar 2008, Schaefer 2007].

Recently Tierny *et al.* [Tierny 2008a] proposed a new approach to precisely extract the kinematic skeleton for a given 3D dynamic mesh based on Reeb graph

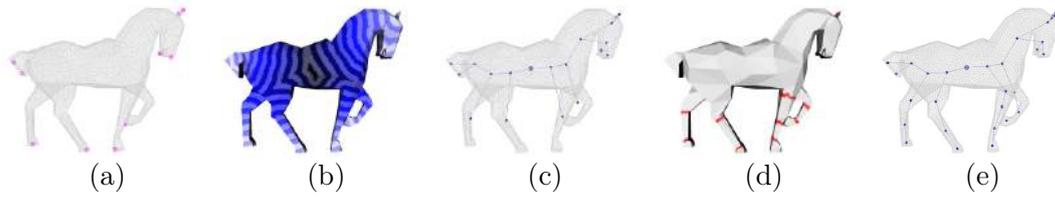


Figure 3.7: Main stages of Tierny *et al.* [Tierny 2006a] proposed enhanced topological skeleton approach. (a) Feature points extraction, (b) Mapping function definition, (c) Reeb graph construction, (d) Constriction approximation and (e) Enhanced skeleton construction.

construction. The 3D shape is represented by a set of level lines which depend on the edge-length deviation induced by the 3D shape transformation over the time. Tierny *et al.* [Tierny 2008a] algorithm allows detecting the shape articulations. However, in some cases, it fails to detect these articulations when the edges located on those articulations remain immobile through time. To overcome this problem, more recently Lavoué *et al.* [Benhabiles 2012] developed a precise kinematic skeleton extraction method for 3D dynamic meshes. The objective aimed at was to detect all the boundaries, including the immobile parts connected to the shape's articulations. Authors in [Benhabiles 2012] proposed to separately compute a set of regions of interest for each mesh in the sequence. A unique segmentation is then performed for the whole sequence by linking all the interest regions obtained from each mesh to extract the kinematic skeleton.

Topological skeletons have shown to be higher level shape descriptions, They are widely used in divers field applications such as shape metamorphosis, deformation, 3D shape retrieval and texture mapping.

Within the framework of pattern recognition, skeleton based descriptor allow conserving the high-level information that corresponds to the intuitive description of the 3D shape. Nevertheless, graph representation based on the construction of the skeleton which may encounter some difficulties like high computation cost, sensitivity to noise on the shape surface, and/or the need of predefined control points. Li *et al.* [Li 2014] applied a skeleton-based approach in a non rigid 3D shapes matching scheme. The extracted skeleton integrate both geometrical and topological features of 3D object. The main objective of there algorithm is to compare geodesic paths between skeleton graphs. In order to measure the similarity between 3D object, authors in [Angela 2004] proposed to use internal skeleton graphs. The latter is generated using a progressive mesh technique. The obtained skeleton graphs is than simplified by removing insignificant nodes of degree two. Finally the similarity is computed based on backtracking.

Skeleton graphs are also good supports for shape deformation. To show the usability

of the enhancing 3D mesh topological skeleton, Tierny *et al.* [Tierny 2006a] used this approach for shape deformation. Each skeleton node references each mesh sub-component vertex. Thus it is easy to apply deformations on selected parts of the shape. In this work, authors applied simple rotation to components on the shape. Then, they compute a rotation matrix using an angle and an axis of rotation. Finally, the obtained matrix will be applied to each vertex of the selected node. Thus producing nice-looking deformations as showing in Fig. 3.8 .

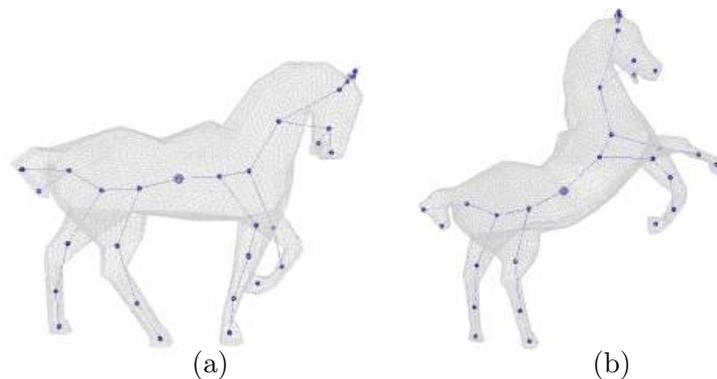


Figure 3.8: Example of application: mesh deformation. (a) Enhancing 3D mesh topological skeleton, (b) Its application to deformation.

3.3.2 Segmentation

Segmentation and skeletonisation are two closely related tasks. One being the driver to the other. Mesh segmentation is often formulated as graph clustering. It has been studied in computer vision, especially for compression and simplification purposes. It consists in partitioning mesh elements (vertices, edges and faces) into disjoint sets according to certain criterion which can be grouped into trois classes. The first one is defined as a feature of the mesh such as area, size or length. The second class is defined as differential property such as curvature, normal direction. The third class is defined as distance measures such as geodesic distance, distance to the medial axis, or the shape diameter. These surface measures are generally extracted prior to the segmentation of the mesh.

Existing segmentation approaches can be roughly classified into five categories: region growing, hierarchical segmentation, iterative segmentation, spectral analysis based methods and graph-cut approaches.

Choosing the segmentation category and/or the criterion function used for the optimization are released specially to the field of applications and the segmentation objectives (surface-type and part-type segmentations). For example, surface-type segmentation is mostly used for texture mapping, parametrization,

building charts and geometry-image creation. These kind of applications require the topologically equivalence of the sub-mesh patch to a disk. Remeshing and simplification are also two important applications that use surface-type segmentation. The main idea is to replace each cluster either by one or set of planar polygon.

Furthermore, part-type segmentation is used for 3D shape recognition and understanding based on structural decomposition. It is also used for modeling by assembling parts of shape and reforming new objets from these parts.

The criterion choice, used to decompose shape into parts, can affect directly the segmentation results. The main idea is to extract attributes from the surface mesh. According to the desired application, there is different attributes such as: planarity of forms, geodesic distance, curvature information, medial axis, shape diameter function, difference in normals of vertices or dihedral angle between faces and other geometric proxies (spheres, cylinders, developable, surfaces).

The most useful attributes are based on surface properties of the mesh. We note the differential property curvature and the averages geodesic distances. In [Sokrine 2005] another related attributes is defined as the shape diameter function (SDF). The latter measures the local diameter of the shape at the points located on its boundary. In Fig. 3.9, authors in [Shamir 2006] compare three attributes, minimum curvature, average geodesic distance and shape diameter function. We can notice that compared to the minimum curvature attribute, using geodesic distance and SDF allow partitioning the mesh into clusters consisting of topologically connected vertices. Furthermore, the SDF attribute allows detecting all the boundaries, including the immobile parts connected to the shape's articulations.

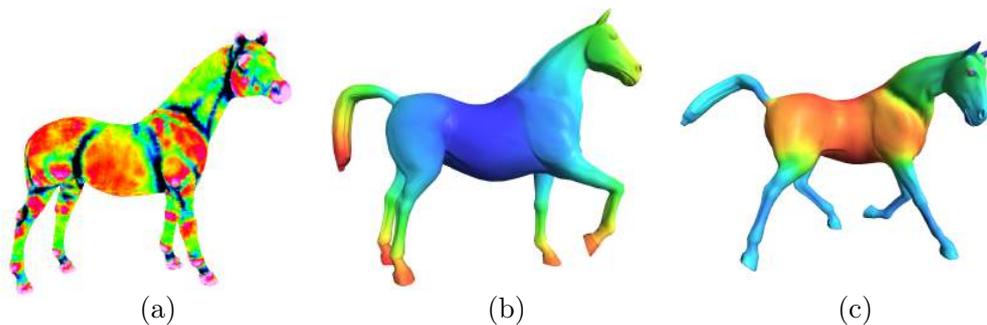


Figure 3.9: Example of mesh attributes used for segmentation. (a): Minimum curvature, (b): Average geodesic distance and (c) Shape diameter function.

In [Shapira 2008], authors proposed to use volume-based shape-function usually called the shape diameter function (SDF) in order to partition articulated objects to parts. The definition of the SDF is invariant to rigid body transformations and to pose changes of the shape and maintains similar values in analogue parts of different shapes. It is defined on the mesh faces as scalar function. It consists in

computing the diameter of the object's volume in the neighborhood of each vertex on the shape. Thus, SDF, allows us to explore the connection from the mesh to the shape volume instead of the surface attributes.

Recently, the issue of 3D static mesh segmentation has rapidly gained the interest of the scientific community. However, few work have dealt with motion based segmentation for 3D dynamic meshes [Lengyel 1999, Amjoun 2006, Boulfani-Cuisinaud 2007]. The pioneer method developed by Lengyel [Lengyel 1999] proposes to partition the mesh into components, whose motion can be accurately described by a 3D affine transform. This heuristic approach select randomly 10% of triangles in mesh. The remaining vertices are classified according to the motion of these triangles. The drawback of this approach resides in the large number of clusters which is independent of the motion.

Amjoun *et al.* [Amjoun 2006] extended this approach, to develop region growing segmentation algorithm. The authors propose to search the N farthest triangles in geodetic sense. The choice of triangles germs is based only on the geometry of the first frame. This may justify the lack of efficiency of the segmentation approach. The authors of [Boulfani-Cuisinaud 2007] propose another alternative of [Lengyel 1999]. The main idea consists on calculating, for each vertex v , the affine transform that optimally describes the motion of his neighborhood. All the vertices whose motion can be described by the same affine transform with respect to a minimum error motion compensation are grouped in the same cluster. The sub-optimal selection of vertices leads to a misclassification of the ones located on the borders between clusters.

More recently Rosman *et al.* [Rosman 2012, Rosman 2013] developed a motion-based segmentation technique to partition an articulated 3D shape into rigid parts. The proposed approach is rebuilt on the Ambrosio-Tortorelli scheme for Mumford-Shah segmentation [Mumford 1989]. The latter establishes an optimality criterion to segment the 3D shape into sub-regions. The technique presented in [Rosman 2013] consists on performing a segmentation by resolving a piecewise-smooth regularization problem.

Mamou *et al.* [K. Mamou 2006] proposed a 3D mesh compression scheme, based on a skinning animation technique. The segmentation process is based on 3D affine transforms in order to obtain the frame-wise motion of each region by weighting previous affine transforms. Motion compensation is then performed followed by Discrete Cosine Transform (DCT) applied on residual errors.

3D mesh segmentation has emerged as an important issue because of the multitude of related applications. A shape can be considered from a purely geometrical point of view or a more semantically oriented. The manner in which

a mesh is interpreted depends on the intended application and leads to a surface patch segmentation or partitioning in significant parts. In the first case, the mesh is partitioned into patches which characteristics can meet the constraints on curvatures border flatness, convexity, etc. In the second case, the objective is to detect significant parts of the object. It is worth noting that segmentation surface patches are often used in the texture mapping, mesh parametrization, re-meshing and simplification, pattern recognition, compression and so many other applications.

3.3.3 Differential topology based modeling (Reeb Graph)

The notion of Reeb graph has been introduced in 1946 by Georges Reeb [G. 1946]. It is a topological structure determined using a continuous scalar function defined on an object of arbitrary dimension. Reeb graphs allow obtaining a skeleton type of representation preserving the 3D object topological structure of objects [G. 1946]. More specifically, a Reeb graph is a data structure that represents the evolutions of the level lines of a scalar function over a mesh. According to the Morse theory, a Reeb graph can be obtained assuming a continuous function μ calculated over the 3D object surface [T. 2008]. Indeed, their construction allows characterizing the closed surface topologies. It is noted that the number of cycles in the graphs depends on the number of holes in the objects.

3.3.3.1 Morse theory

Reeb graph representation is based mainly on Morse theory, which allows to study the relationship between functions defined in a space and its shape. In the context of 3D triangular meshes, considered spaces of 2-manifold. Its worthily to note that Morse theory uses variational calculation to establish relationship between critical points (minimum, maximum and reflection point) of a differentiable function defined on a differentiable manifold and overall surface topology.

In order to study the critical points, let's designate by f a function with a unique variable, denoted by $f(x) = y$. The critical points of f are the set of points $\{x_0\}$ satisfied $f'(x_0) = 0$ where f' is the primitive of f . In the case of a function with 2 variables noted $f(x, y) = z$, the critical points are null gradient points ($\nabla f = 0$; whether $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0$).

3.3.3.2 Reeb graph definition

In the following we only consider objects which are closed 2-manifold triangular meshes with vertices located in a Cartesian frame $\mathbb{R}(x; y; z)$. Given a surface S of a 3D object and a real continuous function $\mu : S \mapsto \mathbb{R}$, the Reeb graph is the quotient space of the graph of μ in $S \times \mathbb{R}$ by the equivalence relation " \sim " between $X \in S$

and $Y \in S$:

$$X \sim Y \iff \begin{cases} \mu(X) = \mu(Y) \\ X, Y \in \mu^{-1}(\mu(X)). \end{cases} \quad (3.14)$$

For two nodes v_i, v_j :

$$(v_i, \mu(v_i)) \sim (v_j, \mu(v_j)),$$

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

Two points belonging to the same equivalence class, only if they have the same scalar function value and they belong to the same connected component. As a result, the triangular mesh M is divided into regions depending on the values of μ . A node is then associated to each region and the graph structure is obtained by linking the nodes of the connected regions. Fig.3.10 shows the Reeb graph of a torus using the height function.

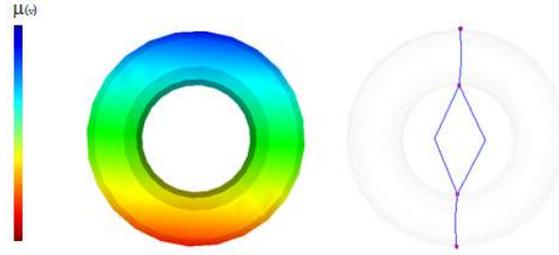


Figure 3.10: The Reeb graph of a 3D torus object using the height function.

The principal advantage of the Reeb graph representation relies on its ability to easily represent the shape topological structure. Various scalar functions have been proposed by the computer graphics community to construct Reeb graphs. According to the desired applications, the choice of the scalar function μ can effect directly the stability properties of the topological structure. In the next section, we present some of these continuous scalar functions.

3.3.3.3 Scalar functions

We note that the obtained Reeb graph aspect is closely related to the chosen scalar function. Various continuous scalar functions, having different properties, allow to construct the Reeb graph for 3D triangular meshes. The most commonly used are:

Height function: given a point $v(x, y, z)$ on the surface S , the height function is defined by $\mu(v(x, y, z)) = z$. Experiments have shown that this function is well suited for human representation, which privileging the vertical [Hilaga 2001]. Despite its simplicity and its low computation cost, this function penalized by the fact that it is totally dependent on the object orientation. In particular, its use is inappropriate for objects whose points are mainly dispersed in the directions

perpendicular to the z axis.

Barycenter function: given a point v on the surface S , the centroid function (or barycenter) is defined as the distance between v and the barycenter B of the object ($\mu(v) = d(B, v)$) where d is the Euclidean distance. This function is well known by its simplicity and stability to small perturbations.

Geodesic function: the geodesic distance between two vertices is the length of the shortest path between them along the mesh, according to a given metric. It is based on region growths from local Gaussian curvatures of vertices (germs). The obtained result usually depends on germ positions and require the calculation of local curvatures which is not always realistic, especially for objects with bad geometric quality.

Hilaga *et al.* [Hilaga 2001] proposed a new form of scalar function based on the geodesic distance. The main idea consists in computing the integral of the geodesic distances $g(v, p)$ between a given point v and all the other points of the whole surface S :

$$\mu(v) = \int_{p \in S} g(v, p) dS. \quad (3.15)$$

According to [Hilaga 2001], through the integral form, the μ function is invariant to rotation and stable against different local modifications in the shape. Applying the integral on the whole surface allows to measure the eccentricity of the surface points. The high eccentricity is driven by high value of μ and conversely the point having a minimum value of μ has a central portion of the object. Indeed, when a point is eccentric, its distance to the other points is large and therefore the integral geodesic distance is higher. Thus, this property characterizes intuitively the object topologies compared to the centroid function. Fig. 3.11 depicts the distribution of the μ function using the height, the barycenter, and the geodesic distance, on armadillo object.

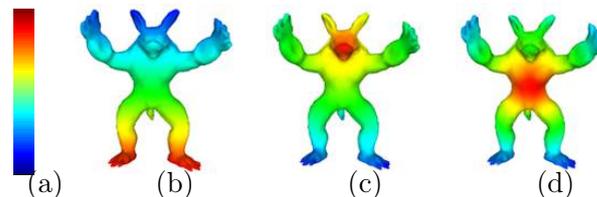


Figure 3.11: Example of different scalar function distribution in armadillo object. (a) Height function, (b) Using barycenter function and (c) Using geodesic function.

3.3.3.4 Applications

The Reeb graph can be used in a very wide range of applications; it can be extremely useful as local shape descriptor for 3D pattern recognition. In the medical field, it can be used for the segmentation of 3D scanner of human body. It is also exploited in the cartography field for terrain modeling.

In the context of 3-D model retrieval, Biasotti *et al.* [Biasotti 2003] compare Reeb graphs obtained by different functions μ and study their impact on the matching accuracy. From this study, the authors demonstrate that the integral geodesic distance [Mitchell 1987, Kanai 2000] is particularly adapted for articulated objects, whereas the distance to the barycenter may be favored if the distinction of different poses of an articulated object is required. The height function is characterized by its simplicity but it varies under isometric and affine transformations and thus it is not intrinsic to the surface.

Reeb graph driven segmentation, in fact mesh segmentation and Reeb graph representation have a close link. From a Reeb graph, we can perform segmentation and inversely. In [Tierny 2008b], Tierny *et al.* proposed automatic algorithm for Reeb graph construction based on geodesic distance. This approach consists in detecting the set of feature points (significant features) in the mesh that will be used to compute a mapping function in each vertex. Then, a segmentation process is driven used high level shape information. The segmentation algorithm is summarized in two steps based on [Berreti 2006]. First, they reduce the number of clusters by merging adjacent nodes having a degree greater than 2. Then, authors selected the most concave boundaries.

3.4 Conclusion

In this chapter we presented different modeling types. For each category, we discussed the advantages, the drawbacks, and the main applications. In general, the researchers chose the appropriate modeling strategy depending on the required application. For example, in the case of 3D shape retrieval, the shape descriptor used to extract the signature must be robust to various transformations and deformations. If the database contains only 3D rigid meshes, the robustness to non-rigid transformations may not be considered. Topological representation can be used for various applications in computer vision. In our work, we focus on the use of Reeb graph representation in 3D non rigid shape retrieval and segmentation-based compression for 3D dynamic triangular meshes.

Reeb Graph extraction based on Heat Diffusion

Contents

4.1	Introduction	49
4.2	Heat diffusion	50
4.2.1	Heat kernel	50
4.2.2	Laplace-Beltrami operator	52
4.2.3	Diffusion distance	53
4.3	Survey on Reeb graph extraction	54
4.4	Proposed method	56
4.4.1	Feature points extraction	56
4.4.2	Reeb graph construction	58
4.5	Experimental results	62
4.5.1	Parameter setting	63
4.5.2	Accuracy assessment	63
4.5.3	Robustness evaluation	67
4.5.4	discussion	67
4.6	Conclusion	68

4.1 Introduction

In this Chapter, we describe the heat diffusion principle, adapted to Riemannian manifolds which is the base of Reeb graph construction approach. Thereafter, we present the principal contribution, which is the core of our research. It is a novel approach to construct Kinematic Reeb Graph by exploiting the temporal information. The main contribution consists in defining a new scalar function, based on the eccentricity in term of diffusion distance. The latter is computed from the discrete representation of the shape. To compute the scalar function, we propose extracting the set of feature vertices in the shape. This crossed analysis is based on two diffusion distance functions. Thank's to the important properties of the heat kernel, the obtained Kinematic Reeb graph is invariant to isometric , affine and non rigid transformations.

4.2 Heat diffusion

The heat diffusion on a given surface M is fully described by the heat kernel associated with the Laplace-Beltrami operator Δ_M . In this section, we give a brief overview of the heat diffusion on Riemannian manifolds. It is worth mentioning that the proposed Reeb graph construction approach is rebuilt on defining a new scalar function. The latter is based on this concept of heat diffusion on 3D shape surface, which is fully described by the heat kernel associated with the Laplace Beltrami.

4.2.1 Heat kernel

Let's denote by M a Riemannian manifold, the heat diffusion for $x \in \partial(M)$ is given by the following expression:

$$\Delta_M^{x,t} = -\frac{\partial u(x,t)}{\partial t}, \quad (4.1)$$

where u being a function that satisfies the Dirichlet boundary condition $u(x,t) = 0$. Considering the initial heat distribution given by $f : M \rightarrow \mathbb{R}$; the heat diffusion, denoted by $H_t(f)$, satisfies the heat constrain: $\lim_{t \rightarrow 0} H_t(f) = f$, where H_t is called the *heat operator*, H_t and Δ_M are two operators that map one real function defined on M with respect to the following equality: $H_t = e^{-t\Delta_M}$. Thus if we designate Δ_M eigenvalue by λ , the eigenvalue that corresponds to H_t will be expressed by $e^{-t\lambda}$. Authors in [Hsu 2002] claimed that for each Riemannian manifolds M , we can define a heat kernel function denoted by $K_t(x,y)$. This function represents the amount of the heat transmitted from x to y at t .

The heat kernel is called also the Gaussian transition density function. It is the basic analytic object associated with the Brownian motion on the manifold \mathbb{R}^n . This explains that for any Borel subset $C \subseteq M$:

$$\int_C K_t(x,y)dy = P(X_t^x \in C), \quad (4.2)$$

where P is the probability of Brownian motion displacement from a point x to C over t .

The function of the heat kernel is characterized by important properties:

Intrinsic property

The following proposition demonstrate the heat kernel invariance under isometric deformations. This is a direct consequence of the invariance of the Laplace Beltrami operator. The equation of the heat kernel involves intrinsic properties of the manifold. Thus it can be extremely used to process and analyze 3D object under different isometric deformation.

Proposition 1. *We define an isometry noted T between two Riemannian manifolds M and N . For any $x, y \in M$ and any $T > 0$:*

if $T : M \rightarrow N$,

then $K_t^M(x, y) = K_t^N(T(x), T(y))$.

Informative property

The heat kernel contains all the information about the geometry of the object. Thus it characterizes completely the shapes up to isometry as demonstrated in the following proposition:

Proposition 2. *We define a subjective map T between two Riemannian manifolds M and N . For any $x, y \in M$ and any $T > 0$:*

if $K_t^N(T(x), T(y)) = K_t^M(x, y)$,

then $T : M \rightarrow N$ is an isometry.

This proposition is demonstrated by the following equation eq.4.3:

$$\lim_{t \rightarrow 0} t \log k_t(x, y) = \frac{-1}{4} d^2(x, y), \quad (4.3)$$

where $d(x, y)$ designs the geodesic distance between two points x and y on a manifold M .

Multi scale property

Another important property; the heat kernel function is determined by the neighborhoods of a point i on a manifold M . for small values of t , these neighborhoods are small, and grow bigger as t increases. Consequently, we can notice that the heat kernel function $K_t(i, \cdot)$ only reflects local properties of the object in the neighborhoods of i , for small values of t . While, increasing t , $K_t(i, \cdot)$ reflects global properties of M around i .

For more explanation, we consider the heat diffusion paradigm on a subset of M , D denotes a compact and smooth domain on M . H_t^D is the heat operator associated with D satisfies the dirichlet-boundary conditions. K_t^D is a local amount depends just on D . In the following proposition, we demonstrate that the heat kernel function $K_t^D(x, y)$ is a good approximation of $k_t(x, y)$ even when the values of t are large as long as D is big.

Proposition 3.

For any compact and smooth domain $D \subseteq M$: $\lim_{t \rightarrow 0} K_t^D(x, y) = k_t(x, y)$.

D_1 , and D_2 denote two compact and smooth domains on M . For any $x, y \in D_1$ and any $t \in \mathbb{R}^+$: if $D_1 \subseteq D_2$ then $K_t^{D_1}(x, y) = K_t^{D_2}(x, y)$.

If $\{D_n\}$ is an expanding compact domain sequences, then $\lim_{n \rightarrow \infty} K_t^{D_n}(x, y) = k_t(x, y)$.

Stability property

The heat kernel is stable against perturbations on deformable models. This important property comes from the heat kernel interpretation as the transition probability of the Brownian motion on M . The latter is a basic continuous time Markov process with continuous sample paths. Its probabilistic behavior is only defined by its transition density function *GaussianHeatkernel*. This means that the heat kernel function between two points x and y is obtained by the weighted average over all possible euclidean paths between x and y at t . Which should not be significantly affected by local perturbations of deformable surface. For example, if we suppose a Brownian motion on a manifold M and we introduce a perturbation on a subset $D \subset M$, then only the paths in D will be affected.

4.2.2 Laplace-Beltrami operator

The heat kernel has nice properties which makes it very suitable in various applications. However it does not admit an explicit function; it can be obtained as the fundamental solution of the heat equation associated with the Laplace-Beltrami operator, given by the following expression:

$$k_t(x, y) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(x) \phi_i(y), \quad (4.4)$$

With λ_i and ϕ_i are respectively the i^{th} eigenvalues and eigenfunctions of the Laplace-Beltrami.

The laplace-Beltami operator serves as the infinitesimal generator for Brownian motion on a manifold. We can briefly defined this operator by the following definition:

Definition 6. *Let's denotes by $\langle \cdot, \cdot \rangle$, a Riemannian metric. The gradient $\text{grad}f$ of a function f on a manifold M is a vector field defined by:*

$$\text{For any } X \in \Gamma(M), \langle \text{grad}f, X \rangle = X(f), \quad (4.5)$$

where $\Gamma(M)$ is the space of smooth vector field. The laplace-Beltrami operator $\Delta_M f$ is given by:

$$\Delta_M f = \text{div}(\text{grad}f). \quad (4.6)$$

We observe that, in the theory (eq. 4.4), the heat kernel is related only on the eigenvalues and eigenfuctions of the Laplace-Beltrami operator. Thus this fundamental can be exploited in the pattern recognition field to compare shapes in different representations especially in the internet application.

In this work, we suggest to use the solution proposed in [BelkiIn 2008] to approximate the Laplace-Beltrami operator and calculate the set of eigenvalues

and eigenfunctions. For a given Mesh M , the Laplacian operator matrix L can be established as $L = A^{-1}W$, with W is an adjacency matrix and A is a positive diagonal matrix. $A(i, i)$ represents the area associated with the corresponding vertex i . The objective is to solve the equation $W\phi = \lambda A\phi$. Consequently L can be written by $L = \Phi \Lambda \Phi^T A$, where Λ denotes the diagonal matrix of eigenvalues and Φ designates the matrix whose columns correspond to the right eigenvectors of L .

In the case of discrete setting, equ. (4.12) is replaced by $L_{ut} = \frac{\partial u_t}{\partial t}$. The solution of this equation takes the following form $u_t = e^{-tLu_0}$, where u_0 is an arbitrary vector which represents the distribution of the initial heat, and e^{-tL} is an exponential matrix given by equ. (5.2):

$$e^{-tL} = \sum_{i=0}^{\infty} \frac{(-tL)^i}{i!}. \quad (4.7)$$

Based on equ. (4.4), the exponential matrix can be written as: $e^{-tL} = \Phi e^{-t\Lambda\Phi^T} A = k_t A$. Therefore, in the discrete setting, the heat kernel matrix k_t is given by:

$$k_t = e^{(-tL)A^{(-1)}}. \quad (4.8)$$

4.2.3 Diffusion distance

The diffusion distance is closely related to the heat kernel. It measures the connectivity path between two points $x, y \in M$ at the given time t . It is defined by the following equation:

$$\begin{aligned} d_t^2(x, y) &= k_t(x, x) + k_t(y, y) - 2k_t(x, y) \\ &= \sum_i^{\infty} e^{(\lambda_i t)} (\phi_i(x) - \phi_i(y))^2. \end{aligned} \quad (4.9)$$

The heat kernel $k_t(x, x)$ is represented by the average behavior of the heat dissipation in the neighborhood determined at a given time t . The average behavior in a small neighborhood is sensitive to changes in the temporal domain. Conversely, the average behavior in a large neighborhood is resistant. Therefore, the diffusion distance depends on local information when the variable t is small. This is due to the lack of significant distribution spread. The eccentricity of x in term of the diffusion distance denoted by $ecc_t(x)$ is defined as the mean square of the diffusion distance on the whole surface of M :

$$\begin{aligned} ecc_t(x) &= \frac{1}{A_M} \int_M d_t^2(x, y) dy \\ &= k_t(x, x) + H_M(t) - \frac{2}{A_M}, \end{aligned} \quad (4.10)$$

being A_M the surface area of M and $H_M(t)$ its heat trace. At the given time t , $H_M(t)$ is defined by:

$$H_M(t) = \sum_i e^{-\lambda_i t}. \quad (4.11)$$

4.3 Survey on Reeb graph extraction

Reeb graph is an interesting candidate for 3D shape intrinsic structural representation. Several works have been developed to propose a Reeb graph construction for rigid models [Hilaga 2001, Tung 2005a]. The proposed approaches have been later extended to non-rigid models [Tierny 2006b, Gal]. In particular, the pioneer method developed by Hilaga *et al.* [Hilaga 2001] proposes to calculate a scalar function based on geodesic distance. Gal *et al.* [Gal] extended this context for non-rigid models. Nevertheless, scalar function based on geodesic distance is penalized by its sensitivity to topology changes.

Based on Hilaga *et al.* [Hilaga 2001] scheme, for a given point $v(x, y, z)$ on the object surface S , the function $\mu(v(x, y, z))$ is defined as the integral of the geodesic distance $g(v, p)$ from v to the other points p of the surface:

$$\mu(v) = \int_{p \in S} g(v, p) dS. \quad (4.12)$$

For a practical implementation, the function μ is computed using a discrete formulation. Assuming that the surface S is approximated through a discrete mesh M represented by a triangulation of vertices. For each vertex $v \in M$, $\mu(v) = \sum_{p \in M} g(v, p) \cdot area(p)$, with $area(p)$ is the area that p occupies, and $\sum_{v \in M} area(v)$ is the $area(M)$. Since it is computationally expensive to compute all geodesic distances, Hilaga *et al.* propose to compute the function μ by approximating the geodesic distance with the length of the shortest piecewise linear path on mesh vertices, according to the Dijkstra's algorithm. During the Reeb graph construction, the geodesic distances are not calculated from all the surface vertices, but rather from a small number of evenly spaced vertices. The latter are taken away from a distance $d = 2\sqrt{0.005 \cdot area(S)}$, with $area(S)$ represents the whole area of the surface S . Even so, this alternative is still diminished by high computational cost.

Hilaga *et al.* [Hilaga 2001] introduced the concept of multi-resolution Reeb graph (MRG), which is a hierarchy of dichotomies of μ base domain. Each node of a MRG corresponds to a connected set of a pre-image by μ of a given interval of \mathbb{R} . As a result, one can study the geometry of these connected sets. However, MRG does not distinguish the important critical vertices of μ from the insignificant ones. Consequently, it may comprises many insignificant branches and could thus make the definition of a simplification strategy difficult in the smooth case.

To overcome this problem, Ni *et al.* [Ni 2004] suggested to compute *fair* Morse functions with a minimum number of critical vertices. However, this alternative reduces the number of critical points based on a user-controlled process, which prevents its application in automatic Reeb graph extraction contexts.

Biasotti *et al.* [Biasotti 2006] proposed an extended Reeb graph construction, using a finite set of contour levels, and spherical harmonics sub-parts signatures. The obtained Reeb graph is robust to isometric transformations.

Tierny *et al.* [Tierny 2009] proposed an automatic technique for the computation of function μ which is intrinsic to the surface, and whose critical vertex set depicts the most important geometrical features of the surface. Their approach consists in taking $\mu(v) = g(v, v_f)$, where $g(v, v_f)$ stands for the geodesic distance and v_f for the closest feature point (point located on the extremity of the mesh) from v . Authors in [Tierny 2009] have shown that their approach gives better emphasis on local shape features thanks to the feature points. However, the Reeb graph construction may be discriminated by the use of the only closest feature points in the geodesic scalar function formulation. Steered by the outcome of this analysis, we present a novel Reeb graph construction strategy, which forms the core of the proposed 3D-shape matching system.

In [Aouada 2010], authors developed a new topological shape skeleton representation called *suigraph* based on Morse theory. This proposed geometric modeling approach consists in embedding, in Euclidean space, a manifold of new characteristic iso-geodesic curves. Recently, authors in [Mohamed 2012] proposed an invariant Reeb graph algorithm based on normalized mixture distance function.

More recently, El Khoury *et al.* [El Khoury 2012] proposed a 3D-model retrieval scheme based on indexed closed curves. To construct the Reeb graph, the authors have defined a mapping function using commute-time distance which is proved to be robust to isometric transformations as well as non rigid ones.

In the last decade, with technological progress in telecommunication, hardware design and multimedia, 3D dynamic content are becoming a media of increasing importance. It is subject to a wide variety of processing operations such as segmentation, compression and indexation. Consequently it needs an intrinsic structural descriptor before being processed by one of the aforementioned operations. The issue of kinematic structural representation has rapidly gained the interest of the scientific community in recent years. Few existing work in the literature have been proposed to extract kinematic skeletons for 3D dynamic meshes [Aguiar 2008, Schaefer 2007]. 3D dynamic shapes are usually represented by a sequence of 3D meshes with constant connectivity and temporal information provided by time-varying geometry (only the vertices position changes over time).

Recently Tierny *et al.* [Tierny 2008a] proposed a new approach to precisely extract the kinematic skeleton for a given 3D dynamic mesh based on Reeb graph construction. The 3D shape is represented by a set of level lines which depend on the edge-length deviation induced by the 3D shape transformation over the time. Tierny *et al.* [Tierny 2008a] algorithm allows detecting the shape articulations. However, in some cases, it fails to detect these articulations. Principally, this phenomenon

occurs when the edges located on those articulations remain immobile through time. To overcome this problem, more recently Lavoué *et al.* [Benhabiles 2012] developed a precise kinematic skeleton extraction method for 3D dynamic meshes. The objective aimed at was to detect all the boundaries, including the immobile parts connected to the shape’s articulations. Authors in [Benhabiles 2012] proposed to separately compute a set of regions of interest for each mesh in the sequence. A unique segmentation is then performed for the whole sequence by linking all the regions of interest obtained from each mesh to extract the kinematic skeleton.

Reeb graph is an interesting candidate for 3D shape intrinsic structural representation. Several works have been developed to propose a Reeb graph construction for rigid models [Hilaga 2001, Tung 2005a]. The proposed approaches have been later extended to non-rigid models [Tierny 2006b, Gal]. In particular, the pioneer method developed by Hilaga *et al.* [Hilaga 2001] proposes to calculate a scalar function based on geodesic distance. Gal *et al.* [Gal] extended this context for non-rigid models. Nevertheless, scalar function based on geodesic distance is penalized by its sensitivity to topology changes.

In our work, we propose a kinematic Reeb graph extraction approach which exploits the temporal information. The main contribution consists in defining a new scalar function, based on the eccentricity in term of diffusion distance. The latter is computed from the discrete representation of the shape. To extract the feature points, we propose to run a crossed analysis using two diffusion distance functions, based on the farthest vertices. Non-isolated extrema are then purged by intersecting the set of local extremities of both diffusion distance functions.

4.4 Proposed method

A Reeb graph is a structure that represents the evolutions of the level lines of a given continuous function, defined over objects of any dimension (k -manifolds) [Biasotti 2008]. The proposed approach for Reeb graph construction is performed in two steps. First, we extract the feature points based on the diffusion distance. Then, the set of these feature points is taken as the initial data to compute the continuous function μ based on the eccentricity in term of the diffusion distance.

4.4.1 Feature points extraction

Vertices located on the extremities of the mesh, representing the local extremum, are considered as feature points. In the literature, many approaches were proposed to extract feature points for rigid and non rigid models. Tierny *et al.* [Tierny 2008b] proposed to detect the local extremum using two functions based on geodesic distances. The algorithm starts with the search of the two farthest points in the geodetic sense. Eventually, the authors defined two geodesic functions that correspond to extremity regions in order to extract two local proprieties groups. The intersection of these two groups provides the set of feature points.

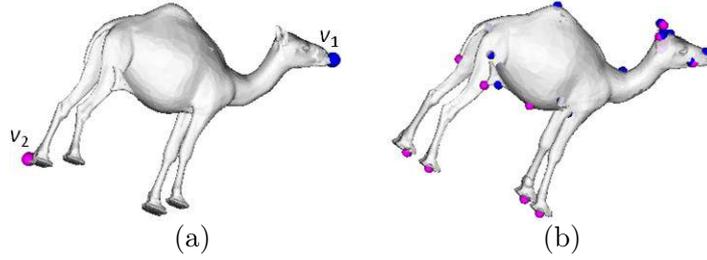


Figure 4.1: Feature points detection. Fig (a) shows the two farthest points v_1 and v_2 and Fig (b) shows the two sets of local minima of $f(v, v_1)$ and $f(v, v_2)$ corresponding to v_1 and v_2 respectively

This approach produces a set of well-localized points. But they are very sensitive against topological changes.

In the case of 3D dynamic Meshes, we propose to proceed in the same way for better accuracy. To ensure stability under eventual perturbations over time, we propose to use the diffusion distance instead of geodesic distance. It consists to determine the connectivity paths between two points $x, y \in M$ at the given time t . In Section 4.2 we showed the important properties of heat diffusion on manifolds. For this reason, we want to take advantage of its benefits to propose a new strategy to well detecting feature points, on 3D mesh sequences sharing the same connectivity. Our proposition is inspired principally from the famous Tierny's [Tierny 2008b] algorithm. Global properties of the shape in large neighborhoods are detected through the behavior of heat diffusion over longer time, while local properties are detected through the behavior of heat diffusion over short time. In addition, for small t , the variation of the heat kernel function is large but decays as t increases. This is explained by the fact that the heat kernel function is defined by the average behavior of heat dissipation in the neighborhood determined at the given time t . In a small neighborhood, the average behavior is sensitive to temporal changes, while average behavior in a large neighborhood is more stable. Therefore, to ensure an accurate detection of feature points which are stable over time, we scale the temporal domain logarithmically. This gives a more faithful approximation of local shape properties at the choosing time range $[t_1, t_2]$. In the following equations we define two function $f_{v_1}, f_{v_2} : M \rightarrow \mathbb{R}$, which allow to calculate the scaled diffusion distance between each vertex on M and the two farthest points (v_1 and v_2 respectively).

$$f_{v_1} = df_t^2(v, v_1) = \int_{t_1}^{t_2} k_t(v, v) + k_t(v_1, v_1) - 2k_t(v, v_1) d \log t.$$

$$f_{v_2} = df_t^2(v, v_2) = \int_{t_1}^{t_2} k_t(v, v) + k_t(v_2, v_2) - 2k_t(v, v_2) d \log t.$$

Let v_1 and v_2 be the most distance vertices, in the geodetic sense, of an given closed connected triangulated surface M .

we denote by v_i the vertex that maximizes f_{v_1} on M . For each vertex on M , $f_{v_i} = df_t^2(v, v_i)$. The best approximation is given by $df_t^2(v_2, v_i)$. The ensemble of feature points, F , is the intersection of the two sets f_{v_2} and f_{v_i} , which contain the detected points. Thus, $F = f_{v_2} \cap f_{v_i}$ will be used as origin to compute the scalar function defined in the next subsection.

$$v \in F = f_{v_2} \cap f_{v_i} \iff \begin{cases} \exists v_{f_{v_2}} \in f_{v_2} & df_t^2(v, v_{f_{v_2}}) < \varepsilon \\ \exists v_{f_{v_i}} \in f_{v_i} & df_t^2(v, v_{f_{v_i}}) < \varepsilon \\ df_t^2(v, v_{f_{v_n}}) > \varepsilon & \forall v_n \in F \\ \varepsilon, df_t^2 \in [0, 1]. \end{cases} \quad (4.13)$$

Fig. 4.1(a) shows the two farthest vertices v_1 and v_2 illustrates with two different colors. In Fig. 1.1(b), the vertices in subsets F_1 and F_2 have been displayed in blue and in magenta respectively. The feature points extraction procedure is summarized in Algorithm 1.

Algorithm 1 Feature points extraction

Require: a discrete triangular surface S .

Require: two farthest points v_1 and v_2 .

Ensure: a set of feature points F .

- 1: $\varepsilon = 0.074$
 - 2: Two collections of feature points F_1 and F_2
 - 3: **for all** $v \in S$ **do**
 - 4: **while** $f(v; v_1) < \varepsilon$ **do**
 - 5: $F_1 \leftarrow v$
 - 6: update(F_1)
 - 7: **end while**
 - 8: **while** $f(v; v_2) < \varepsilon$ **do**
 - 9: $F_2 \leftarrow v$
 - 10: update(F_2)
 - 11: **end while**
 - 12: **end for**
 - 13: $F \leftarrow F_1 \cup F_2$.
-

4.4.2 Reeb graph construction

According to the Morse theory, a continuous function defined on a closed surface characterizes the topology of the surface on its critical points. Thus, a Reeb graph can be obtained assuming a continuous function μ calculated over the 3D object surface [T. 2008]. In the following we only consider objects which are closed 2-manifold meshes of triangles with vertices located in a Cartesian frame $\mathbb{R}(x; y; z)$.

Several studies have been carried out in order to define the scalar function for Reeb graph construction. A non pertinent function entails an important number of critical points. As a result, the associated Reeb graph will be made up of a large number of nodes and will not offer a meaningful description of the model. Furthermore, depending on the application needs, the scalar function must verify the stability invariance properties of the topological structure.

To guarantee invariance and stability of the graph structure we propose to define an appropriate continuous scalar function based on diffusion distance. Since the diffusion distance preserve all the geometric information about the neighborhoods of a given point over the time. The proposed continuous function is concise and preserve almost all of the information contained in the heat kernel. In section 4.2 we have shown that there is a lot of redundant information in the heat kernel. That is means that the proposed scalar function is informative if we only consider its restriction to the temporal domain. This scalar function based on diffusion distance is easily commensurable since it defined over the common temporal domain. It can be computed faithfully from the discrete representation of the shape.

As the diffusion distance does not admit an implicit expression, it can be calculated by using a cotangent scheme called mesh Laplace operator. For an input closed connected and orientable triangulated mesh M containing n vertices, the mesh Laplace operator L is a sparse matrix of $n \times n$ size. This matrix is given by: $L = A^{-1}W$ with A is a positive diagonal matrix whose element $A(i, j) = A(j, i) = 0$ and $A(i, i)$ defines the area associated with vertex i . W is a symmetric matrix. To calculate the diffusion distance, we must simply solved the following problem $W\phi = \lambda A\phi$ and therefore computed the set of eigenvalues λ and eigenvectors ϕ . The mesh Laplace operator matrix can be writen as $L = \Phi\Lambda\Phi^T A$, where Λ and Φ are two matrix of eigenvalues and eigenvectors of L respectively.

Previously in Section 4.2, we have demonstrate that the heat kernel is stable in the smooth case. In the discrete case, estimating the heat kernel through the eigen-decomposition of the mesh Laplace operator is still also stable against noise. In the following proposition, we demonstrate this property.

Proposition 4. *Let $L = A^{-1}W$ be the mesh laplace operator of a given mesh M , if we introduce a noise on M we obtain M' noisy mesh with $L' = A'^{-1}W'$. $A' = A + E$ and $W' = W + F$ where $\|E\| < \varepsilon$ and $\|F\| < \delta$. If K_t and K'_t are the induced heat kernels computed from L and L' , then the difference $\|K_t - K'_t\| = O(\sqrt{\varepsilon} + \delta)$.*

After the feature points extraction, we define the continuous function μ which will be used for the Reeb graph construction. Hence, for a given vertex v we propose to compute $\mu(v)$ based on its eccentricity, given by equation 6.3. The continuous



Figure 4.2: The stability of the scalar function over time on the horse model, red to blue colors express the increasing values of the scalar function.

function μ is then defined as follows:

$$\mu(v) = \frac{1}{\text{area}(S)} \int_{p \in S} d_t^2(v, p) dp \quad (4.14)$$

being $\text{area}(S)$ the area of the surface S . For a practical implementation the μ function is computed, using discrete formulation, as the sum of the eccentricity from v to each one of the feature points. Assuming that the surface S is then approximated through a discrete triangular mesh M , for each vertex $v \in M$, $\mu(v)$ is established as follows:

$$\mu(v) = \frac{1}{\text{area}(M)} \sum_{p \in F} d_t^2(v, p) \text{area}(p). \quad (4.15)$$

where $\text{area}(M)$ is the surface area of M , F represent the set of feature points which are extracted in the first step and $\text{area}(p)$ is the area that p occupies. The defined μ function has the advantage to be invariant to isometric transformations. Due to the integral form over M , the μ function is more stable in the case of local modifications. Indeed, the integral of the diffusion distance over the mesh surface allows measuring the eccentricity of surface points. More the value of μ is high at a point, more this point is eccentric. Conversely, the point having a minimum value of μ occupies a central portion of the object, which make it a centered point. If a point is judged to be eccentric, this means that its distance to the other points is greater, and therefore the integral of geodesic distances is higher. Then, we can argue that compared to the the geodesic distance, the proposed function characterizes intuitively the shape topologies. Fig. 4.2 shows the distribution of scalar function on the horse sequence. The function values increase as the color goes from blue to green and to red, with the mapping consistent across the shapes. From Fig. 4.2 we can notice that the colors remain the same on each frame, which clearly highlight the high stability of the proposed scalar function over time. Finally, the Reeb graph is obtained by iteratively partitioning the triangular mesh M into regular intervals depending on the values of μ_n . A node is then associated to each region and the graph structure is obtained by linking the nodes of the connected regions. The proposed strategy of Reeb graph construction is summarized in Algorithm 2.

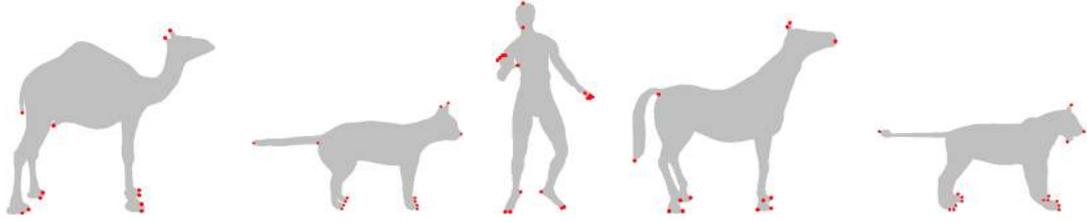


Figure 4.3: Feature points extraction in the reference frame of the different sequences.

Algorithm 2 Discrete Reeb graph computation algorithm.

Require: a discrete triangular surface M .

Require: a set of feature points F .

Ensure: the discrete Reeb graph R .

- 1: Surface scalar Field T
 - 2: Number of Tuples n in $T \leftarrow$ number of vertices in M
 - 3: **for all** $v \in M$ **do**
 - 4: **for all** $v_f \in F$ **do**
 - 5: $\mu(v) += g(v; v_f).area(v_f)$
 - 6: **end for** $\frac{1}{area(M)} \sum_{p \in F} d_t^2(v, p) area(p)$.
 - 7: $T \leftarrow$ Tuple $(v, \mu(v))$.
 - 8: free(v).
 - 9: **end for**
 - 10: $\mu^{-1}(\mu(v)) \leftarrow$ connected set of T containing v .
 - 11: update(R).
-

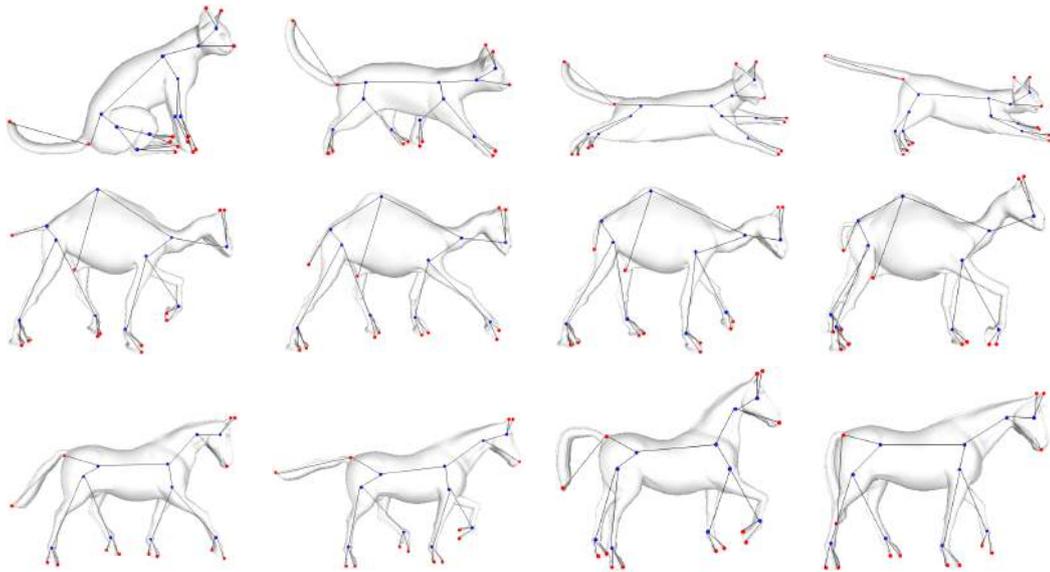


Figure 4.4: Kinematic Reeb Graph of different 3D mesh sequences.

4.5 Experimental results

The simulations were conducted for some 3D dynamic meshes characterized by their constant connectivity. Moreover, they offer a good variability in terms of spatial and temporal sizes. As an additional set of experiments, we tested the proposed Reeb graph extraction approach on 3D non rigid meshes. Table 4.1 summarizes the 3D dynamic sequence properties, expressed in terms of number of vertices, number of faces and the average running time of the feature points extraction and Reeb graph construction for each frame.

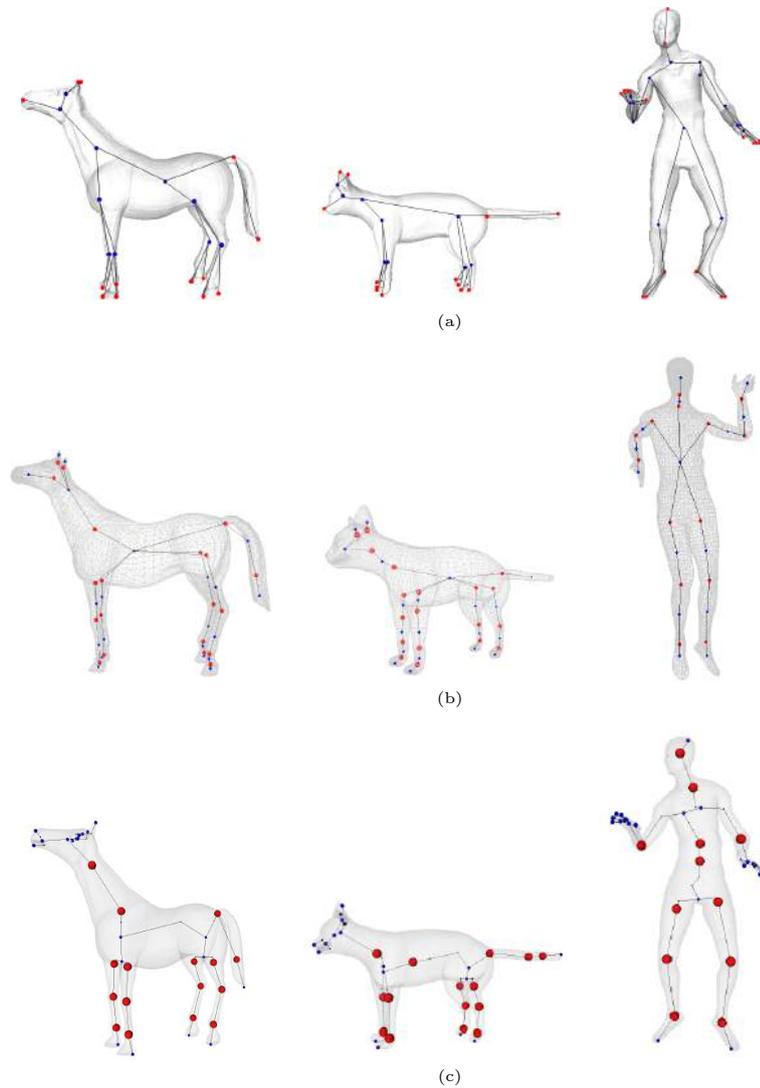


Figure 4.5: Visual comparison between Lavoué *et al.* [Benhabiles 2012] algorithm (b), Tierny *et al.* [Tierny 2008a] algorithm (c) and our method (a).

Table 4.1: Properties and computation time of the tested dynamic meshes.

	Vertices	Faces	Feature points extraction (s)	KRG extraction(s)
Horse	8431	16843	2.475	4.311
Cat	1297	2576	0.059	2.017
Lion	5000	9996	0.929	2.726
Camel	4307	8850	0.574	2.052
Dance	7061	14118	1.993	3.817

4.5.1 Parameter setting

Two parameters intervene in the proposed kinematic Reeb graph construction approach. The first one is the ε parameter, which is used during the feature points detection stage. For faithful and accurate detection, we choose an intermediate value of $\varepsilon = 0.075$, which performs better compared to all others in terms of detected vertices number. The second parameter is the time range parameter $[t_1, t_2]$, t_1 and t_2 are computed using the eigen-decomposition of Laplacian matrix. They are expressed based on the eigenvalues and eigenvectors (λ_i, ϕ_i) of the Laplace-Beltrami operator. In our implementation, we used the solver proposed in [BelkiIn 2008]. To ensure an accurate detection of feature points, we scale the temporal domain logarithmically at the choosing time range $[t_1 = \frac{1}{2\lambda_1} \frac{A_{v_i}}{A_M}, t_2 = \frac{1}{2\lambda_2} \frac{A_{v_i}}{A_M}]$. Where $A_{v_i}/i = 1, 2$ denotes the area associated with the vertices v_1 or v_2 and A_M is the global area of M . Here, we exploit the fact that larger areas may contain more global structures. Finally, during the scalar function computation, we choose a time parameter $t = \frac{1}{2\lambda_1}$, that makes the time selection invariant to the scale of the processed mesh M .

In order to assess the performance of our approach, we conducted a visual comparison between our algorithm and the kinematic skeleton extraction methods from [Aguiar 2008] and [Benhabiles 2012]. To examine the effectiveness of our method we used two analysis criteria: accuracy of Reeb graph construction, and robustness under various perturbations through time.

4.5.2 Accuracy assessment

To evaluate the execution-time, the tests were conducted on a laptop with an Intel Core 3 CPU M350 at 2.23 GHz, and operating system Windows 7 SP 1. The source code was written in C++ and the executables were generated by Visual Studio 9.0 operating in release mode. From Table 4.1 one can notice that the running time is smaller when the number of vertices decreases. This is due to the Laplacian construction and the computation of the eigenvalues vector and the eigenvector matrix. The computation process is performed independently for each frame of the tested model. Table 5.3 lists the running times for all frames while comparing our results to those obtained by the algorithms presented in [Benhabiles 2012] and [Tierny 2008a]. From the results listed in Table 5.3, we can deduce that the entire process for computing the kinematic Reeb Graph runs at a reasonable time (few

seconds) and faster than the algorithm developed by Lavoué *et al.* [Benhabiles 2012]. On the other side, the algorithm developed by Tierny *et al.* [Tierny 2008a] runs faster. Fig. 4.3 shows the effectiveness of our feature point extraction method. Fig.

Table 4.2: Computation times for the tested dynamic meshes.

Extraction time (s)	Horse	Cat	Lion	Dance
Our algorithm	173	11	23	187
Lavoué [Benhabiles 2012]	204	37	28	411
Tierny [Tierny 2008a]	12.7	9.6	6	21.2

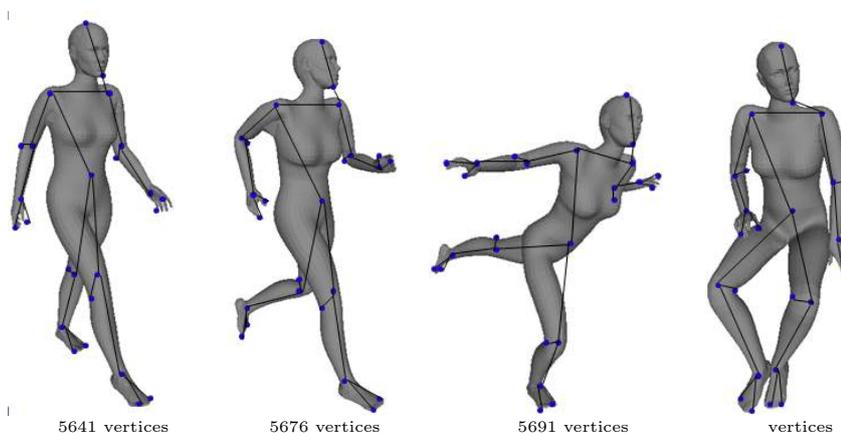


Figure 4.6: Kinematic Reeb Graph of women 3D sequence with variable connectivity.

4.3 presents reference frames corresponding to five 3D dynamic meshes (cat, horse, lion, camel and flamingo). Using a fixed $\varepsilon = 0.075$ parameter ensures obtaining a reduced number of well-localized feature points.

Fig. 4.4 illustrates the tested models, represented by a set of selected frames and their extracted kinematic Reeb graph. From Fig. 4.4 we can clearly see that the proposed approach detects a set of extremum local points (red nodes) located on the object boundary. For each tested model, we observe that the obtained Reeb graph preserves the object topology thanks to the extracted feature vertices. Indeed each node correspond to an rigid parts and each edge correspond to an articulation of the objects. Moreover the rigid parts are easily captured and the nodes maintain the same position over time, since we use a concise and provably informative scalar function based on the properties of the heat diffusion process on a shape. In Fig. 4.5, we make a visual comparison with two kinematic skeleton extraction methods [Tierny 2008a] and [Benhabiles 2012]. In [Tierny 2008a], Tierny *et al.* developed a new algorithm to precisely extract kinematic skeleton, for 3D animated sequences, using a Reeb graph construction approach. The latter is based on the edge-length deviation induced by the 3D shape transformation over the time. This approach allows detecting shape articulations (mobile parts). However, when the edges

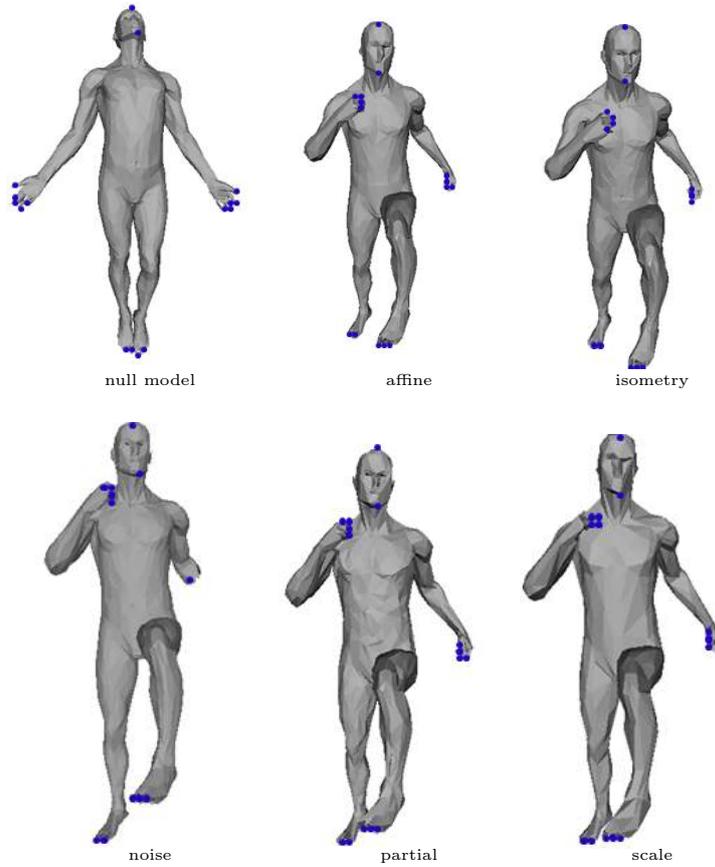


Figure 4.7: Robustness of the feature vertices detection against various transformations.

located on those articulations remain immobile through time, it fails to detect these articulations. In [Benhabiles 2012], Lavoué *et al.* developed a new precise kinematic skeleton extraction method for 3D dynamic meshes. The objective is to detect all the boundaries, including the immobile parts connected to the shape’s articulations. Authors in [Benhabiles 2012] proposed to separately compute a set of regions of interest for each mesh in the sequence. A unique segmentation is then performed for the whole sequence by linking all the interest regions obtained from each mesh to extract the kinematic skeleton. This method allows detecting of the shape articulations in both mobile parts and immobile parts.

Notice that the red nodes in Fig. 4.5(a) are the extracted feature points and the red nodes in Fig. 4.5(b) and (c) are the motion nodes that correspond to articulations of the object. Fig. 1.2(c) clearly shows that the algorithm from Tierny *et al.* [Tierny 2008a] allows to detect different articulations of non rigid parts but it fails to detect the articulations of immobile parts. Thanks to the boundary edge function, in Fig. 1.2(b) Lavoué *et al.* [Benhabiles 2012] algorithm is able to detect

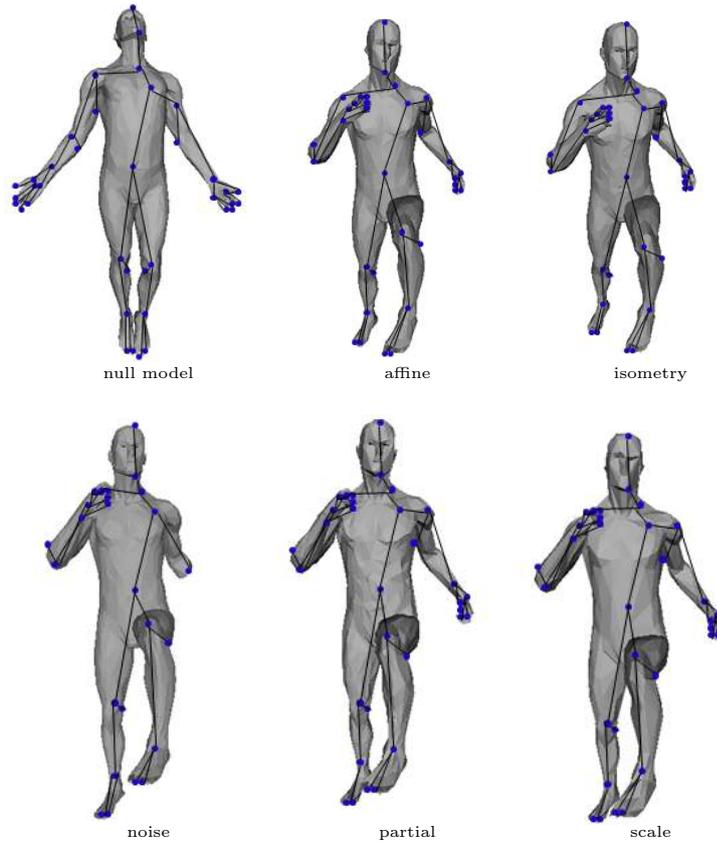


Figure 4.8: Robustness of the Reeb graph construction against various transformations.

rigid and non rigid parts, while the motion nodes are well localized. However it fails to detect the local feature points. Our method overcomes this drawback and allows to detect a well localized feature vertices and nodes. Computing the scalar function by restricting the diffusion distance to the temporal domain, captures global information contained in the heat kernel and characterized the dynamic mesh up to isometry. To prove the accuracy of our method and its applicative interest, we propose to extract the Reeb graphs associated to a sequence of 3D meshes which have variable connectivity. Fig. 4.6 shows a sequence of 3D woman meshes, the number of vertices varies from one frame to another. From Fig. 4.6 we can notice that despite the connectivity change, the feature point extraction still unchanged when varying the vertices number from one frame to another. This another advantage of our proposed scalar function, which is defined canonically. Thus allows to identify and differentiate between feature vertices on shapes. Our algorithm can be applied to both dynamic and static 3D shapes. Consequently, it is considered as a hybrid (static/dynamic) shape structural descriptor.

4.5.3 Robustness evaluation

In order to assess the robustness of the proposed scalar function, which is the central core of the Reeb graph construction, we applied some transformations like affine transformation Fig. 4.7.b, isometry Fig. 4.7.c, noise addition Fig. 4.7.c, missing parts Fig. 4.7.e and scale Fig. 2.1.f. These transformations were performed to a null model Fig. 4.7.a. From Fig. 4.7 we can observe that our algorithm produces well-localized feature points, which are stable against the tested transformations. That's a direct consequence of using the heat kernel properties, in particular, the stability under perturbations of the shape and the invariance under isometric deformations.

In Fig 4.8, we compare the extracted Reeb graphs under the tested transformations. From Fig. 4.8 we can notice that the obtained Reeb graphs are stable and invariant against the tested transformations. The nodes maintains the same positions. This proves the high stability of the proposed scalar function under almost all transformations, notably through the eccentricity in term of diffusion distance.

4.5.4 discussion

The reconstructed Reeb graph is extremely useful in various applications including full and partial shape comparison, structure detection, partial matching, shape classification and retrieval. In these applications, shapes or parts of shapes are considered to be similar if there exist rigid or isometric transformations between them. Thus, using the diffusion distance to measure the scalar function allows to detect faithfully and efficiently the set of feature vertices which leads to a concise and invariant Reeb graph that focus on most significant shape features. It also captures all information about the neighborhood of a given vertex by following the heat dissipation from a vertex to the rest of the shape over time.

The scalar time parameter provides a natural motion of scale to describe the shape around a vertex because heat diffuses progressively to large neighborhoods. That means, in particular, that local shape features are detected through the behavior of heat diffusion over short time, while the summaries of the shape in large neighborhoods are observed through the behavior of heat diffusion over large time.

Beside the proposed scalar function is invariant under isometric deformations, therefore it can be used in applications needing high level shape representations. This representation allow to perform Multi scale matching between nodes in Reeb graph by comparing their signatures and has the potential to benefit many applications such as robust discovery of correspondence, shape registration and partial matching especially in the context of deformable models. To demonstrate

the particle relevance of the proposed Reeb graph algorithm, we propose a new global 3D retrieval approach for non rigid shapes. So our reconstructed Reeb graph will be amalgamated into global signatures that can be used for non rigid shape retrieval system presented in the next chapter.

4.6 Conclusion

In this chapter, we presented a novel approach of Reeb graph construction for 3D dynamic triangular Meshes. Our main contribution consists in defining an efficient and stable continuous function based on heat diffusion properties to construct kinematic Reeb graphs for 3D models with constant and variable connectivity. The proposed scalar function is concise and provably informative. Experimental results have shown the high accuracy of the feature point detection over time, which is a direct consequence of the heat kernel invariance. Moreover, the constructed kinematic Reeb graphs preserve the topology of the tested 3D models despite the perturbations occurred over time.

In the next chapter, we plan to investigate the proposed approach and integrate it in a partial matching scheme, which can be used for non-rigid 3D shape retrieval by associated with each node, in the Reeb graph, two signatures.

Application to 3D pattern recognition

Contents

5.1	Introduction	69
5.2	Previous work	70
5.3	3D shape retrieval system	72
5.3.1	Signatures computation	72
5.3.2	Global shape similarity calculation	75
5.4	Experimental results	76
5.4.1	Experimental setup	76
5.4.2	Efficacy evaluation	79
5.4.3	Robustness assessment	79
5.4.4	Comparison with previous methods	80
5.4.5	Computation times	87
5.5	Conclusion	88

5.1 Introduction

This chapter presents a 3D shape retrieval method which is based on Reeb graph (RG) representation, described in previous chapter. First, we review some existing work about 3D shape retrieval. Then, we describe The central contribution of this work which consists to reinforce the topological consistency conditions of the graph-based description. The generated Reeb graph, based on the heat diffusion properties, is segmented into Reeb charts having a controlled topology. Each Reeb chart is mapped to its canonical planar domain. Two stretching signatures, corresponding to the area and angle distortion, are then determined. The set of all the pairs of stretching signatures is taken as 3D-shape descriptor. The similarity estimation is performed in two steps. The first one consists in forming the pairs of similar Reeb charts, according to the minimal distance between their corresponding signatures. The second step is to measure the global similarity which quantifies the similitude degree between all the matched Reeb charts. Finally, we provides experiment results and empirical comparison to demonstrate that our method provides an overall retrieval efficiency gain compared to very recent state-of-the-art methods.

5.2 Previous work

Nowadays, 3D-models constitute a crucial multimedia data type that involves a large number of applications in different domains. The blossoming demand of large 3D object repositories has triggered the need of efficient 3D search and retrieval techniques. 3D shape retrieval is the mechanism which allows users to find similar 3D objects from a database in a ranked order. The higher the ranking of an object the better the match to the query shape is by using a measure of similarity. The definition of an appropriate similarity measure is a crucial task in order to automatically and accurately evaluate the correspondence between any pair of 3D objects. For this purpose, various research studies have been investigated to define intrinsic shape descriptors. Indeed, the computed descriptor should be compact, able to perfectly discriminate 3D shapes, invariant to rigid and non-rigid transformations, and robust to any shape modifications.

Various research studies have been investigated to define 3D shape descriptors, as similarity metrics, for the purpose of 3D object retrieval [Tangelder 2008, Osada 2002, Lian 2013, Dutagaci 2012]. The earlier approaches are based on global shape representation [Chen 2003, Osada 2002, Ohbuchi 2003], whereas recent works have focused on local approaches.

The 3D shape retrieval methods, based on global shape descriptors are divided into three main categories consisting in the use of: (i) model based approaches, (ii) 2D projections of the 3D model (view based) [Chen 2003, Daoudi 2007], and (iii) shape distributions [Osada 2002, Ohbuchi 2003]. These “first generation” techniques are characterized by their compactness and ease of implementation. Nevertheless, they are not discriminative enough to resist to non-rigid deformations and articulations. To overcome these limitations, some researchers paid attention to non-rigid 3D shape matching techniques, which are based on: topology approaches [Hilaga 2001, Biasotti 2006, Dey 2006], low dimensional embedding [Reuter 2009, Jain 2007], distance over 3D model [Bronstein 2009, Smeets 2010, Sun 2010], or set of local features. The last alternative uses local descriptors (Bag of Words) to represent the 3D model [Dutagaci 2012, Lavoué 2011, Furuya 2009, Li 2012].

Local feature matching methods based on graph representation have also been investigated. Particularly, the Reeb Graph (RG) has been a very popular shape abstraction for several computer graphical tasks, especially for 3D model retrieval [Hilaga 2001, Tung 2005b, Biasotti 2008, Tierny 2009]. Its popularity is mainly due to the fact that it forms a high-level skeletal representation that corresponds to the intuitive description of the 3D shape. However, the graph construction exhibits weaknesses such as high computation cost, sensitivity to noise on the shape surface, and/or the need of predefined control points.

The concept of multi-resolution Reeb graph (MRG) in the context of global shape retrieval and indexing has been initiated by Hilaga *et al.* [Hilaga 2001]. Tung and Schmitt [T. 2008] proposed the augmented multiresolution Reeb graph (aMRG) algorithm which enhances the matching of the original MRG scheme by adding geometrical and colorimetric attributes to the topological criteria. Tierny *et al.* [Tierny 2007], developed the RCU (Reeb Chart Unfolding) scheme which estimates global shape similarity using a set of geometrical signatures. The input 3D shape is segmented into charts of controlled topology based on the constructed Reeb graph. To compare two models, the authors use the bi-parts matching algorithm presented in [Tam 2007].

Reeb graph representation has also been investigated for partial 3D-model retrieval. In this case, the 3D-model is segmented to sub-parts in order to retrieve objects that have similar sub-shapes even if they visually differ globally. Biasotti *et al.* [Biasotti 2006] proposed an extended Reeb graph construction, using a finite set of contour levels, and spherical harmonics sub-parts signatures. Experiments have shown that the Reeb graph computation is robust to isometric transformations, but sub-parts signatures (spherical harmonics) are not. Tierny *et al.* [Tierny 2009] extended the matching scheme based on Reeb chart unfolding signatures [Tierny 2007] to address the partial shape retrieval problem. The partial similarity framework presented in [Tierny 2009] consistently outperforms competing approaches for 3D shape retrieval. Nevertheless, shape topological description might be too discriminate which penalizes partial similarity.

Authors in [Aouada 2010] used the constructed suigraphs to quantify the dissimilarity between 3D shapes. Recently, authors in [Mohamed 2012] proposed an invariant Reeb graph algorithm based on normalized mixture distance function. This skeletal graph is used as a shape descriptor in a 3D shape retrieval scheme. The similarity is computed by comparing the relative shortest paths between the Reeb graph nodes. The authors [Mohamed 2012] assume the feasibility of the computed signature based on the proposed Reeb graph. Recently, Li *et al.* [Li 2013] proposed an efficient shape descriptor for non rigid 3D shape retrieval based on spectral graph wavelet analysis. The multiresolution aspect allows to compute the shape signature using a cubic spline wavelet generating kernel, in order to capture both local and global geometry of 3D shape. Experimental results on two standard 3D shape datasets demonstrate the high retrieval accuracy of this approach [Li 2013] in comparison with other state-of-the-art methods.

More recently, El Khoury *et al.* [El Khoury 2012] proposed a 3D-model retrieval scheme based on indexed closed curves. To construct the Reeb graph, the authors have defined a mapping function using commute-time distance. The proposed mapping function is proved to be robust to isometric transformations as well as non rigid ones. However, it is penalized by its low accuracy regarding 3D-model retrieval.

The Reeb graph concept has been shown to be an effective tool in 3D matching systems. Its main limitation stems from generating a graph with many insignificant branches, which may hamper the definition of global description. It is also noteworthy that the similarity calculation is a crucial task in the context of 3D-model retrieval.

In chapter we propose a 3D-shape retrieval technique based on graph representation and local feature extraction. First, an enhanced topological skeleton of the input surface is determined by defining a new scalar function. The latter is computed based on the eccentricity in term of diffusion distance. Second, a similarity measure is defined by exploiting global and local properties simultaneously. Indeed, the shape description is performed by segmenting the 3D model, denoted by Reeb charts, using its Reeb graph. A pair of geometrical signatures is then computed for each Reeb chart based on parametrization techniques. Finally, a concise global similarity measure is determined relatively to the pairs of signatures associated to all the Reeb charts in the query model. Retrieval accuracies were assessed based on commonly used performance measures. Experiments have shown that our matching method provides quite good results. It outperforms the existing 3D shape-based matching techniques in the majority of cases while ensuring high robustness to rigid transformations as well as non rigid transformations.

5.3 3D shape retrieval system

The comparison between two 3D models entails two main steps: the signature extraction (or feature vector), and similarity measure. The proposed 3D-shape matching method is based on Reeb graph construction and computation of global shape similarity. It can be subdivided into three stages:

1. shape description (Reeb graph construction),
2. signatures extraction,
3. computation of similarity between signatures.

In the following, we will describe in detail the proposed signatures and global similarity measure.

5.3.1 Signatures computation

The retrieval scheme presented in [Tierny 2009] uses a Reeb graph based segmentation to compute a concise sub-part geometrical signature which rebuilt on parameterization techniques. The input triangulated surface M is segmented into a set of Reeb charts of controlled topology, which have either disk or annulus topology. An unfolding signature is then associated to each chart, by measuring

the distortion of its mapping to the canonical planar domain D . Particularly, the authors in [Tierny 2009, Tierny 2007] use an unfolding signature of the canonical mapping, based on the evaluation of an area distortion.

In this work, we extend Tierny *et al.*'s approach by using a pair of geometric signatures which correspond to the area and angle distortion introduced by the mapping of Reeb chart to its canonical planar domain. As a result, our matching scheme uses a pair of unfolding signatures associated to each Reeb chart, unlike Tierny *et al.*'s algorithm which uses only one unfolding signature, relative to the area distortion. Another fundamental difference between the work of Tierny *et al.* [Tierny 2009] and our proposed method is that the matching system in [Tierny 2009] uses the geometrical signature as attribute to identify a partial shape similarity, whereas our approach handle global and local properties simultaneously by defining a global shape similarity measure.

In what follows, we explain the principle of computing area and angle distortions, according to the surface parametrization study presented in [Floater 2005].

Let $\Psi : M \rightarrow R(\mu)$ map each point $p \in M$ to its equivalence class in the Reeb graph R and E_i be an edge of the same Reeb graph. $C_i = \Psi^{-1}(E_i)$ is defined as a Reeb chart, which corresponds to a surface segment. Thus, each Reeb chart C_i is parameterized by its mapping ψ_i to its canonical planar domain \mathcal{D} . Owing to Reeb graph properties, the Reeb charts have either disk or annulus topology. Each edge E_i , of the Reeb graph, has two extremities whose pre-images by Ψ are circles. The latter form the two boundary components of the closure of the chart C_i having genus zero. Therefore, Reeb charts have the topology of an open annulus. Disk-like Reeb charts constitute a particular case. Indeed, a Reeb chart is mapped to a disk if it is adjacent to only one local extremum of the function μ .

In the case of annulus topology, let's designate by B_1 the boundary component of the shortest perimeter that corresponds to the annulus-like chart C_i and by B_2 the other one. As depicted in Figure 1.1, ψ_i maps B_1 to the inner boundary component of the unit planar annulus \mathcal{D}_A and B_2 to its outer boundary component. Being $\rho(v) \in (0, 1]$ the normalized absolute difference of μ values between B_1 and a vertex $v \in C_i$, from Figure 5.1, we can notice that the sub-level set of ρ have also annulus topology. Specifically, the increase of ρ parameter varies the geometry of the sub-level sets, and thus, intensifies the distortion introduced by their mapping to \mathcal{D}_A .

Consequently, to capture the evolution of Reeb charts' area variation, the unfolding signature λ_{a_i} of ψ_i mapping can be expressed by (1 is the inner radius of the unit annulus):

$$\lambda_{a_i} = \frac{\text{area}(C_i)}{\text{area}(\mathcal{D}_A)} = \frac{\text{area}(C_i(\rho))}{\pi(\rho + 1)^2 + \pi}. \quad (5.1)$$

For a given disk-like chart C_i , the mapping ψ_i , associates the local extremum O of the function μ to the center of the unit planar disk \mathcal{D}_D (the canonical domain of

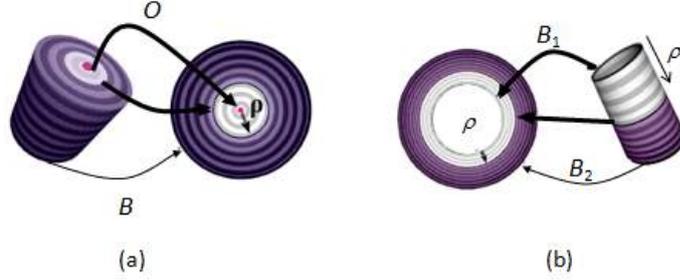


Figure 5.1: Unfolding signature computation relative to the area distortion for a Disk-like chart (a) and Annulus-like chart (b) from [Tierny 2007].

disk-like surfaces), B to its boundary and μ level lines to concentric circles. Let $\rho(p) \in]0, 1[$ be the normalized absolute difference of μ values between O and a point $p \in C_i$, the unfolding signature λ_{ai} , relative to the area distortion, of C_i according to the mapping ψ_i is given by:

$$\lambda_{ai} = \frac{\text{area}(C_i)}{\text{area}(\mathcal{D}_D)} = \frac{\text{area}(C_i(\rho))}{\pi\rho^2}, \quad (5.2)$$

where $\text{area}(C_i)$ stands for the area of the sub-level set for parameter ρ on the original surface C_i and $\text{area}(\mathcal{D}_D)$ stands for the area of the sub-level set on the unit planar disk \mathcal{D}_D .

It is noteworthy, that the mapping procedure introduces distortion in both angle and area. Consequently, an analog reasoning can be applied by computing the unfolding signature with regard to the angle distortion introduced by the mapping ψ_i of C_i to its canonical planar domain [Floater 2005, Floater 2002, Wang 2006].

Let us denote by $\theta_{c_i}(\rho)$ the angles of the sub-level set for parameter ρ on the original surface segment C_i . $\theta_D(\rho)$ stands for the angles of the sub-level set on the canonical planar domain D (either the unit disk or the unit annulus). $\theta_{c_i}(\rho)$ is approximated by the sum of the angles around ρ . Being θ_k one of these angles, $\theta_{c_i}(\rho)$ is calculated as follows:

$$\theta_{c_i}(\rho) = \sum_k \theta_k.$$

It is important to note that for any anterior node $v \in D$, the planar angles $\theta_D(\rho)$ sum up to 2π , whereas those associated to the Reeb chart C_i generally do not. As a result, the angular deformation due to the planar mapping is distributed evenly around the ρ parameter. Thus, by taking into account the angle distortion, the unfolding signature is defined by:

$$\lambda_{\theta_i} = \frac{2\pi}{\theta_{C_i}(\rho)}. \quad (5.3)$$

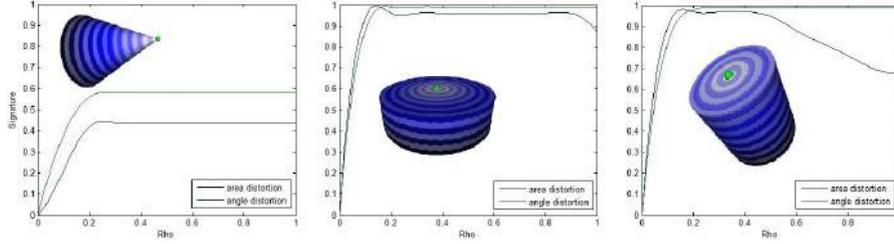


Figure 5.2: Example of stretching signatures for altered versions of primitive charts.

Figure 5.2 shows the area and the angle signatures of a specific point (green) on various disk-like primitives. From this figure we can see that these two signatures are close for small values of ρ . Furthermore, we can notice that for the 3 depicted shapes, the angular signature remains unchanged when increasing the ρ parameter. This is confirmed by equation 5.3. Indeed, the angular signature is insensitive to the variation of ρ since it is measured as a function of the sum of the angles around ρ .

To summarize, the query mesh model M_q is represented by a Reeb graph R_q which is partitioned in Reeb charts. An attribute is then associated to each Reeb chart C_i . In concrete terms, the pair of signatures $(\lambda_{\theta_i}, \lambda_{a_i})$ is taken as attribute in the global similarity measure. The signature of the whole considered mesh model consists of its dual Reeb graph, denoted by ξ_q , associated with the pairs of unfolding signatures of all the Reeb charts. A full explanation of how the global similarity measure is derived is provided in the following subsection.

5.3.2 Global shape similarity calculation

At this stage, the two mesh models M_q and M_c to be compared are represented by their Reeb graphs R_q and R_c . The similarity between two Reeb charts $C_i \in R_q$ and $C_j \in R_c$ is defined as the similarity between their attributes: $sim(C_i, C_j)$. Specifically, the similarity $SIM(R_q, R_c)$ between the two Reeb graphs R_q and R_c is given by the sum of the similarities between each pair of Reeb charts. One of the most critical tasks is how to find the Reeb chart pairs to be matched.

In our work, we propose to match each Reeb chart $C_i \in R_q$ to the most similar one $C_j^* \in R_c$ which provides the highest value of the similarity function sim . The latter is defined by:

$$sim(C_i, C_j) = 1 - L_{N1}(C_i, C_j),$$

with L_{N1} denotes the normalized L_1 distance between the signatures of C_i and C_j . The L_1 distance is determined using the pair of λ_a and λ_θ signatures which are calculated as above in equations 5.2, 5.1, and 5.3. On the other hand, the L_{N1} distance is determined as weighted sum (linear combination) of L_1 distance between

the area and angle signatures of both Reeb charts. As result, the similarity measure between C_i and the most similar charts C_j^* is given by:

$$\begin{aligned} \text{sim}(C_i, C_j^*) &= 1 - \alpha \cdot L_{N1}(\lambda_{a_i}, \lambda_{a_j}) \\ &\quad - (1 - \alpha) \cdot L_{N1}(\lambda_{\theta_i}, \lambda_{\theta_j}), \end{aligned}$$

where $(0 < \alpha < 1)$ controls the weighting of the area and angle signatures in the $\text{sim}(C_i, C_j^*)$ computation. The global similarity is then defined as follows:

$$\text{SIM}(R_q, R_c) = \frac{\sum_{C_i \in R_q, C_j^* \in R_c} \text{sim}(C_i, C_j^*)}{|R_q|}, \quad (5.4)$$

being $|R_q|$ the number of Reeb charts in the Reeb graph R_q that corresponds to the query model. The overall matching scheme is summarized in Algorithm 3.

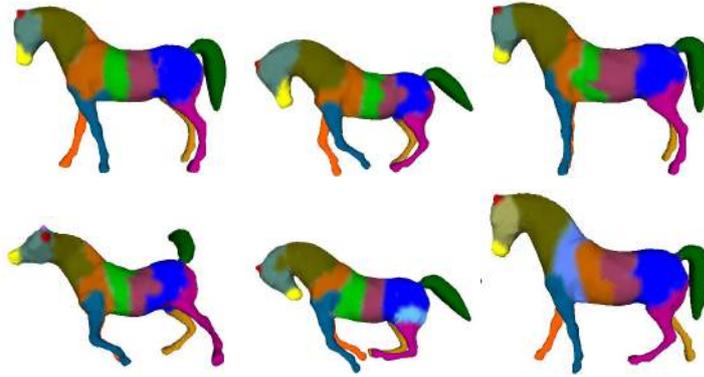


Figure 5.3: Chart similarity matchings between a query model and the top 4 retrieved objects.

Figure 5.3 shows a horse query model and the the first five results retrieved by our method. The matched charts have been displayed with the same color. From this Figure, we can observe that exceptionally for the last two retrieved models, the Reeb chart that corresponds to the leg and the neck (blue color) does not fit with the one of the query model. From this, we conclude that the hybrid signature is pose-insensitive.

5.4 Experimental results

5.4.1 Experimental setup

5.4.1.1 Data set

Three databases were used to evaluate the performance of our system.

Algorithm 3 Proposed 3D shape retrieval algorithm

Require: a collection V of vectors containing the set of charts C_j ; $V = C_1; \dots; C_n$.

- 1: Compute the Reeb graph R_q of the query model M_q .
 - 2: **for all** $C_i \in R_q$ **do**
 - 3: Compute the pair of unfolding signatures λ_{a_i} and λ_{θ_i} .
 - 4: **for all** R_c in the collection index **do**
 - 5: **for all** $C_j \in V$ **do**
 - 6: Compute $\text{sim}(C_i, C_j)$.
 - 7: **end for**
 - 8: Select $C_j^* = \arg \min_{C_j \in R_c} (\text{sim}(C_i, C_j))$.
 - 9: Remove C_j^* from V .
 - 10: **end for**
 - 11: Calculate $\text{SIM}(R_q, R_c)$ as the global similarity using $\text{sim}(C_i, C_j^*)$ for each $C_i \in R_q$.
 - 12: **end for**
 - 13: Sort the collection entries according to the decreasing order of $\text{SIM}(R_q, R_c)$ values.
-

- The McGill Database¹. It contains a rich variety of highly articulated, watertight 3D objects. Specifically, it comprises 25 objects divided into 10 classes (Ants, Crabs, Hands, Humans, Octopuses, Pliers, Snakes, spectacles, Spiders and Teddy-bears).
- SHREC 2007 - Watertight Models². This data set consists of 400 3D watertight models. It contains 20 categories, each one is composed of 20 object models represented by seamless surfaces without defective holes or gaps.
- SHREC 2010 - Shape Retrieval Contest of Non-rigid 3D Models³. This data set contains 10 categories, each one is composed of 20 non-rigid objects chosen from the McGill Articulated Shape Benchmark database. The objective of this 3D Shape Retrieval Contest is to evaluate the effectiveness of 3D shape retrieval methods for non rigidly deformed 3D models.
- SHREC 2011 - Shape Retrieval Contest of Non-rigid 3D Watertight Meshes⁴. This data set contains 30 categories, each one is composed of 20 non-rigid 3D objects. The objective of this 3D Shape Retrieval Contest is to evaluate the effectiveness of 3D shape retrieval methods on a large-scale database of non-rigid 3D watertight meshes.
- SHREC 2012 - Generic 3D Model Retrieval⁵. The data set contains different categories of 3D-models based on the combination of models from previous

¹<http://www.cim.mcgill.ca/>

²<http://watertight.ge.imati.cnr.it/>

³<http://www.itl.nist.gov/iad/vug/sharp/contest/2010/NonRigidShapes>

⁴<http://www.itl.nist.gov/iad/vug/sharp/contest/2011/NonRigid/>

⁵<http://www.itl.nist.gov/iad/vug/sharp/contest/2012/Generic3D>

generic 3D benchmarks. It consists of 1200 3D models, classified into 60 object categories based mainly on visual similarity. The collection we used comprises 12 watertight models classes (ants, humans, Tables, Sharks, Pliers, Planes, Hands, Glasses, Cups, Chairs, Bunnies, and Birds) with 10 3D-models each.

5.4.1.2 Evaluation criteria

Different performance measures can be used to evaluate retrieval methods. In general, evaluation over the data set is performed by leaving out one model to act as the query, and ranking the remaining models from most similar to least similar. This ranked list can be evaluated in different manners. Performance for a particular method is given by averaging the performance over all query models. In our work we retain the following measures:

- Nearest Neighbor (NN): The percentage of closest matches that are in the same class as the query.
- First Tier (FT): The ratio of models in the query's class that appear within the top $N_C - 1$ matches, where N_C the cardinality of the query's class.
- Second Tier (ST): The ratio of models in the query's class that appear within the top $2(N_C - 1)$ matches.
- Precision vs Recall plot: A curve illustrating the relationship between the precision and recall of a retrieval method. Precision measures the ability to retrieve all models that are relevant, while recall measures the ability to retrieve only models that are relevant.

The Precision vs Recall are defined as follow:

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

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

- E-measure (E) is a combined measure of the precision and recall for a fixed number of results. The E-Measure is given by:

$$E = \frac{2}{\frac{1}{P} + \frac{1}{R}},$$

where P and R are the precision and recall, respectively, computed for the top retrieved models.

- Discounted Cumulative Gain (DCG): A statistic that measures the gain of a matched model based on its position in the ranked list. The gain is accumulated from the top of the ranked list to the bottom with the gain of each result

reduced logarithmically proportional to the position of the result. The main idea is to reflect how well the overall retrieval would be viewed by a human.

For more details about definitions of the aforementioned performance measures, the interested reader is directed to the reference [Shilane 2004].

5.4.2 Efficacy evaluation

To evaluate the performance of our matching method, retrieval experiments were carried out from SHREC 2012 database, using the five quantitative measures: Nearest Neighbor, First Tier, Second Tier, E-measure, and Discounted Cumulative Gain. The E-measure only considers the first 10 retrieved models for each query and calculates the precision and recall values over those results since the user is more interested in the very first retrieved results than in the later ones.

Table 5.1 lists the (NN, FT, ST, E, and DCG) scores, for the whole SHREC 2012 data set trained and for some selected classes. These scores reveal the excellent results of our method for the majority of classes like Planes, Ants, Pliers, Hands, Teddies but lower accuracy for shapes as Glasses and Birds, in terms of FT and DCG scores. These limitations stem from the fact that models like Glasses and Birds have a few number of feature points, which explicitly affects the hole matching system. The results, given in Table 5.1, also indicate that our method is particularly suited for nearest neighbor classification since its NN value attains 100% for 7 categories. Line 2 of Table 5.1 shows the performance results without considering the angular measure proposed in subsection 5.3.1. From these reported results, we can notice that the additional angular measure slightly improves the retrieval accuracy in terms of ST, NN, E and DCG scores.

Figure 5.4 illustrates the precision-recall plots of our method, for the whole data set and each category from the selected collection. From this figure we can notice that all the precision-recall curves are in the top right parts. Particularly, from all the depicted curves, we observe that our method owns a high precision for low recall values, which proves that it yields to satisfactory results on all of classes.

5.4.3 Robustness assessment

Using the eccentricity of the diffusion distance, for the purpose of Reeb graph construction, ensures robustness against rigid transformation. Furthermore, the integral calculation from a finite set of feature points provides stability in the case of local deformation. Consequently, topological skeletons, computed with regard to the proposed function μ , benefit from the invariance and robustness properties of μ against affine and isometric transformations.

From a qualitative point of view, in order to prove the robustness of our matching method, different modifications have been made to a neutral pose null model. These modifications include topology change, isometric transformation

Table 5.1: Retrieval performance of our method evaluated using five standard measures on the whole collection selected from SHREC 2012 data set and for each category.

Class	FT (%)	ST (%)	NN (%)	E-M (%)	DCG (%)
All classes	92	94	86	91	85
Cups	85	82	72	67	74
Ants	92	100	100	89	92
Tables	82	85	100	77	85
Teddies	87	91	100	78	88
Birds	76	74	96	89	91
Glasses	76	82	69	59	87
Planes	87	93	100	95	88
Pliers	99	100	100	97	95
Humans	83	79	100	84	89
Sharks	79	83	99	71	84
Chairs	88	93	98	86	88
Hands	89	100	98	69	74

(triangulation and distance-preserving deformation), scaling, affine transformation, random noise, simplification and partiality missing parts. Some modified models are shown in Table 5.4.I. Performance measures, in terms of similarity distance, are listed in Table 5.4.B. From these measures, we can see that our method is able to identify objects correctly even after further modifications. Particularly, the proposed matching scheme shows strong resistance when the models undergo non-rigid deformations like isometric and scale changes. Regarding the other modifications like random noise and topology change, our method gives very satisfactory results for low to medium strength perturbations.

5.4.4 Comparison with previous methods

As an additional set of experiments, we compared the proposed 3D matching method against previous approaches from the state-of-the-art:

- The MR-BF-DSIFT-E based technique from Furuya and Ohbuchi [Furuya 2009]: The BF-DSIFT-E (Bag-of-Feature Dense-SIFT with ERC-Tree) algorithm consists on rendering a 3D model into a set of depth images. Local visual features are then extracted from each image, using the Scale Invariant Feature Transform (SIFT) algorithm [Lowe 2004]. To efficiently compare among large sets of local features, the algorithm employs bag-of-features approach. As an extension to BF-DSIFT-E algorithm, MR-BF-DSIFT-E uses a distance-metric learning approach named Manifold Ranking (MR) to derive ranked list of retrieval results given a query model.

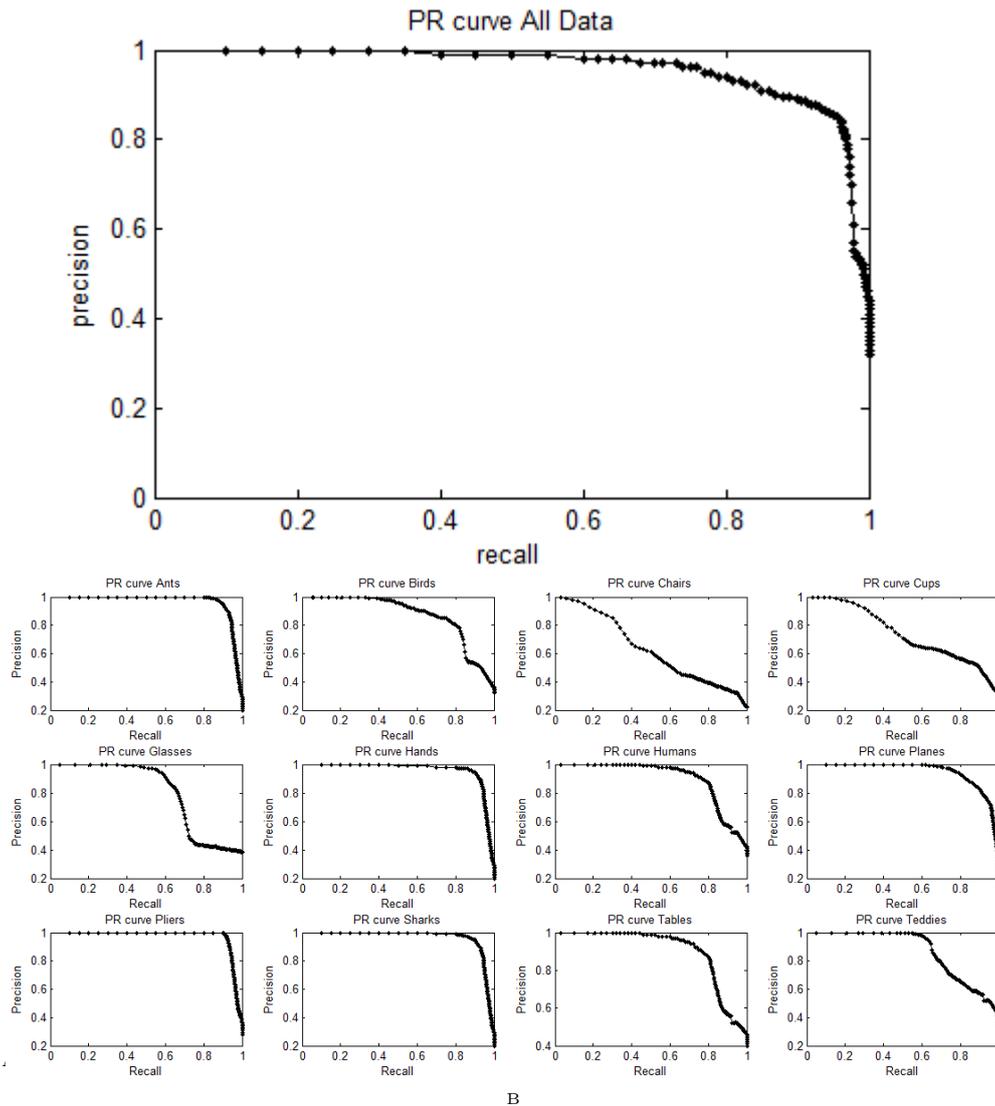
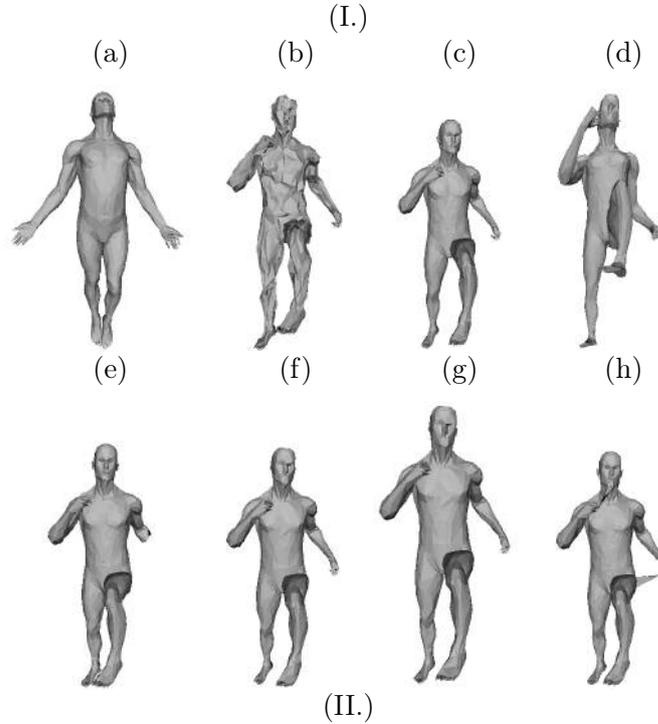


Figure 5.4: Precision-recall curves for: (A) the whole collection selected from SHREC 2012 data set and (B) each category.

- The Matching technique based on Kernels from Barra *et al.* [Barra 2013]: This method is based on Extended Reeb Graphs (ERG) description using Kernels. The similarity measure is computed through kernels adapted to the ERG description and the 3D model is represented as bags of shortest paths defined over the ERG.
- DMEVD based technique from Smeets *et al.* [Smeets 2010]: The proposed retrieval approach is based on two invariant matrices for inelastic deformation invariant object recognition. 3D objects are represented by diffusion distance tensors (DDT). In addition to DDT, geodesic distance matrices (GDM) are

Table 5.2: (I) Screen shots of the null model (a) after different transformations: random noise (b), isometry (c), affine transformation (d), partiality (e), sampling (f), scale (g), topology (h). (II) Similarity results of the matching experiments.



	a	b	c	d	e	f	g	h
a	1	0.71	0.92	0.61	0.73	0.79	0.85	0.64
b	0.71	1	0.75	0.47	0.61	0.69	0.76	0.55
c	0.92	0.75	1	0.58	0.75	0.87	0.91	0.65
d	0.61	0.47	0.58	1	0.47	0.51	0.53	0.47
e	0.73	0.61	0.75	0.47	1	0.54	0.56	0.47
f	0.79	0.69	0.87	0.51	0.54	1	0.89	0.68
g	0.85	0.76	0.91	0.53	0.56	0.89	1	0.67
h	0.64	0.55	0.65	0.47	0.47	0.68	0.67	1

also used to represent the 3D objects independent of the reference frame. The final object pair dissimilarity is given by the sum or product of dissimilarities determined by model representations of the GDM and DDT.

- The BOF based technique from El Khoury *et al.* [El Khoury 2012]: This method uses a commute time mapping function to create index of closed curves generated from the center of a 3D model. To describe all the mesh, a set of indexed closed curves is computed. The distance between models is then calculated by comparing the indexed curves.

- The BOW-LSD method from Lavoué [Lavoué 2011] : This method is based on the Bag of Words with Local Spectral Descriptors. For a given 3D shape, the proposed approach considers a set of feature points uniformly sampled on the surface and associated with local Fourier descriptors. In a pre-processing stage, a visual dictionary is constructed by clustering a large set of feature descriptors, then each 3D model is described by a histogram of occurrences of these visual words.
- The EMD-PPPT method from Agathos *et al.* [Agathos 2009] : This retrieval technique is based on the construction of attributed relation graph combined to graph matching algorithm. The latter is based on the Earth Movers Distance (EMD) similarity measure.
- The RPU method from Tierny *et al.* [Tierny 2009]

First, we compare results obtained with our method with those obtained with Barra *et al.* method [Barra 2013] using the aggregation process on SHREC 2011 database. The performance is evaluated in terms of the quantitative measures (*i.e.*, FT, ST, NN, E-M, and DCG) for the complete SHREC 2011 database. From the results reported in Table 5.3 we can see that our method achieves the best performance in terms of FT, and E-measure scores. However, the reference algorithm performs better in terms of ST, NN and DCG. This may be explained by the fact that Barra *et al.*'s approach, uses a gaussian kernel similarity measure.

Table 5.3: Comparison of similarity estimation scores on the SHREC 2011 data set obtained by our method and the method proposed in [Barra 2013].

Method	FT (%)	ST (%)	NN (%)	E-M (%)	DCG (%)
Aggregation[Barra 2013]	88.6	95.2	100	70.46	97.46
Our method	90.36	93.86	98.62	74.63	96.23

We have also tested our method on the SHREC 2010 data set and compared it with three recent methods in the state-of-the-art: MR-BF-DSIFT-E [Furuya 2009], DMEVD [Smeets 2010], and BOF [El Khoury 2012]. We conducted our tests on the average performance of the whole database. Table 5.4 shows the retrieval accuracies in terms of (NN, FT, ST, Measure and DCG) scores. For BOF method, it was not possible to make comparison in terms of the DCG score due to a lack of availability. Table 5.5 clearly indicates that compared with DMEVD and BOF algorithms, MR-BF-DSIFT-E system achieves the best performance in terms of ST, and E-measure scores. Considering the values of NN, BOF method gets better performance than MR-BF-DSIFT-E and DMEVD. But if we base the evaluation on FT and DCG, DMEVD outperforms MR-BF-DSIFT-E. We also notice that our method performs the best for all scores. For instance, our 3D shape retrieval scheme leads to a ST score of 92.72%, which outperforms the one obtained by MR-BF-DSIFT-E algorithm (with a FT score of 90.92%). With E-measure, the gain on MR-BF-DSIFT-E method is of 12.71. Furthermore, compared to BOF method the gain in NN is of

1.31. Considering the precision-recall curves, Figure 5.5 clearly demonstrates that

Table 5.4: Similarity estimation scores on the SHREC 2010 data set.

Method	FT (%)	ST (%)	NN (%)	E-M (%)	DCG (%)
MR-BF-DSIFT-E [Furuya 2009]	98.5	90.92	96.32	70.55	97.63
DMEVD [Smeets 2010]	100	86.11	95.71	70.12	97.73
BOF [El Khoury 2012]	88	82.40	96.80	64.53	–
Our method	100	92.72	98.11	83.26	97.83

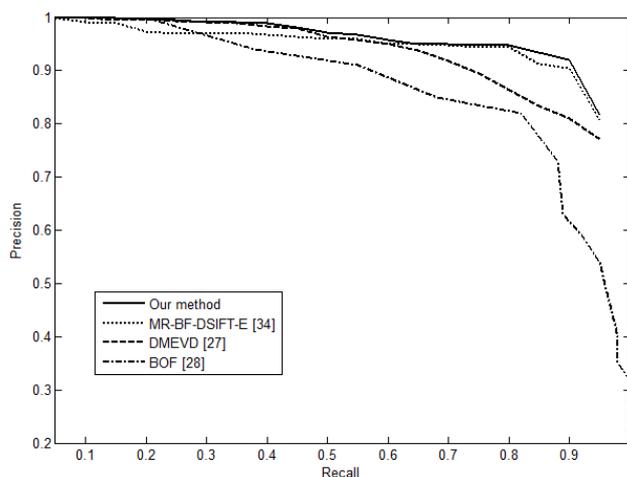


Figure 5.5: Precision-recall curves of the tested methods for the SHREC 2010 database.

our method outperforms MR-BF-DSIFT-E [Furuya 2009], DMEVD [Smeets 2010] and BOF [El Khoury 2012] algorithms. One may also notice that our retrieval scheme and DMEVD algorithm perform very similarly for low recall values. For high recall values, MR-BF-DSIFT-E owns a better precision compared to DMEVD [Smeets 2010] and BOF [El Khoury 2012] method, but our algorithm still possess higher precision. It is important to mention that in Figure 5.5, the recall values were bounded by 0.95. The main reason for doing so was to enable a direct comparison with DMEVD and MR-BF-DSIFT-E. Indeed, for both methods, we used the results which have been made publicly available on SHREC 2010 web site ⁶. From the results reported in this benchmark, the recall values are lower than 0.95.

We have also tested our method on the SHREC 2007 data set and compared it with RPU method developed by Tierny *et al.*'s [Tierny 2009]. Figure 5.6 shows two queries and their corresponding top 7 retrieved objects from SHREC 2007 database using our algorithm (a) and RPU method (b). As we can see from this

⁶<http://control.nist.gov/sharp/SHREC10/Non-Rigid/SHREC.html>

figure, the retrieved 3D objects in the top 7 positions of the rank lists belong to the same categories as their corresponding queries, which again verifies the effectiveness of our matching scheme in 3D shape retrieval applications.

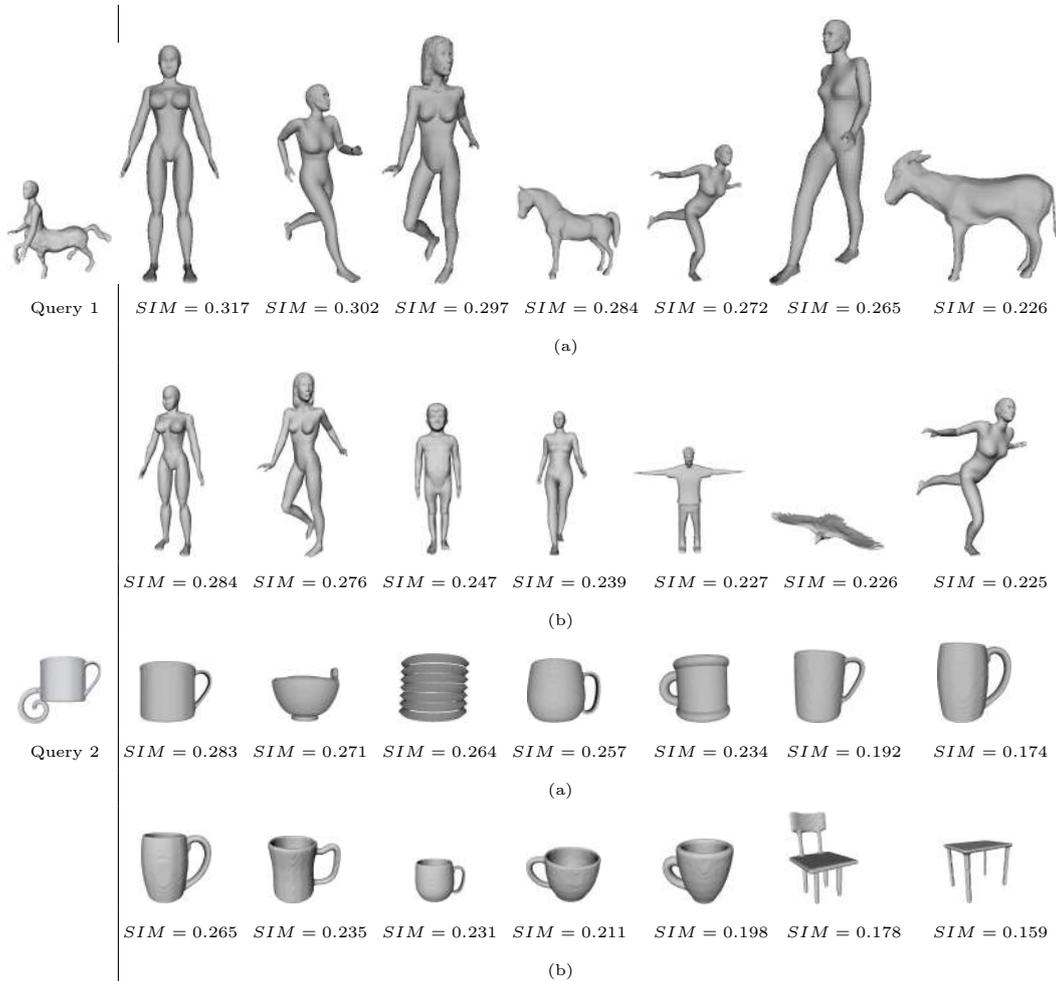


Figure 5.6: Examples of query objects from the SHREC 2007 query-set and the top-7 retrieved models.

From a more quantitative point of view, we compared the average Normalized Discounted Cumulative Gain NDCG vector of our approach and Tierny *et al.*'s competing method [Tierny 2009]. Figure 5.7 shows the curves corresponding to these NDCG vectors. From this figure, we can observe that the curve obtained by our method is situated above the two reference ones.

The last comparison has been made with two algorithms on the McGill database: the Bag of Words (BoW) based global shape retrieval algorithm from Lavoué [Lavoué 2011] and the graph-based approach (EMD-PPPT) from Agathos *et al.* [Agathos 2009]. We carried out evaluations on the average performance of the whole database, as well as the result corresponding to each specific class.

Table 5.5: Retrieval performance for the McGill database.

Class	method	NN (%)	FT (%)	ST (%)	DCG (%)
Whole	Our method	98.3	78.6	92.8	94.3
	BoW [Lavoué 2011]	94.5	62.9	77.6	88.1
	EMD-PPPT [Agathos 2009]	97.6	74.1	91.1	93.3
Ants	Our method	100	64.7	91.3	91.1
	BoW [Lavoué 2011]	96.7	58.3	86.7	89.1
	EMD-PPPT [Agathos 2009]	96.7	54.9	79.7	88.4
Crabs	Our method	100	94.7	99.4	99.7
	BoW [Lavoué 2011]	100	61.1	73.9	91.4
	EMD-PPPT [Agathos 2009]	100	98.2	99.8	99.9
Spectacles	Our method	100	86.4	93.3	96.4
	BoW [Lavoué 2011]	100	54.5	69.7	87.8
	EMD-PPPT [Agathos 2009]	95	83.9	88.9	95.2
Hands	Our method	100	93.1	96.6	97.6
	BoW [Lavoué 2011]	100	68.2	96.4	98.1
	EMD-PPPT [Agathos 2009]	96.6	93.5	88.8	93.4
Humans	Our method	91.6	100	83.2	89.7
	BoW [Lavoué 2011]	66	24.7	38.2	64.2
	EMD-PPPT [Agathos 2009]	88	58.8	81.8	88.1
Octopuses	Our method	100	97.8	99.3	99.8
	BoW [Lavoué 2011]	100	92.1	98.4	99.2
	EMD-PPPT [Agathos 2009]	100	100	100	100
Pliers	Our method	100	47.3	94.7	85.2
	BoW [Lavoué 2011]	88	20.8	25	64.3
	EMD-PPPT [Agathos 2009]	100	43.2	95.2	84.7
Snakes	Our method	100	94.1	100	99.2
	BoW [Lavoué 2011]	100	90.2	98.7	99
	EMD-PPPT [Agathos 2009]	100	70.3	99.8	94
Spiders	Our method	100	93.1	100	96.7
	BoW [Lavoué 2011]	100	71.6	96	93.6
	EMD-PPPT [Agathos 2009]	100	87.2	100	98.4
Teddy	Our method	100	97.1	100	99.7
	BoW [Lavoué 2011]	100	96.6	100	99.9
	EMD-PPPT [Agathos 2009]	100	45.3	63.2	83.9

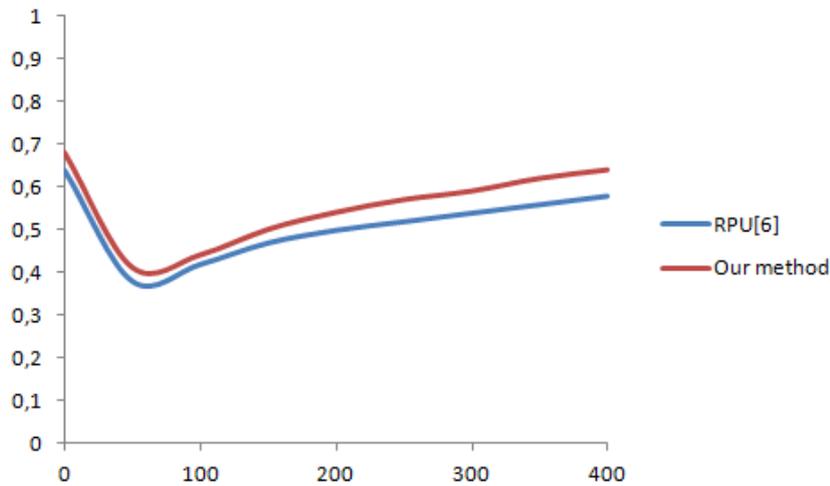


Figure 5.7: Average Normalized Discounted Cumulated Gain (NDCG) vectors for Our method and Reeb pattern unfolding RPU [Tierny 2009] on the SHREC 2007 data-set.

In Table 5.5 the four scores (*i.e.*, NN, FT, ST, and DCG) for each of the retrieval methodologies for each class of the McGill database as well as the average scores for the complete databases are shown. From the results provided in Table 5.5, we observe that our matching scheme performs better in total. One may also notice that our method has better results in most of the classes except for Crabs and Octopuses classes where the EMD-PPPT method slightly outperforms our technique. This may be explained by the fact that both models possess high skeletal articulation deformations and then, may be well categorized by the entire graph-based representation used in EMD-PPPT. For teddies class, BoW method slightly outperforms our technique if we base the evaluation on FT and DCG scores.

In Figure 5.8, precision-recall curves show the performance of EMD-PPPT retrieval algorithm against the proposed method for the whole McGill database. From the depicted curves it is shown that our retrieval scheme outperforms the EMD-PPPT scheme for all the recall values.

5.4.5 Computation times

In order to assess the time complexity of the proposed matching algorithm, execution-time tests were conducted employing executables generated by Visual Studio 9.0 operating in release mode. The source code was written in C++. All tests were conducted on a laptop with an Intel Core 3 CPU M350 at 2.23 GHz, and operating system Windows 7 SP 1. Table 5.6 presents the average processing time for the whole indexing of a model in each tested database. From the reported results, we can deduce that our matching system is computationally efficient. De-

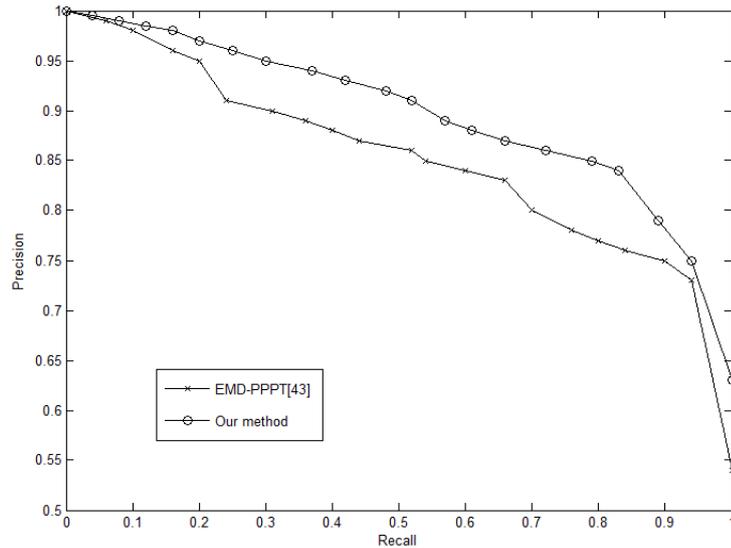


Figure 5.8: Precision-recall curves, of our method and the EMD-PPPT algorithm [Agathos 2009] for the McGill database.

pending on the number of vertices in the query model and the number of models in the database, the full processing time of a query varies from 11.2 to 17.3 seconds. For instance, for the SHREC 2010 data set where the average model size is 13.65K vertices, the whole indexing of a model takes an average of 11.2 seconds.

Table 5.6: Average execution times searching 3D models in McGill, SHREC 2010 and SHREC 2012 data sets.

Data set	Average model size (K vertices)	Average execution time (seconds)
McGill (255 objets)	13.59	17.3
SHREC2010 (200 objets)	13.65	11.2
SHREC2012 (1200 objets)	8.8	16.4

5.5 Conclusion

In this chapter, we presented a novel technique for content-based 3D model retrieval. Our contribution was to exploit the Reeb graph concept to define an efficient local shape descriptor. In our matching framework, the query mesh model is represented by its Reeb graph that is partitioned in Reeb charts. Each Reeb chart is associated with its pair of signatures which is encoded as an attribute in the similarity measure. A thorough experimental evaluation has shown that our method achieves excellent performance in terms of both accuracy and efficiency. Experimental as-

assessments also indicate that the proposed retrieval scheme exhibits substantial performance improvements over state-of-the-art algorithms on various shape retrieval benchmarks. We have to specify moreover that our algorithm is computationally efficient. Depending on the number of vertices in the query model, the average processing time of a query varies from 11.2 to 17.3 seconds.

Application to Segmentation-based compression scheme

Contents

6.1	Introduction	91
6.2	Survey on 3D dynamic compression	92
6.3	Proposed 3D segmentation-based compression scheme	95
6.3.1	Proposed Segmentation approach	96
6.3.2	Compression scheme	98
6.3.3	Rate control	99
6.4	Experimental results	101
6.4.1	Evaluation criteria	101
6.4.2	Performance results	103
6.5	Conclusion	109

6.1 Introduction

The main purpose of this chapter is to propose a new compression method based on motion estimation of successive frames. Intrinsic mesh surface segmentation has been studied in the field of computer vision, especially for compression and simplification purposes. Therefore we use a segmentation-based compression scheme for animated 3D meshes with constant connectivity. The proposed segmentation method exploits the temporal coherence of the geometry information by using the heat diffusion properties. The motion of the resulting regions is accurately described by 3D affine transforms. These transforms are computed according to the first frame to match the subsequent ones. In order to improve the performance of our encoder, the quantization of the temporal prediction errors is optimized by using a bit allocation procedure. The objective aimed at is to optimize the quantization of the mesh geometry by minimizing the reconstruction error. We will experimentally show that compared to the reference methods, the proposed coding scheme offers good compression performance. Furthermore, the rate control mechanism, allows supporting archiving and transmission needs.

6.2 Survey on 3D dynamic compression

The recent technological advances in the fields of telecommunication, computer graphics and multimedia allow access to an ever finer 3D dynamic modeling of the world. The critical challenges with 3D animated models lie in their rendering, storage and speed transmission over channels with limited bandwidth. In this context, the need for efficient compression techniques is crucial. A 3D dynamic model is commonly represented by a sequence of 3D triangular meshes, with constant connectivity and temporal information provided by time-varying geometry (only the vertices position changes over time).

Similar to static 3D mesh compression, spatial coherence can also be exploited in dynamic 3D mesh compression. As a result, each frame of the sequence, can be independently encoded using conventional static geometry and/or connectivity compression techniques [Guskov 2004a, K. Mamou 2006, Payan 2007, Cho 2010]. As an example, MPEG-4 Part-2 video coding standard has included a tool named 3D mesh compression (3DMC), for static meshes exploiting spatial dependencies of adjacent polygons.

Despite their efficiency, such static methods are still limited in terms of compression ratio since they do not exploit the dependencies across the temporal scale. To increase the compression performances, it is important to exploit statistical redundancies in both spatial and temporal directions. Indeed, it is desirable to use inter-frame coding by employing temporal prediction. In [Lengyel 1999], Lengyel proposed to model the temporal mesh deformation by special animation parameters, corresponding to affine motion. The main idea consists in splitting a mesh into several sub-meshes, and estimating a rigid-body motion for each sub-mesh. The difference between the real mesh deformation and the estimated one is then coded and saved.

Other compression systems based on predictive coding have been developed in order to exploit the temporal and spatial correlations of mesh sequences [Ahn 2011, Yang 2002, Ibarria 2003]. The main idea of all these methods is to predict the displacements of the vertices along the sequence and then to encode the residual errors.

In [Alexa 2000, Karni 2004, Lee 2007], the authors introduced three coding methods, for animated 3D meshes, based on the principal of component analysis (PCA). These techniques exploit only the temporal coherence of the geometry component instead of focusing on both spatial and temporal coherence. Another alternative developed by [Briceno 2003] proposed to project each frame of the sequence onto a 2D image, and then compress the resulting sequence of bi-dimensional images with some conventional video coding techniques. In this coding method, sophisticated mesh cutting needs to be performed to find a suitable mesh-to-image

mapping.

Similar to the motion compensated temporal filtering (MCTF) based video coding, relevant work were conducted recently on temporal wavelet-based coders for animated 3D meshes [Guskov 2004a, Yang 2004, Payan 2007].

The developed coders focus on an analysis step that transforms the geometry information to reduce the signal amount of data, and then a differential coding strategy is applied on the resulting details to exploit the temporal dependencies between successive frames.

The first animated compression algorithm, proposed by Lengyel, is based on the affine transformations [Lengyel 1999]. A mesh is partitioned into different regions, and a rigid-body motion is computed for each region. Doing so, the deformation in a region is only represented by a set of affine transformations, instead of the displacements of all the vertices in the same region. Later, [Shamir 2001] have extended Lengyel's coding scheme by introducing a multi-resolution coding approach for temporal deformation. Affine transformations have also been used by [K. Mamou 2006]. The authors proposed a 3D mesh compression scheme, based on a skinning animation technique. The segmentation process is based on 3D affine transforms in order to obtain the frame-wise motion of each region by weighting previous affine transforms. Motion compensation is then performed followed by Discrete Cosine Transform (DCT) of residual errors.

Other compression techniques, of dynamic 3D meshes, have been developed [Guskov 2004b, K. Mamou 2006, Payan 2007, Cho 2010]. The overall idea of all these methods is to reduce temporal redundancy by estimating the temporal coherence between rigid meshes."

A relevant compression method, proposed by [Alexa 2000], performs the PCA of geometry covariance matrix to reduce spatial correlation. [Karni 2004] have extended this technique by applying a second-order linear predictive coding (LPC) on PCA components, to exploit the temporal coherence. This algorithm has been shown to be efficient only for sequences with few global motion. In addition, its is penalized by its high time and memory complexity. More recently, [Lee 2007] proposed to improve Alexa and Müller's compression algorithm by optimizing the number of key-frames. The drawback of these methods is that the number of key-frames may be quite high. Consequently, their effectiveness is diminished when applied to dense meshes with high number of vertices.

In [Müller 2006], authors introduced a dynamic 3D mesh coder based on rate-distortion optimization. Both of the spatial partition and the prediction mode are determined using the Lagrangian cost function. A rate-distortion optimization model has been also used by [Payan 2007]. To exploit temporal coherence, the

authors proposed to use a temporal wavelet filtering. During the encoding step, the quantization of the wavelet coefficients is optimized by a temporal model based on bit allocation mechanism. In the same context, [Cho 2010] proposed a wavelet based compression scheme, for 3D mesh sequences, using integer analysis and synthesis filter bank.

Recently, Bici *et al.* [Bici 2011] proposed three predictive coding approaches based on weighted spatial prediction. The authors introduced a weighted spatial prediction scheme in the first contribution. Then, in the second one, they integrated a refinement step. Finally, they introduced an angle based predictor. The proposed structures achieve a significant improvement in the prediction error accuracy and the compression rate. Particularly, Bici *et al.*'s coding method [Bici 2011] is suitable for low-delay streaming scenarios.

More Recently, [Váša 2014] proposed a compression system, for dynamic 3D meshes that exploits geometric laplacian. The main contribution consists in calculating, for the entire sequence, an average mesh. The latter is used, by applying a discrete geometric Laplacian, to encode the coefficients describing the mesh vertex trajectories, using a static mesh compression method. In order to improve the compression performance, the authors have integrated a spatio-temporal predictor.

The issue of 3D dynamic mesh compression has rapidly gained the interest of the scientific community in recent years. However few existing work have been investigated on motion-based segmentation for dynamic 3D mesh compression [Lengyel 1999, Amjoun 2006, Boulfani-Cuisinaud 2007].

Although various researches have been conducted in the area of animation compression, however, few work have been done regarding the segmentation based motion estimation of dynamic 3D meshes [Lengyel 1999, Amjoun 2006, Boulfani-Cuisinaud 2007].

The segmentation algorithm developed by [Lengyel 1999] partitions the mesh into sub-meshes, whose motion can be accurately described by 3D affine transforms. Lengyel's segmentation method is based on heuristic approach that selects randomly 10% of triangles to represent the motion information. The remaining vertices are classified according to the motion of these triangles. Lengyel's method has been shown to be quite effective. However, its is hindered by the use of large number of clusters independently from the motion nature. [Amjoun 2006] have extended this approach, to develop region growing segmentation algorithm. The main idea was to search the N farthest triangles in geodesic sense. The choice of triangle germs is based only on the geometry of the first frame in the mesh sequence, causing a lack of segmentation efficiency. [Boulfani-Cuisinaud 2007] proposed another alternative which represents each vertex by an affine transform that optimally describes the motion of its neighborhood. The vertices whose motions are described by the same

affine transform, with respect to a minimum prediction error, are grouped in the same cluster. Due to the sub-optimal selection, the vertices located on the cluster borders are misclassified, which in turns affects the segmentation accuracy. More Recently [Rosman 2012, Rosman 2013] developed a motion-based segmentation technique that partitions an articulated 3D shape into rigid parts by resolving a piecewise-smooth regularization problem [Mumford 1989].

In the rest of this chapter, we propose a segmentation based compression scheme. Our contributions are twofold. First, we introduce a novel segmentation scheme, based upon ideas from Morse theory, to partition the first mesh of the sequence into sub-meshes having independent deformations. The best affine transformations, that represent the displacements of the sub-meshes over the subsequent frames, are then computed and encoded. Second, we propose to increase the compression performance by using a bit allocation mechanism that optimizes the selection of quantizer step sizes to be applied on the affine transform coefficients.

6.3 Proposed 3D segmentation-based compression scheme

This section describes our coding system, which is illustrated in the block diagram of Fig. 6.1. We assume that the input data is a sequence of triangular meshes (frames) sharing the same connectivity. In what follows, we present the overall compression scheme, then we introduce the segmentation process, and finally, we describe the bit allocation strategy, which is based on rate-distortion models.

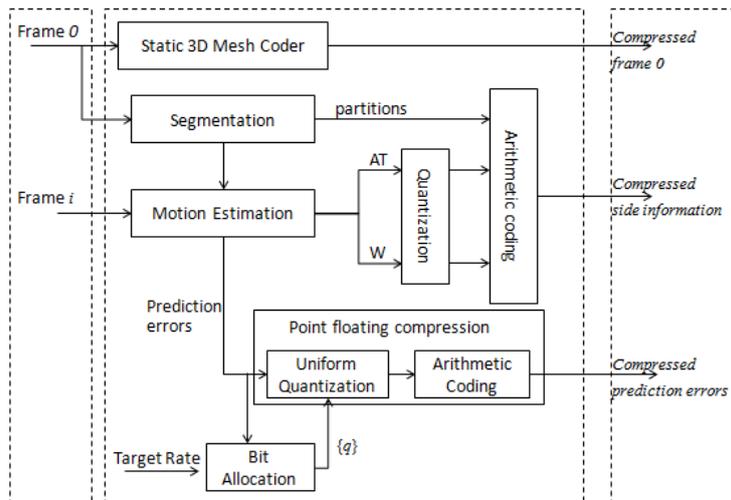


Figure 6.1: Block diagram of the proposed coding scheme.

6.3.1 Proposed Segmentation approach

In this work we propose an implicit segmentation method which exploits the temporal information of dynamic 3D meshes. The main contribution consists in partitioning dynamic meshes into segments of homogeneous properties, based on kinematic Reeb graph (KRG) representation presented in Chapter 3. Particularly, a new scalar function, based on the eccentricity in term of diffusion distance, has been created to compute the Reeb graphs.

Curvature information is exploited to refine boundaries between object parts in accordance to the minima rule. Thus, the boundaries of the regions automatically follow the most concave parts of the surface. Vertices are distributed into regions according to the value of the continuous function μ computed in section 6.3. Contiguous regions with the same number of connected components are merged into a single interval. Each interval is characterized by its scalar function value and the number of its connected components. This process is repeated iteratively in order to reduce the number of regions (clusters). The iterative loop stops when each region admits a different number of connected components. Consequently, each vertex of the mesh is marked based on the region to which it belongs. Its worth mentioning that the obtained partition of vertices, located on the boundary regions, may be slightly corrected by exploiting the curvature information. The aforementioned segmentation approach is summarized in Algorithm 4.

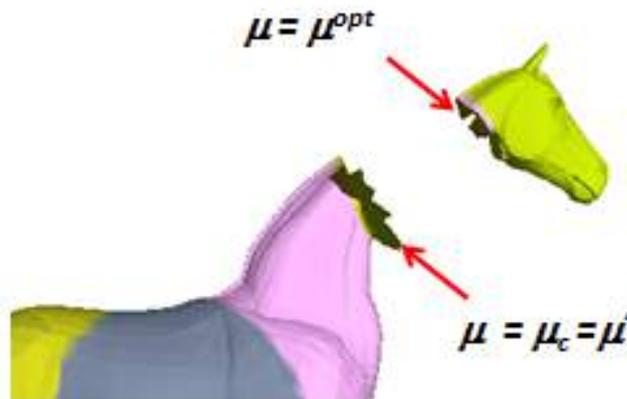


Figure 6.2: Match region boundaries with deep surface concavities.

A refinement step has been integrated in our segmentation scheme in order to improve the mesh decomposition. This is done by adjusting the region boundaries according to curvature information.

Each region boundary is considered as a level set and is thus associated with a value of μ that corresponds to a critical point. In our case, the proposed μ func-

Algorithm 4 3D mesh segmentation algorithm.

Require: a discrete triangular surface S .

Ensure: initial partition $P(S)$.

- 1: Interval collection I
- 2: Number of Tuples k in $I \leftarrow$ number of vertices in M
- 3: **for all** $v_k \in M$ **do**
- 4: Compute the scalar function $\mu(v_k)$
- 5: Number of connected components $N(\mu(v_k))$ associated with $\mu(v_k)$
- 6: $I_k \leftarrow (\mu(v_k), N(\mu(v_k)))$
- 7: **end for**
- 8: Sort I according to the increasing order of $\mu(v)$
- 9: **while** \exists adjacent intervals have the same $N(\mu(v))$ **do**
- 10: **for all** $I_k \in I$ **do**
- 11: find the interval I_j adjacent to I_k
- 12: **if** $N(\mu(v_k)) = N(\mu(v_j))$ **then**
- 13: merge the two intervals I_j and I_k together
- 14: **end if**
- 15: Update(I)
- 16: **end for**
- 17: **end while**
- 18: **for all** $v_k \in M$ **do**
- 19: v_k is labeled based on the interval it maps to
- 20: **end for**
- 21: Update $P(S)$

tion is stable against perturbations and invariant under isometric transformations. Thus, we have overcome the locality problem. During the refinement step, to define a perceptually salient decomposition, we only consider the concavity problem. The optimal value of μ_{opt} should determine a boundary that matches a deep concavity profile on the object surface. Implicitly, μ_{opt} is close to μ_c that corresponds to the closest critical point as shown in Fig. 6.2.

The objective aimed at is to find the optimal value μ_{opt} that determines the boundary profile. The issue can be considered as an optimization problem, which consists in minimizing the concavity function $E_{concave}(\mu)$ of each region boundary associated with a value of μ . $E_{concave}(\mu)$ is defined by:

$$E_{concave}(\mu) = \min_{\mu'} (K_{min}(c_{(\mu),R}(t)) \otimes G_{\sigma}(t)), \quad (6.1)$$

with $K_{min}(\cdot)$ being a function returning the sequence of K_{min} curvature values, computed according to [Taubin 1995a] along the boundary profile, and $c_{(\mu),R}(t)$ the curve-parameterized with respect to the normalized arc-length t . $c_{(\mu),R}(t)$ represents the set of μ values corresponding to the boundary of region R . The convolution with a Gaussian kernel $G_{\sigma}(t)$ leads to smoothing values of K_{min} . Consequently, the minimum identification will be more efficient and stable. In addition, curvature information is exploited to refine the segmentation and adjust region boundaries in order to match deep surface concavities.

6.3.2 Compression scheme

As illustrated in Fig. 6.1, our approach consists in representing the geometry of the mesh sequence by a piecewise affine geometry predictor minimizing the prediction errors. This is accomplished by exploiting the partition obtained by a segmentation algorithm, which is applied only on the reference frame. The obtained key-vector $\pi = \{1, \dots, V\}$ associates each of the mesh vertices to the index k of the cluster to which it belongs. This key-vector is coded using the lossless arithmetic coder. The connectivity information is coded only once, together with the geometry information of the reference frame. The latter is compressed using a static 3D mesh encoder. We privilege using the progressive lossless mesh encoder of [Valette 2009], which is based on Incremental Parametric Refinement (IPR). In our setting, the vertex coordinates are quantized to 12 bits per coordinate. Experiments have shown that the IPR method provides very competitive results compared to previous work in terms of rate/distortion trade-off [Valette 2009].

During the prediction procedure, the first mesh of the sequence is taken as a reference frame. The motion estimation consists in describing the affine motion of clusters $k \in \{1, \dots, K\}$ at frames $i \in \{1, \dots, F\}$ by an affine transforms AT_i^k . The latter is computed as follow with respect to the reference frame:

$$AT_i^k = \arg \min_A \left(\sum_{v \in \pi} \|A_{\chi_v^0} - \chi_1^v\|^2 \right), \quad (6.2)$$

where A is 4×4 matrix representing an affine transform, and χ_i^v is a 4D vector that designates the homogeneous coordinates of the vertex v at frame i . The piecewise affine predictor of the frame i from the reference frame (frame 0) is expressed as follows:

$$\hat{\chi}_i^v = \sum_{k=1}^K W_k^v AT_i^k \chi_0^v, \quad (6.3)$$

where W_k^v , defined in (6.3), is the optimal weight vector that controls the motion influence of all the clusters $k \in \{1, \dots, K\}$ over the vertex v . The motion of the vertices in each cluster are obtained by weighting the corresponding affine transforms. The homogeneous estimated coordinates of the vertex v at frame i are provided from the set AT_i^k and W_k^v :

$$W_k^v = \arg \min_A \sum_{i=0}^{F-1} \left\| \sum_{k=1}^K AT_i^k \chi_0^v - \chi_i^v \right\|. \quad (6.4)$$

The two matrices AT_i^k and W_k^v are quantized and encoded by using the entropy arithmetic coder. It is worth noting that these two matrices represent the motion information which highly influences the motion compensation accuracy, and consequently the reconstruction quality. For this reason, we proposed to use a rate control mechanism to efficiently calculate the quantizer step sizes during the quantization stage. The proposed bit allocation procedure is reviewed in Section 6.3.3 in more detail.

The residual error is calculated as a simple difference between the original 4D vector, representing the homogeneous coordinates of the vertex v at frame i , and the estimated one. We note that the prediction error vector, relative to each vertex v , is represented by floating point components with low-magnitudes (that tend to zero). We proposed using the normalized scientific notation of the floating point components, to encode separately the signs, exponents and mantissas of the original vertex and the estimated one. This process allows realizing lossless compression with high precision.

The overall compressed file is composed of two types of data:

- inside information, which contains the partition, affine transforms and animation weights,
- outside information, that comprises the geometry and the connectivity of the reference frame as well as the prediction errors.

6.3.3 Rate control

The rate control (or the rate-distortion optimization) mechanism is the core of our compression scheme. For a target bit-rate R_{budget} , our rate control process allows to minimize the distortion of the final code-stream produced by the coding system.

As cited earlier, this process is performed on the prediction error data, which is split into 3 subbands corresponding to the prediction errors of the 3 coordinates x , y , and z . The rate-distortion trade-off is determined by calculating the adequate quantizer step size, for each subband, in order to attain the target bit-rate. For a given subband S_i and a quantizer step size q , $D_i(q)$, the contribution to total distortion from subband S_i , is defined by the mean square error, given by:

$$D_i(q) = \frac{1}{3} \text{Mean}\{(S_i - ((S_i/q) \times q))^2\}, \quad (6.5)$$

where the mean is taken over all the coefficients in S_i , and “/” denotes division followed by rounding to the nearest integer. Similarly, define $R_i(q)$, the bit-rate contribution as:

$$R_i(q) = \frac{1}{3} \text{Entropy}\{(S_i/q)\}, \quad (6.6)$$

where the entropy is calculated over all the components in S_i using statistical models. Inspired by the theoretical models proposed in [Payan 2005] and from results provided in appendix .1, we assume that the probability density functions of the prediction error coordinates can be modeled by Generalized Gaussian Distributions (*GGD*). More details about the used distribution modeling approach are provided in appendix .1.

It is worthy to mention that the side information is encoded without loss. Consequently, it does not intervene in the quality degradation. The total distortion is then calculated by considering only the distortion due to the quantization of the three coordinate subbands. On the other hand, the total bit-rate is calculated using both of side and outside information.

Considering that the total distortion is an additive metric, calculated as $D = \sum_{i=1}^3 D_i$,

and that the total bit-rate of the code-stream is given by $R = \sum_{i=1}^3 R_i$, the rate-distortion problem can be formulated as follows. Given an input 3 subbands with a target bit-rate R_{budget} , one wants to select the set of quantization step sizes $Q = \{q_i : i = 1, \dots, 3\}$ to minimize the total distortion D :

$$D(Q) = \sum_{i=1}^3 D_i(q_i), \quad (6.7)$$

subject to the bit-rate constraint:

$$R(Q) = \sum_{i=1}^3 R_i(q_i) \leq R_{budget} - R_s - R_{ff}, \quad (6.8)$$

with R_s and R_{ff} are the bit-rates of the side information and the reference frame, respectively. Using Lagrange multiplier, equations 6.7 and 6.8 are equivalent to the

following unconstrained problem:

$$\min_Q J(\lambda) = D(Q) + \lambda((R(Q) + R_s + R_{ff}) - R_{budget}), \quad (6.9)$$

where λ being the Lagrangian multiplier, and $J(\lambda)$ is the Lagrangian cost. For a fixed λ , $J(\lambda)$ is minimized when

$$\frac{\partial J(\lambda)}{\partial D(Q)} = 0 \quad (6.10)$$

$$\frac{\partial J(\lambda)}{\partial R(Q)} = 0. \quad (6.11)$$

It is important to note that for a given Lagrange multiplier, the resulted λ and $J(\lambda)$ might not meet the overall rate constraint. Therefore, we should find the optimal Lagrange multiplier λ^{opt} such that the total bit-rate would be equal to R_{budget} . In our work, λ^{opt} is obtained by using the bisection method described in [Krongold 1998].

Fig. 6.3 shows the obtained rate/distortion curve for a set of bit-rates. This figure represents an instance of the problem given by eq. 6.9 where the bit-rate has to be allocated in order to correspond to the target bit-rate. Empty circles in Fig. 6.3 represent the points that are located over the convex hull. These points should be rearranged in order to be located on the convex hull (red circles). Minimizing the Lagrangian cost $J(\lambda)$, for $\lambda = 0$, is analogous to minimizing the distortion, i.e, finding the point closest to the distortion-axis in Fig. 6.3. On the contrary, minimizing the Lagrangian cost $J(\lambda)$, for high values of λ , is equivalent to minimizing the bit-rate and thus selecting the point closest to the rate-axis in Fig. 2.1.

6.4 Experimental results

In order to evaluate the proposed segmentation approach, we consider some 3D dynamic meshes named: Dance, Chicken, Cow, and Snake. These models are characterized by their various motions and complexities. Moreover, they offer a good variability in terms of spatial and temporal sizes. Table 6.1 summarizes their properties, expressed in terms of numbers of vertices, number of frames, and number of connected components.

6.4.1 Evaluation criteria

To evaluate the performance of the proposed segmentation method, we choose the mean square error, introduced by the motion compensation procedure. The objective aimed at is to obtain a partition $\Pi = (\pi_k)_{k \in 1, \dots, K}$ of the whole mesh into K

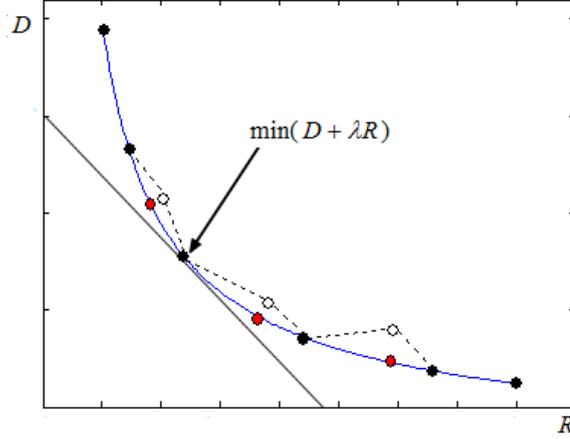


Figure 6.3: Rate/distortion characteristic using Lagrangian optimization.

Table 6.1: Properties of the tested dynamic meshes.

	Vertices	Frames	Connected Components
Cow	2904	204	1
Dance	7061	201	1
Chicken	3030	400	41
Snake	9179	134	1

regions. The mean square error denoted by $E(\Pi)$ is defined by:

$$E(\Pi) = \frac{1}{V \times T \times D^2} \sum_{i=1}^T \sum_{k=1}^K \sum_{v \in \pi_k} \|\chi_i^v - A_i^k \chi_1^v\|^2,$$

where V and T denote the number of vertices and the number of frames of the mesh sequences, respectively. D is the bounding box diagonal of the first frame. A_i^k is the 3D affine transform associated with the region (π_k) at frame i , and χ_i^v is a vector that consists of the homogeneous coordinates of the vertex v at frame i .

To assess the performances of our proposed compression scheme, we evaluate the introduced distortion as a function of the attained bit-rates. The quality degradation is assessed using two error metrics:

- The KG error introduced by [Karni 2004]. The latter, expressed in percent, corresponds to the relative discrete L2-norm both in time and space. It is defined by:

$$KG = 100 \frac{\|G - \hat{G}\|}{\|G - E(G)\|},$$

where G and \hat{G} are two matrices containing the (3-D) coordinates of the original sequence and the reconstructed one, respectively. $E(G)$ is an average matrix containing the mean values of the coordinate sets of each frame.

- The root mean square error RMSE between original and decoded meshes. The RMSE distortion is defined as the mean value of the frame to frame, computed over all the frames of the sequence. This metric is computed by the METRO tool [Cignoni 1998].

The bit-rates are expressed in terms of bits per vertex per frame (bpvf). The reported bit-rates include the rate needed to encode both of side and outside information.

6.4.2 Performance results

In this section we start by analyzing the accuracy of the proposed segmentation method. Then, we evaluate the compression performance of our coding scheme.

6.4.2.1 Segmentation accuracy

Fig. 6.4 depicts some segmentation results for Dance and Snake sequences. From this figure we can see that the segmentation process allows to partition the mesh into rigid clusters consisting of topologically connected vertices which are characterized by similar motion properties.

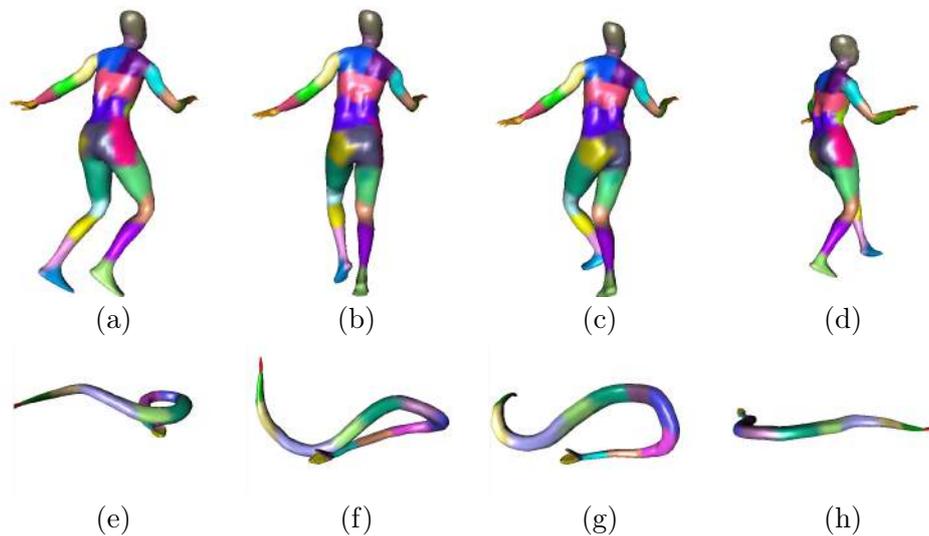


Figure 6.4: Segmentation results on four selected frames, extracted from (a-b-c-d) Dance, and (e-f-g-h) Snake sequences.

The results reported in Table 6.2 present the values of the squared error $E(\cdot)$ before and after the refinement step. The number of iterations has also been provided in order to evaluate the convergence rate. From these results, we can notice

Table 6.2: Evaluation of the mean square error of the motion compensation $E(\Pi)$.

	$E(\Pi_0)$	Nb iterations	$E(\Pi)$
dance	0.0023	19	0.0012
Cow	0.0028	16	0.0016
Snake	0.0024	25	0.0014

that, on average, the refinement step converges after 20 iterations. The motion estimation error obtained using the refinement step is about 0.0014 against 0.0025 without. Thus we conclude that, the refinement post-processing stage allows to increase, significantly, the motion estimation accuracy.

6.4.2.2 Compression performance

In order to assess the performance of our compression scheme, we conduct some comparisons with previous methods from the state-of-the-art. For a fair comparison, we divided our tests on two sets according the used distortion metric. In the first set of comparisons, we retain the KG error as a quality metric. To perform the comparisons, WSP[Bici 2011] and FAMC[Mamou 2008, N. Stefanoski 2008] algorithms are used as references:

- The Weighted spatial prediction (WSP)[Bici 2011] algorithm, described earlier in Section 6.2, integrates three prediction structures,
- Frame-based Animated Mesh Compression (MPEG-4 FAMC)[Mamou 2008, N. Stefanoski 2008], based on skinning approach [K. Mamou 2006] and Context-Adaptive Binary Arithmetic coding [Marpe 2003].

In the second set of comparisons, the RMSE is used as distortion metric, whereas RT[Collins 2005], D3DMC[Müeller 2005], GV[Briceno 2003], and skinning algorithms are taken as references:

- The RT clustering-based approach [Collins 2005], consists in splitting the mesh into sub-parts whose motion is expressed only in terms of Rigid Transforms (RT). The object's motion is described by a set of rigid motion parameters associated with each cluster.
- Dynamic 3D Mesh Coder D3DMC [Müeller 2005] is a clustering-based method, where the motion field is described by a set of motion vectors represented by an octree structure.
- The Geometry Video (GV) [Briceno 2003] algorithm applies a global affine motion compensation procedure. It uses a stretch minimizing parametrization and a conventional video encoding approach to encode a geometry image sequences. The latter is obtained by applying a uniform sampling on the parametric domain.

- The skinning [K. Mamou 2006] algorithm, described earlier in Section 6.2, is a piecewise affine predictor coupled with a DCT representation of the prediction errors.

Fig. 6.5 illustrates the rate/distortion curves obtained by using WSP [Bici 2011], FAMC [Mamou 2008, N. Stefanoski 2008], and our method for Cow, Chicken, and Dance sequences. From Fig. 6.5 (a)-(c), we can see that, on average, our coding scheme systematically yield superior compression performance when compared to WSP and FAMC reference codecs. Furthermore, the rate/distortion curves show that the FAMC offers the worst results, except for Chicken sequence, which seems to be better compressed by FAMC.

Fig. 6.6(a) illustrates the rate/distortion results obtained by D3DMC [Müller 2005], skinning [K. Mamou 2006] and our method for Chicken sequence. From this figure, we can notice that D3DMC codec leads to the best performances. For all the tested bit-rates, our codec slightly outperforms the skinning algorithm. The gap between the bit-rate-distortion curves representing our method and the skinning algorithm may be explained by the fact, that our segmentation approach exploits the temporal coherence of the geometry component based on heat diffusion properties. Additionally, the accuracy of the vertices distribution on the border between clusters has been enhanced by exploiting the curvature information.

Fig. 6.6(b) depicts the plots of RMSE variation as a function of bit-rates for Snake sequence. The RMSE curves in 6.6(b) clearly show that the proposed codec outperforms the state-of-the-art for low bit-rates. Specifically, at 2.7 bpvf, our codec achieves around 70% lower distortion than RT method.

The compression results of our codec, GV, and skinning for Dance sequence are illustrated in Fig. 6.6(c). When examining the figure in its whole, it very clear that the proposed coding scheme surpasses the state-of-the-art at low bit-rates (less than 3 bpvf).

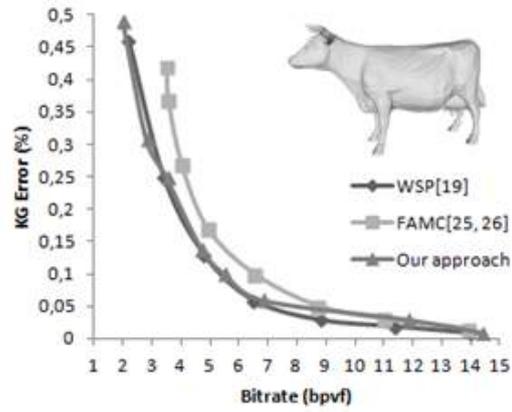
Finally in Fig. 6.7, we provide the RMSE distortion values individually for each frame of the tested sequence Cow and Snake. In our simulation we fixed the number of clusters K to 30. Examining the results shown in Fig. 6.7(a) and (b), we notice that for Cow sequence, our codec yields a very low distortion spatially for bit-rate of 7 bpvf. For Snake sequence, the lowest RMSE values attained at bit-rate of 4 bpvf. Specifically, the RMSE varies in the range of $[6.8 \times (10^{-5}), 9.4 \times (10^{-5})]$ and $[4.6 \times (10^{-5}), 13.7 \times (10^{-5})]$ for the Cow and Snake models, respectively.

Fig. 6.8 shows five key-frames, extracted from Cow model, coded on different bitrates. From this figure, we can clearly observe that our codec provides superior visual quality even at low bit-rates (2 bpvf).

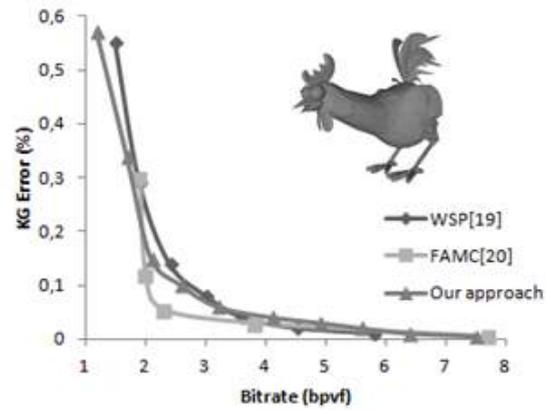
6.4.2.3 Complexity evaluation

In order to assess the time complexity of the proposed coding scheme, execution-time tests were conducted employing executables generated by Visual Studio 9.0

(a)



(b)



(c)

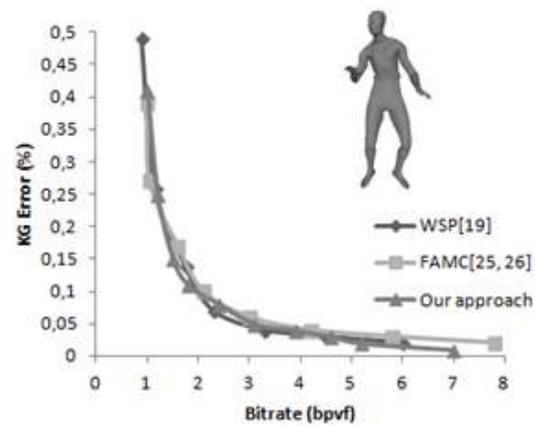


Figure 6.5: Rate/distortion performances for Cow (a), Chicken (b) and Dance (c) sequences.

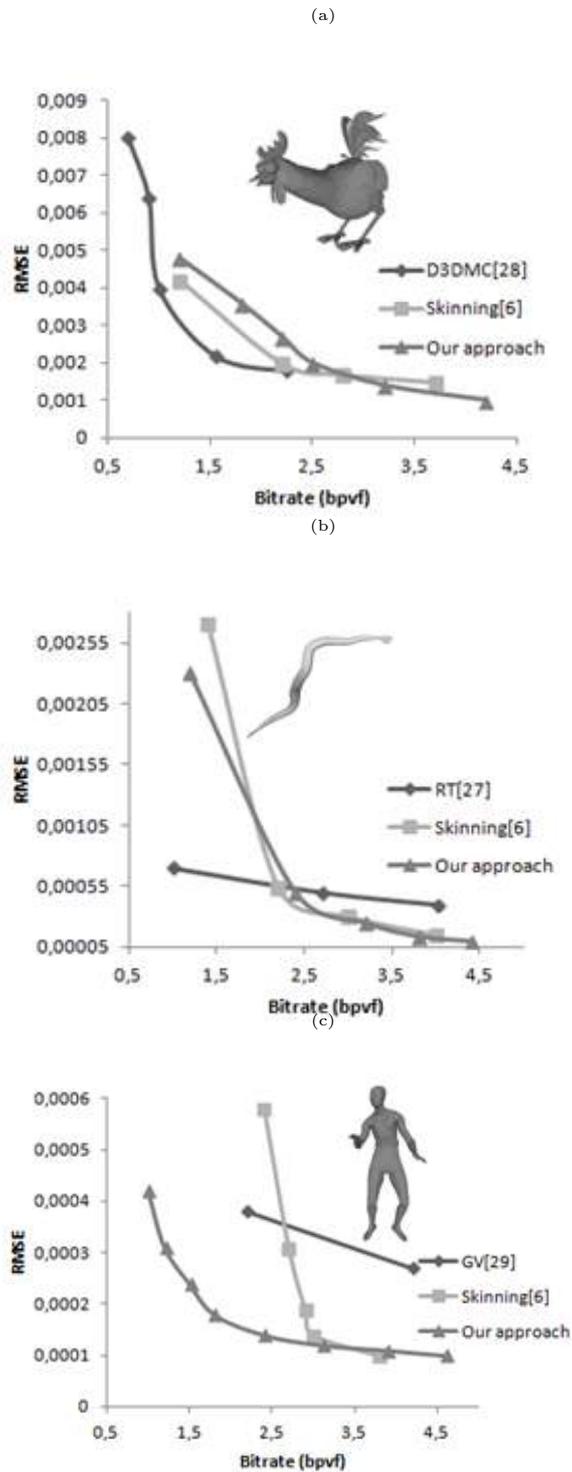


Figure 6.6: Rate/distortion performances for Chicken (a), Snake (b) and Dance(c) sequences.

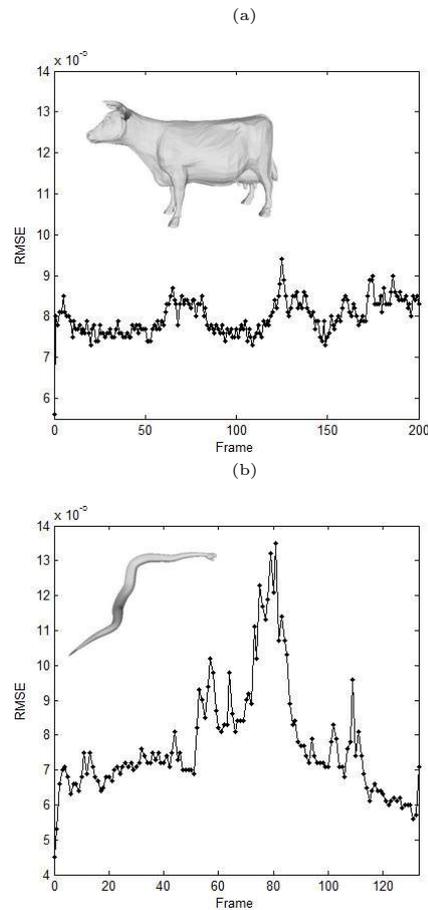


Figure 6.7: RMSE as a function of the frame index for cow and snake sequences.

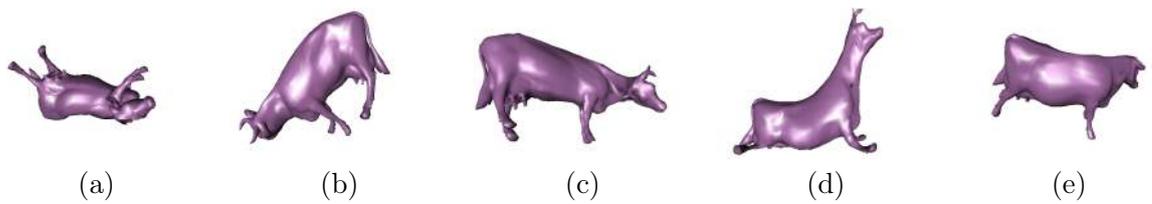


Figure 6.8: Key-frames extracted from the Cow sequence, (a) frame decoded at $2bpvf$, (b) frame decoded at $3.5bpvf$, (c) frame decoded at $4.7bpvf$, (d) frame decoded at $5.5bpvf$, (e) frame decoded at $6.8bpvf$.

operating in release mode. The source code was written in C++. The tests were conducted on a laptop with an Intel Core 7- 4720HQ at 2.6 GHz, and operating system Windows 8.1. Table 6.3 illustrate the processing time for each test model. From the reported results, we can deduce that our coding scheme is computationally efficient. Depending on the number of vertices in the 3D mesh and the number of frames in the sequence, the full processing time varies from 726 to 9823 *ms*. Note that it was not possible to compare the processing time of our method with the

FAMC [Mamou 2008] and WSP [Bici 2011] algorithms, since its authors do not publish any timings.

Table 6.3: Average execution times for Cow, Dance and Chicken models.

	Average execution time (ms)
Cow	726
Dance	9823
Chicken	3712

6.4.2.4 Discussions

To reduce the motion estimation error, our coder uses a faithful segmentation technique based on heat diffusion properties. Also the accuracy of the vertices distribution, on the border between clusters, has been enhanced by exploiting the curvature information. We notice that the refinement post processing stage used by our segmentation algorithm converges on average 20 iterations. This refinement process allows to reduce the motion estimation error. All this reasons justify the good results, for our coder, obtained in terms of distortion when comparing our method to the reference one.

Additionally, computing a piecewise affine predictor allows to minimize the temporal prediction errors. Their quantification is subsequently optimized using the bit allocation strategy, which allows to obtain the best results in terms of bitrates and introduced distortion.

The final code-stream produced by our coding system includes the compressed reference frame, the partition, the affine transforms and the animation weights matrices and the set of prediction errors separately for each frame. All of this compressed data penalized the coding efficiency of our method at low bitrates..

6.5 Conclusion

In this chapter, we presented a hybrid coding system adapted to dynamic 3D meshes. In order to perform accurately the motion estimation we integrated a segmentation process that allows to partition the 3D model according to the heat diffusion properties by exploiting the temporal and curvature information. The obtained partition is used in order to compute a piecewise affine predictor which minimizes the prediction errors. The rate/distorsion performance of our encoder is improved by optimizing the quantization of the temporal prediction errors using a rate control mechanism. Preliminary experimental results have shown that our approach leads to a satisfactory performance. Compared to the state of the art, our compression results are very promising.

Conclusion

Contents

7.1	Summary of contribution	111
7.2	Open problems and Perspectives	113
.1	Appendix A	115

7.1 Summary of contribution

In this thesis, we have presented our research study on topology modeling of 3D object based on Reeb graph representation. As part of our work, our main contribution is to define a new function based on the continuous heat diffusion properties. This Reeb graph construction approach can be extremely helpful as a local shape descriptor for recognition of 3D shape. It can also be introduced into a dynamic compression system based on the segmentation. In this context, we exploit Reeb graphs concept into two applications that are widely used: pattern recognition and 3D dynamic segmentation-compression.

First, we highlight in chapter 2 the notion of 3D shape its field applications and creations. Furthermore we review the modeling of three-dimensional objects. In particular, we focused on 3D triangular meshes that are frequently used to represent 3D objects. Despite its simplicity of use, this representation suffers from its sensitivity against topological, affine or isometric transformation. This motivates research for intrinsic shape modeling techniques before processing this kind of 3D data.

In chapter 3, we introduce the notion of intrinsic shape modeling. In particular we describe two categories: geometry and topology based modeling. Finally, we focused on topological modeling existing work and specially Reeb graph based representation which is the core of our research. Theoretically Reeb graph representation appears an an interesting object abstraction. It captures efficiently the topology of the object which leads to a complete topology control for more geometry modeling.

The contributions of this thesis can be summarized as follows.

Reeb Graph extraction based on Heat Diffusion Properties

In chapter 4, we proposed a new Reeb graph construction algorithm adapted for 3D dynamic meshes. The main contribution consists in defining a new continuous function based on Morse Theory. The latter is calculated as the eccentricity in term

of diffusion distance from a given point on the surface to feature points located on the extremity of the 3D shape. Feature points are first detected using the heat diffusion notion. They are calculated from the two farthest points in the geodesic sense. In each one we defined a function based on the diffusion distance in order to extract two local proprieties groups. The intersection of these two groups provides the set of feature points. This approach produces a set of well-localized points which are stable against topological changes. The restriction of the heat kernel to the time domain makes the proposed scalar function intrinsic and stable against disturbances. The experimental results on dynamic 3D models demonstrated the robustness and effectiveness of the proposed scalar function.

Application to non-rigid 3D shape retrieval In chapter 5, we proposed to exploit the Reeb graph construction as a local descriptor in a non-rigid 3D pattern recognition system. The objective is to segment the Reeb graph into a set of Reeb charts of controlled topology. Each one is projected to the canonical planar domain which have either disk or annulus topology. This unfolding in the canonical planar domain introduces area and angle distortions. Based on that, a couple of signatures is calculated, which will be used later for matching pairs of Reeb charts. To assess the effectiveness of our proposed 3D shape retrieval method, we used SHREC 2012, SHREC 2011, and MCGill databases. According to the experimental study, it has been shown that our method gives satisfactory results with regards to the state-of-the-art methods.

Applications to the dynamic compression based on segmentation In chapter 6, we proposed a 3D dynamic compression scheme based on a segmentation approach. This latter is based upon idea from Morse theory. The main idea is to exploit the Reeb graph representation proposed previously. The segmentation process is performed based on the values of the scalar function which allows to partition the mesh into rigid parts while considering the motion of each region over time. A refinement step is added based on the curvature information in order to improve the vertex distributions on the borders of the regions. The developed segmentation technique is exploited in a compression system. The first frame of the sequence, considered as reference frame is partitioned. Then, each region is modeled by an affine transform and its associated animation weight. The motion of the vertices in each cluster are obtained by weighting the corresponding affine transforms. The obtained key-vector, associates each vertex to the index of the cluster to which it belongs, is compressed by an arithmetic encoder. The two sets of affine transforms and animation weights are uniformly quantized and compressed by an arithmetic encoder. The first frame of the sequence is compressed using a static mesh encoder. Finally, the residual error is calculated as a simple difference between the original coordinates of a vertex, and the estimated one at each frame. In order to improve the performance of our encoder, the quantization of the temporal prediction errors is optimized by using a bit allocation procedure. Performance evaluation has shown that our proposed coding scheme offers good compression performance in terms of Bitrates and distortion. The experimental study showed that our approach leads to satisfactory results with respect to the state of the art.

7.2 Open problems and Perspectives

In this thesis, we presented a new topological modeling approach based on Reeb graphs. The latter is exploited in two widely used applications which are pattern recognition and compression.

From Reeb graphs to concise skeletons

We proposed to incorporate the morse theory motion to rebuilt concise skeleton representation for 3D dynamic meshes. This structural description offers an expressive representation of the global surface structure. We are thinking of the information on the objet contours instead of feature vertex to define a new scalar function. The feature points may not contain all information about the topology of the object while the contours described with more implicit manner the shape. The question that arises is how the articulations can be detected using this kind of information. Since we we handle dynamic objects, our interest is to obtain a faithful representation with respect to the motions over time.

Toward a partial 3D shape retrieval based on Heat kernel We plan to investigate the proposed 3D shape retrieval approach and integrate it in a partial matching scheme, which can be used for non-rigid 3D shape retrieval. Instead of segmented the Reeb graph into a set of Reeb charts to compute a pair of signature. We proposed calculated a multi scale signature based on the heat diffusion properties. This allows to match most similar shape parts without using the whole objet. Consequently, reduce the computing time while ensuring better accuracy.

Towards anatomically precise segmentation We plan to investigate how to automatically selected to number of clusters and iterations taking into account the introduced motion compensation error. That leads to improve the performance of the proposed compression based segmentation scheme in terms of reconstruction quality. The main idea consists in automatically select the number of clusters and iterations that provides a lower quality degradation distortion.

.1 Appendix A

The prediction errors are calculated as a simple difference between the original coordinates of the vertex v at frame i , and the estimated one. In this appendix, we demonstrate that the difference between the coordinates allows to concentrate the energy around zero. Thus each empirical probability density function, of the prediction error coordinates, fits to a generalized gaussian distribution.

To validate this hypothesis, a χ^2 test was conducted between the components of three subbands, and the probability density functions (pdf) of a Generalized Gaussian Distribution (*GGD*), given by:

$$p(x) = a \exp^{-|bx|^\alpha}, \quad (1)$$

with $b = \frac{1}{\sigma} \sqrt{\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}}$ and $a = \frac{b\alpha}{2\Gamma(1/\alpha)}$. The parameter α is calculated using the variance σ^2 and the fourth-order moment of each subband.

The χ^2 test allows to verify whether each probability density function of three subbands can be modeled by a Generalized Gaussian Distribution (*GGD*). The χ^2 value is computed using the following expression:

$$\chi_{k-1}^2 = \sum_{v=1}^k \frac{(n_v - np_v)^2}{np_v}, \quad (2)$$

- k is the number of cells,
- n_v is the median value of the cell v ,
- n is the number of samples present in the cell v ,
- p_v is the expected theoretical frequency given by Eq. 1.

The χ^2 statistic is a measure of the difference between the expected theoretical numbers and those observed in the sample. When the value of χ_{k-1}^2 test increases, the disagreement will be more important. The coincidence is perfect if $\chi^2 = 0$. Subsequently, we compare χ_{k-1}^2 to the set of values $\chi_{k-1,\beta}^2$ with β represents the tolerance. If $\chi_{k-1}^2 > \chi_{k-1,\beta}^2$ and if it is large enough, then the assumption of actually dealing with the necessary theoretical distribution is to be rejected with an error probability of at most β .

We provide a Chi-Square distribution table (Table 1) for some values of $k - 1$ and β , since the cell number k is, usually, in the range of [6, 30].

Fig. 1 shows the empirical and theoretical probability density functions (pdf) of the three subbands for Cow model. From this figure, we can clearly observe that the energy is concentrated around zero.

The different values of χ_{k-1}^2 are illustrated in Table 2. All the obtained values of χ_{k-1}^2 are strictly less than $\chi_{k-1,0.995}^2$ (see Table 1). From these results, we can notice that the probability density function of the three subbands can be modeled by a Generalized Gaussian Distribution (*GGD*) which gives good estimates with a high level of reliability surpassing 99.5%. This is to say that the *GGD* is well suited to model the empirical distribution of the prediction error coordinates.

Table 1: Chi-Square distribution table.

	0.995	0.99	0.95	0.9	0.1	0.05	0.01
5	0.412	0.554	1.145	1.610	9.236	11.070	15.086
6	0.676	0.872	1.635	2.204	10.645	12.592	16.812
7	0.989	1.239	2.167	2.833	12.017	14.067	18.475
8	1.344	1.646	2.733	3.490	13.362	15.507	20.090
9	1.735	2.088	3.325	4.168	14.684	16.919	21.666
10	2.156	2.588	3.940	4.865	15.987	18.307	23.209
11	2.656	3.053	4.575	5.578	17.275	19.675	24.725
12	3.074	3.571	5.226	6.304	18.549	21.026	26.217
13	3.565	4.107	5.892	7.042	19.812	23.362	27.688
14	4.075	4.660	6.571	7.790	21.064	23.685	29.141
15	4.601	5.229	7.261	8.547	22.307	24.996	30.378
16	5.142	5.812	7.962	9.312	23.542	26.296	32.000
17	5.697	6.408	8.672	10.085	24.769	27.587	33.409
18	6.265	7.015	9.390	10.865	25.989	28.869	34.805
19	6.844	7.633	10.117	11.651	27.204	30.144	36.191
20	7.434	8.260	10.851	12.443	28.412	31.410	37.566
21	8.034	8.897	11.591	13.240	29.615	32.671	38.932
22	8.643	9.542	12.338	14.041	30.813	33.924	40.289
23	9.260	10.196	13.091	14.848	32.007	35.172	41.638
24	9.886	10.856	13.848	15.659	33.196	36.415	42.980
25	10.520	11.524	14.611	16.473	34.382	37.652	44.314
26	11.160	12.198	15.379	17.292	35.563	38.885	45.642
27	11.808	12.879	16.151	18.114	36.741	40.113	46.963
28	12.461	13.565	16.928	18.939	37.916	41.337	48.278
29	13.121	14.256	17.708	19.768	39.087	42.557	49.588

Table 2: χ^2 test for Cow model, with $k \in 6, 16, 26$.

	x coordinates	y coordinates	z coordinates
$k = 6$	$\chi_5^2 = 0.031 < \chi_{5,0.995}^2$	$\chi_5^2 = 0.067 < \chi_{5,0.995}^2$	$\chi_5^2 = 0.034 < \chi_{5,0.995}^2$
$k = 16$	$\chi_{15}^2 = 3.127 < \chi_{15,0.995}^2$	$\chi_{15}^2 = 4.879 < \chi_{15,0.995}^2$	$\chi_{15}^2 = 3.967 < \chi_{15,0.995}^2$
$k = 26$	$\chi_{25}^2 = 8.745 < \chi_{25,0.995}^2$	$\chi_{25}^2 = 10.232 < \chi_{25,0.995}^2$	$\chi_{25}^2 = 9.315 < \chi_{25,0.995}^2$

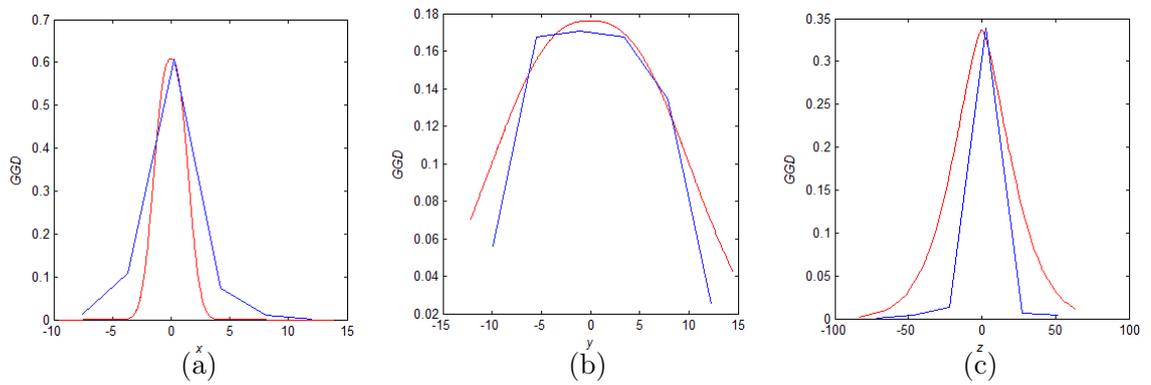


Figure 1: Probability density functions of the predicted coefficient coordinates for the Cow model: (a) X-coordinates, (b) Y-coordinates, (c) Z-coordinates. The blue and red curves represent the real distribution and the approximated one respectively.

Bibliography

- [A. 2009] A. Srivastava A., C. Samir C., S. Joshi S. and M. Daoudi. *Elastic shape models for face analysis using curvilinear coordinates*. Journal of Mathematical Imaging and Vision, vol. 33, pages 253–265, 2009. (Cited page 37.)
- [Agathos 2009] A. Agathos, I. Pratikakis, P. Papadakis, S. Perantonis, P. Azariadis and N. Sapidis. *Retrieval of 3D Articulated Objects using a graph-based representation*. pages 1–8. Eurographics Workshop on Shape Retrieval, 2009. (Cited pages 2, 83, 85, 86 and 88.)
- [Aguiar 2008] E. Aguiar, C. Theobalt, S. Thrun and H.-P. eidel. *Automatic Conversion of Mesh Animations into Skeleton Based Animations*. Computer Graphics Forum - Eurographics, vol. 27, no. 2, pages 389–397, 2008. (Cited pages 39, 55 and 63.)
- [Ahn 2011] J. Ahn, C. Kim, C. Kuo and Y. Ho. *Motion compensated compression of 3D animation models*. IEEE Electronics Letters, vol. 27, no. 24, pages 1445–1451, 2011. (Cited page 92.)
- [Alexa 2000] M. Alexa and W. Müller. *Representing animations by principal components*. Computer and Graphics Forum, vol. 19, no. 3, pages 411–418, 2000. (Cited pages 92 and 93.)
- [Amjoun 2006] R. Amjoun, R. Sondershaus and W. Straber. *Compression of complex animated meshes*. pages 606–613. Graphics International Conference, 2006. (Cited pages 43 and 94.)
- [Angela 2004] B. Angela and I. Tobias. *3D Shape matching using skeleton graphs*. Simulation and Visualization, 2004. (Cited page 40.)
- [Aouada 2010] D. Aouada and H. Krim. *Squigraphs for Fine and Compact Modeling of 3-D Shapes*. IEEE Transactions on Image Processing, vol. 19, pages 306–321, 2010. (Cited pages 55 and 71.)
- [Barra 2013] V. Barra and S. Biasotti. *3D shape retrieval using Kernels on Extended Reeb Graphs*. Pattern Recognition, vol. 46, pages 2985–2999, 2013. (Cited pages 5, 81 and 83.)
- [BelkiIn 2008] M. BelkiIn, J. Sun and Y. Wand. *Discrete Laplace Operator On Meshed Surface*. pages 278–287. In Proceedings of SOCG, 2008. (Cited pages 31, 32, 52 and 63.)
- [Benhabiles 2012] H. Benhabiles, G. Lavoué, J.-P Vandeborre and M. Daoudi. *Kinematic Skeleton extraction based on motion boundaries for 3D Dynamic Meshes*. pages 1–4. Eurographics Workshop on 3D Object Retrieval, 2012. (Cited pages 2, 40, 56, 62, 63, 64 and 65.)

- [Berreti 2006] S. Berreti, A. Del Bimbo and P. Pala. *Partitioning of 3D meshes using Reeb graphs*. pages 19–22. IEEE ICPR, 2006. (Cited page 47.)
- [Biasotti 2003] S. Biasotti, S. Marini, M. Mortara, G. Patané, M. Spagnuolo and B. Falcidieno. *3D Shape matching through topological structures*. Discrete Geometry for Computer Imagery, Computer Science, Springer Berlin Heidelberg, vol. 2886, pages 194–203, 2003. (Cited page 47.)
- [Biasotti 2006] S. Biasotti, S. Marini, M. Spagnuolo and B. Falcidieno. *Sub-part correspondence by structural descriptors of 3D shapes*. Computer-Aided Design, vol. 38, pages 1002–1019, 2006. (Cited pages 55, 70 and 71.)
- [Biasotti 2008] S. Biasotti, S. Marini, M. Spagnuolo and B. Falcidieno. *Reeb graphs for shape analysis and application*. Theor. Comput. Sci, vol. 392, pages 5–22, 2008. (Cited pages 56 and 70.)
- [Bici 2011] M.O. Bici and G.B. Akari. *Improved prediction methods for scalable predictive animated mesh compression*. Journal of Visual Communication and Image Representation, vol. 22, pages 577–589, 2011. (Cited pages 94, 104, 105 and 109.)
- [Bloomenthal 1997] J. Bloomenthal and C. Bajaj. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers, 1997. (Cited page 22.)
- [Blum 1967] H. Blum. *A transformation for extracting new descriptors of models for perception of speech and visual form*. W. Walthen-Dunn ed. MIT Press Cambridge, 1967. (Cited page 38.)
- [Botsch 2007] M. Botsch, M. Pauly, L. Kobbelt, P. Alliez, B. Lévy, S. Bischoff and C. Rössl. *Geometric modeling based on polygonal meshes*. In Proceedings of the ACM Siggraph Course Notes, 2007. (Cited page 26.)
- [Boulfani-Cuisinaud 2007] Y. Boulfani-Cuisinaud and M. Antonini. *Motion-based geometry compensation for dwt compression of 3D mesh sequence*. pages 606–613. IEEE International Conference in Image Processing, 2007. (Cited pages 43 and 94.)
- [Briceno 2003] H. Briceno, P. Sander, L. McMillan, S. Gotler and H. Hoppe. *Geometry videos: a new representation for 3D animations*. pages 136–146. In: ACM symposium on computer animation, 2003. (Cited pages 92 and 104.)
- [Bronstein 2006] A.M. Bronstein, M.M. Bronstein and R. Kimmel. *Efficient computation of isometry-invariant distances between surfaces*. SIAM Journal on Scientific Computing, vol. 28, no. 5, pages 1812–1836, 2006. (Cited page 37.)
- [Bronstein 2009] A.M. Bronstein, M.M. Bronstein and R. Kimmel. *A Gromov-Hausdorff Framework with Diffusion Geometry for Topologically-Robust Non-rigid Shape Matching*. International Journal of Computer Vision, vol. 89, no. 2-3, pages 266–286, 2009. (Cited page 70.)

- [Charpiat 2005] G. Charpiat, O. Faugeras and R. Keriven. *Approximations of shape metrics and application to shape warping and empirical shape statistics*. Journal of Foundations of Computational Mathematics, vol. 5, no. 1, pages 1–58, 2005. (Cited page 37.)
- [Chen 2003] D. Chen, X. Tian, Y. Shen and M. Ouhyoung. *On Visual Similarity Based 3D Model Retrieval*. Computer Graphics Forum, vol. 22, no. 3, pages 223–232, 2003. (Cited page 70.)
- [Cho 2010] J.W. Cho, S. Valette, J. H. Park, H. Y. Jung and R. Prost. *3-D mesh sequence compression using wavelet-based multi-resolution analysis*. Applied Mathematics and Computation, vol. 216, pages 410–425, 2010. (Cited pages 92, 93 and 94.)
- [Cignoni 1998] P. Cignoni, C. Rochini and R. Scopigno. *Metro: measuring error on simplified surfaces*. Computer Graphics Forum, vol. 17, no. 2, pages 167–174, 1998. (Cited page 103.)
- [Collins 2005] G. Collins and A. Hilton. *A Rigid Transform Basis for Animation Compression and Level of Detail*. pages 21–29. ACM Symposium on Interactive 3D Graphics, 2005. (Cited page 104.)
- [Coxeter 1989] H.S.M. Coxeter. *Introduction to Geometry*. Wiley, 2nd edition, 1989. (Cited page 17.)
- [D. Zorin 2000] T. DeRose L. Kobbelt A. Levin D. Zorin P. Schröder and W. Sweldens. *Subdivision for modeling and animation*. In Course notes of ACM SIGGRAPH, 2000. (Cited page 23.)
- [Daoudi 2007] M. Daoudi, T. Filali Ansary, J. Tierny and J.P. Vandeborre. *3D-Mesh Models: View-Based Indexing and Structural Analysis*. pages 298–307. DELOS Conference, 2007. (Cited page 70.)
- [Dey 2006] T.K. Dey and S. Jian. *Defining and computing curve skeletons with medial geodesic function*. pages 143–152. In Eurographics Symposium on Geometry Processing, 2006. (Cited page 70.)
- [Dutagaci 2012] H. Dutagaci, C.P. Cheung and A. Godil. *Evaluation of 3D interest point detection techniques via human-generated ground truth*. Journal of Visual Computer, vol. 28, no. 9, pages 901–917, 2012. (Cited page 70.)
- [El Khoury 2012] R. El Khoury, J.P Vandeborre and M. Daoudi. *Indexed heat curves for 3D-model retrieval*. pages 1964–1967. International Conference on Pattern Recognition, Tsukuba, Japon, 2012. (Cited pages 55, 71, 82, 83 and 84.)
- [Floater 2002] M.S. Floater and K. Hormann. *Parameterization of Triangulations and Unorganized Points*. Tutorials on Multiresolution in Geometric Modelling, pages 287–316, 2002. (Cited page 74.)

- [Floater 2005] M.S. Floater and K. Hormann. *Surface Parameterization: a Tutorial and Survey*. Advances in Multiresolution for Geometric Modelling, Mathematics and Visualization, Springer, pages 157–186, 2005. (Cited pages 73 and 74.)
- [Furuya 2009] T. Furuya and R. Ohbuchi. *Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features*. In Proceedings of the ACM International Conf. on Image and Video Retrieval, 2009. (Cited pages 70, 80, 83 and 84.)
- [G. 1946] Reeb G. *Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique*. Comptes Rendus Acad. Sciences, vol. 222, pages 847–849, 1946. (Cited page 44.)
- [Gal] R. Gal and D. Cohen-OR. IEEE Transactions on Visualization and Computer Graphics, pages 261–271. (Cited pages 54 and 56.)
- [Grenander 2010] U. Grenander, A. Srivastava and M.I. Miller. *Asymptotic performance of bayesian object recognition*. IEEE Transactions of Information Theory, vol. 46, no. 4, pages 1658–1666, 2010. (Cited page 37.)
- [Gu 2003] X. Gu and S.T. Yau. *Surface Classification Using Conformal Structures*. IEEE International Conference on Computer Vision, 2003. (Cited pages 1, 33, 34 and 35.)
- [Gu 2004] X. Gu, Y. Wang, T.F. Chan, P.M. Thompson and S-T. Yau. *Genus Zero Surface Conformal Mapping and its Applications to Brain Surface Mapping*. IEEE Transactions on Medical Imaging, vol. 23, no. 8, page 949, 2004. (Cited pages 33 and 34.)
- [Guskov 2004a] I. Guskov and A. Khodakovsky. *Wavelet compression of parametrically coherent mesh sequences*. pages 183–192. In: Eurographics ACM SIGGRAPH symposium on computer animation, 2004. (Cited pages 92 and 93.)
- [Guskov 2004b] I. Guskov and A. Khodakovsky. *Wavelet compression of parametrically coherent mesh sequences*. ACM SIGGRAPH symposium on computer animation, 2004. (Cited page 93.)
- [Hilaga 2001] M. Hilaga, Y. Shinagawa, T. Kohmura and T.L. Kuni. *Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes*. pages 203–212. ACM SIGGRAPH, 2001. (Cited pages 45, 46, 54, 56, 70 and 71.)
- [Hsu 2002] E.P. Hsu. *Stochastic Analysis on Manifolds*. American Mathematical Society, 2002. (Cited page 50.)
- [Huang 2009] Q. Huang, M. Wicke, B. Adams and L.J. Guibas. *shape decomposition using modal analysis*. EUROGRAPHICS, vol. 28, no. 2, 2009. (Cited page 32.)

- [Ibarria 2003] L. Ibarria and J. Rossignac. *space-time compression of the 3D animations of triangle meshes with fixed connectivity Dynapack*. pages 126–135. ACM symposium on computer animation, 2003. (Cited page 92.)
- [Isenburg 2009] M. Isenburg and P. Lindstrom. *Streaming meshes*. IEEE Visualization, 2009. (Cited page 30.)
- [Jain 2007] V. Jain and H. Zhang. *A spectral approach to shape-based retrieval of articulated 3D models*. Computer-Aided Design, vol. 39, no. 5, 2007. (Cited page 70.)
- [K. Mamou 2006] T. Zaharia K. Mamou and F. Prêteux. *Multi-chart geometry video: A compact representation for 3D animations*. pages 711–718. IEEE 3DPVT, 2006. (Cited pages 43, 92, 93, 104 and 105.)
- [Kanai 2000] T. Kanai and H. Suzuki. *Approximate Shortest Path on Polyhedral Surface Based on Selective Refinement of the Discrete Graph and its Applications*. pages 241–250. IEEE Proc. Geometric Modeling and Processing, 2000. (Cited page 47.)
- [Karni 2000] C. Karni Z.and Gostman. *Spectral compression of mesh geometry*. pages 279–286. In SIGGRAPH Proceedings of the 27th annual conference on Computer graphics and interactive techniques, 2000. (Cited page 31.)
- [Karni 2004] Z. Karni and C. Gostman. *Compression of soft-body animation sequences*. Pattern Recognition, vol. 28, pages 25–34, 2004. (Cited pages 92, 93 and 102.)
- [Kilian 2007] M. Kilian, N.J. Mitra and H. Pottman. *Geometric modeling in shape space*. In Proceedings of SIGGRAPH, 2007. (Cited page 37.)
- [Krongold 1998] B. S. Krongold, K. Ramchandran and D. L. Jones. *Computationally efficient optimal power allocation algorithm for multicarrier communication systems*. pages 1018–1022. Proceedings of International Conference on Communications, 1998. (Cited page 101.)
- [Lam 2014] K.C. Lam, C. Wen and L.M. Lui. *Conformal-Based Surface Morphing and Multi-Scale Representation*. Axioms, vol. 3, no. 2, pages 222–243, 2014. (Cited pages 1, 34 and 35.)
- [Lavoué 2011] G. Lavoué. *Bag of Words and Local Spectral Descriptor for 3D Partial Shape Retrieval*. Eurographics Workshop on 3D Object Retrieval, 2011. (Cited pages 70, 83, 85 and 86.)
- [Lee 1998] A.W.F Lee, W. Sweldens, P. Schroder, L. Cowsar and D. Dobkin. *Maps: Multiresolution adaptative parametrization of surface*. pages 95–104. Computer Graphics Proceedings SIGGRAPH, 1998. (Cited page 28.)

- [Lee 2007] P.F. Lee, C.K. Kao, J.L. Tseng, B.S. Jong and T.W. Lin. *3D animation compression using affine transformation matrix and principal component analysis*. Transactions on Information and Systems, vol. 7, pages 1073–1084, 2007. (Cited pages 92 and 93.)
- [Lengyel 1999] J. Lengyel. *Compression of time-dependent geometry*. pages 89–96. ACM Symposium on Interactive 3D Graphics, 1999. (Cited pages 43, 92, 93 and 94.)
- [Levy 2006] B. Levy. *Eigenfunctions Towards an algorithm that understands geometry*. IEEE International Conference on Shape Modeling and Application, 2006. (Cited pages 29 and 30.)
- [Lévy 2009] B. Lévy and H. Zhang. *Spectral Mesh Processing*. ACM SIGGRAPH, 2009. (Cited page 30.)
- [Li 2012] B. Li, A. Godil, M. Aono, X. Bai, T. Furuya, L. Li, R. Lopez-Sastre, H. Johan, R. Ohbuchi, C. Redondo-Cabrera, A. Tatsuma, S. Yanagimachi and S. Zhang. *SHREC'12 Track: Generic 3D Shape Retrieval*. pages 119–126. Eurographics Workshop on 3D Object Retrieval, 2012. (Cited page 70.)
- [Li 2013] C. Li and A. Ben Hamza. *A multiresolution descriptor for deformable 3D shape retrieval*. The Visual Computer, vol. 29, no. 6, pages 513–524, 2013. (Cited page 71.)
- [Li 2014] C. Li and A.B Hamza. *Symmetry Discovery and Retrieval of Nonrigid 3D Shapes Using Geodesic Skeleton Paths*. Multimedia Tools Appl., vol. 72, no. 2, pages 1027–1047, 2014. (Cited page 40.)
- [Lian 2013] Z. Lian, A. Godil, B. Bustos, M. Daoudi, I. Hermans, S. Kawamuraf, Y. Kuritaf, G. Lavoué, V. Nguyenh, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens and H. Tabiaand D. Vandermeulen. *A comparison of methods for non-rigid 3D shape retrieval*. Pattern Recognition Journal, vol. 46, no. 1, pages 449–461, 2013. (Cited page 70.)
- [Lowe 2004] D.G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, vol. 60, no. 2, 2004. (Cited page 80.)
- [Lu 2005] X. Lu and A.K. Jain. *Deformation analysis for 3D face matching*. pages 99–104. In Proceedings 7th IEEE workshop on Applications of Computer Vision, 2005. (Cited page 37.)
- [Lu 2006] X. Lu, A.K. Jain and D. Colbry. *Matching 2.D face scans to 3D models*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 11, pages 31–43, 2006. (Cited page 37.)

- [Lui 2013] L.M. Lui, C. Wen and A.X. Gu. *A Conformal Approach for Surface Inpainting*. AIM Inverse Problems and Imaging, vol. 7, no. 3, pages 863–884, 2013. (Cited page 31.)
- [Mamou 2008] K. Mamou, T. Zaharia and F. Preteux. *The mpeg-4 standard for animated mesh compression*. pages 2676–2679. IEEE International Conference on Image Processing, 2008. (Cited pages 104, 105 and 109.)
- [Marpe 2003] D. Marpe, H. Schwarz and T. Wiegand. *Context-Based Adaptive Binary Arithmetic Coding in the H.264/MPEG4-AVC Video Compression Standard*. IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, no. 7, pages 620–636, 2003. (Cited page 104.)
- [Meyer 2002] M. Meyer, M. Desbrun, P. Schroder and A.H. Barr. *Discrete Differential Geometry Operators for Triangulated 2-manifolds*. VisMath02", 2002. (Cited page 31.)
- [Mitchell 1987] J.S.B. Mitchell, D.M. Mount and C.H. Papadimitriou. *The Discrete Geodesic Problem*. SIAM J. Computing, vol. 16, pages 647–667, 1987. (Cited page 47.)
- [Mohamed 2012] W. Mohamed and A. Ben Hamza. *Reeb graph path dissimilarity for 3D object matching and retrieval*. The Visual Computer, vol. 28, pages 305–318, 2012. (Cited pages 55 and 71.)
- [Müller 2005] K. Müller, A. Smolic, M. Kautzner, P. Eisert and T. Wiegand. *IEEE International Conference on Image Processing*. Pattern Recognition, vol. 1, pages 621–624, 2005. (Cited pages 104 and 105.)
- [Mullen 2004] P. Mullen, Y. Tong, P. Alliez and M. Desbrun. *Spectral conformal parametrization*. pages 1487–1494. ACM Symposium of Geometry Processing, 2004. (Cited page 32.)
- [Müller 2006] K. Müller, A. Smolic, M. Kautzner and P. Eisert. *Rate-distortion-optimized predictive compression of dynamic 3d mesh*. SP:IC, vol. 9, pages 812–828, 2006. (Cited page 93.)
- [Mumford 1989] D. Mumford and J. Shah. *Optimal approximations by piecewise smooth functions and associated variational problems*. Commun. Pure Appl. Math, vol. 42, no. 5, pages 577–685, 1989. (Cited pages 43 and 95.)
- [N. Stefanoski 2008] J. Ostermann N. Stefanoski. *Spatially and temporally scalable compression of animated 3D meshes with MPEG-4/FAMC*. pages 2696–2699. Proceedings of the IEEE International Conference on Image Processing ICIP, 2008. (Cited pages 104 and 105.)
- [Ni 2004] X. Ni, M. Garland and J. Hart. *Fair Morse functions for extracting the topological structure of a surface mesh*. ACM Transactions on Graphics, vol. 23, pages 613–622, 2004. (Cited page 54.)

- [Ohbuchi 2003] R. Ohbuchi, T. Minamitani and T. Takei. *Shape Similarity Search of 3D Models by Using Enhanced Shape Functions*. Theory and Practice of Computer Graphics, pages 97–104, 2003. (Cited page 70.)
- [Osada 2002] R. Osada, T. Funkhouser, B. Chazelle and A. Dobkin. *Shape distributions*. ACM Transactions on Graphics, vol. 21, no. 4, pages 807–832, 2002. (Cited page 70.)
- [Pauly 2001] M. Pauly and M. Gross. *Spectral processing of point-sampled geometry*. pages 379–386. In SIGGRAPH Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 2001. (Cited page 28.)
- [Payan 2005] F. Payan and M. Antonini. *An efficient bit allocation for compressing normal meshes with an error-driven quantization*. pages 466–486. Computer Aided Geometric Design. Special Issue on Geometric Mesh Processing, 2005. (Cited page 100.)
- [Payan 2007] F. Payan and M. Antonini. *Temporal wavelet-based compression for 3D animated models*. Computers and Graphics, vol. 31, pages 77–88, 2007. (Cited pages 92 and 93.)
- [Reuter 2006] M. Reuter, R-E. Wolter and N. Peinecke. *Laplace-beltrami spectra as shape-dna of surfaces and solids*. Computer-Aided Design, vol. 38, no. 4, pages 342–366, 2006. (Cited page 31.)
- [Reuter 2009] M. Reuter, S. Biasotti, D. Giorgi, G. Patané and M. Spagnuolo. *Discrete Laplace-Beltrami operators for shape analysis and segmentation*. Computers and Graphics, vol. 33, no. 3, pages 381–390, 2009. (Cited page 70.)
- [Rosman 2012] G. Rosman, M.M. Bronstein, A.M Bronstein, A. Wolf and R. Kimmel. *Group-Valued Regularization Framework for Motion Segmentation of Dynamic Non-rigid Shapes*. Scale Space and Variational Methods in Computer Vision, pages 725–736, 2012. (Cited pages 43 and 95.)
- [Rosman 2013] G. Rosman, M.M. Bronstein, A.M Bronstein, A. Wolf and R. Kimmel. *Group-Valued Regularization Framework for Motion Segmentation of Articulated Shapes*. Innovations for Shape Analysis in Mathematics and Visualization, pages 263–281, 2013. (Cited pages 43 and 95.)
- [Samir 2006] C. Samir, A. Srivastava and M. Daoudi. *Conformal-Based Surface Morphing and Multi-Scale Representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 11, pages 1858–1863, 2006. (Cited page 37.)
- [Schaefer 2007] S. Schaefer and C. Yuksel. *Example-Based Skeleton Extraction*. pages 153–162. Proceedings of the fifth Eurographics Symposium on Geometry Processing, 2007. (Cited pages 39 and 55.)

- [Shamir 2001] A. Shamir and V. Pascucci. *Temporal and spatial level of details for dynamic meshes*. pages 423–430. In Proceedings of virtual reality systems and techniques, 2001. (Cited page 93.)
- [Shamir 2006] A. Shamir. *Segmentation and Shape Extraction of 3D Boundary Meshes*. EUROGRAPHICS, 2006. (Cited page 42.)
- [Shapira 2008] L. Shapira, A. Shamir and D. Cohen-OR. *Consistent mesh partitioning and skeletonisation using the shape diameter function*. Visual Computer, pages 249–259, 2008. (Cited page 42.)
- [Shen 2008] T. Shen, Y. Zhu, X. Huang, D. Metaxax and L. Axel. *Active volume models with probabilistic object boundary prediction module*. Medical Image Computing and Computer-Assisted Intervention MICCAI, vol. 5241, pages 331–341, 2008. (Cited page 24.)
- [Shilane 2004] P. Shilane, P. Min, M. Kazhdan and T. Funkhouser. *The princeton shape benchmark*. pages 167–178. The princeton shape benchmark, 2004. (Cited page 79.)
- [Smeets 2010] D. Smeets, T. Fabry, J. Hermans, D. Vandermeulen and P. Suetens. *Inelastic deformation invariant modal representation for non-rigid 3D object recognition*. pages 162–174. International Conference on Articulated Motion and Deformable Objects, 2010. (Cited pages 70, 81, 83 and 84.)
- [Sokrine 2005] O. Sokrine, D. Cohen-OR, D. Irony and S. Toledo. *Geometry-aware bases for shape approximation*. IEEE Transactions on Visualization and Computer Graphics, vol. 11, no. 2, pages 171 – 180, 2005. (Cited pages 28 and 42.)
- [Sun 2009] J. Sun, M. Ovsjanikov and L. Guibas. *A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion*. Eurographics Symposium on Geometry Processing, vol. 28, no. 5, 2009. (Cited page 32.)
- [Sun 2010] X. Sun, J. Pan and X. Wei. *Genus Zero Surface Conformal Mapping and its Applications to Brain Surface Mapping*. ComSIS, vol. 7, no. 1, pages 63–74, 2010. (Cited pages 39 and 70.)
- [T. 2008] Tung T. and F. Schmitt. *SHREC’08 entry: Shape retrieval of noisy watertight models using aMRG*. pages 229–230. IEEE International Conference on Shape Modeling and Applications, 2008. (Cited pages 44, 58 and 71.)
- [Tam 2007] G.K.L. Tam and R.W.H. Lau. *Deformable model retrieval based on topological and geometric signatures*. IEEE Transactions on Visualization and Computer Graphics, vol. 13, pages 470–482, 2007. (Cited page 71.)
- [Tangelder 2008] J.W. Tangelder and R.C. Veltkamp. *A survey of content based 3D shape retrieval methods*. Journal of Multimedia Tools and Applications, vol. 39, no. 3, pages 441–471, 2008. (Cited page 70.)

- [Taubin 1995a] G. Taubin. *Estimating the tensor of curvature of a surface from a polyhedral approximation*. pages 907–909. International Conference on Computer Vision, 1995. (Cited page 98.)
- [Taubin 1995b] G. Taubin. *A signal processing approach to fair surface design*. pages 351–358. In Proceedings of SIGGRAPH, 1995. (Cited page 30.)
- [Terzopoulos 1987] D. Terzopoulos, J. Platt, A. Barr and K. Fleicher. *Elastically deformable models*. Computer Graphics, vol. 21, pages 205–214, 1987. (Cited page 23.)
- [Tierny 2006a] J. Tierny, J.P Vandeborre and M. Daoudi. *3D Mesh Skeleton Extraction Using Topological and Geometrical Analyses*. In proceedings of the 14th Pacific Conference on Computer Graphics and Applications, 2006. (Cited pages 1, 39, 40 and 41.)
- [Tierny 2006b] J. Tierny, J.P Vandeborre and M. Daoudi. *Invariant High Level Reeb Graphs of 3D Polygonal Meshes*. pages 105–112. Third International Symposium on 3D Data Processing, Visualization, and Transmission, 2006. (Cited pages 54 and 56.)
- [Tierny 2007] J. Tierny, J.P. Vandeborre and M. Daoudi. *Reeb chart unfolding based 3D shape signatures*. pages 13–16. Eurographics, 2007. (Cited pages 2, 71, 73 and 74.)
- [Tierny 2008a] J. Tierny, J.-P Vandeborre and M. Daoudi. *Fast and Precise Kinematic Skeleton Extraction of 3D Dynamic Meshes*. pages 1–4. IEEE International Conference on Pattern Recognition, 2008. (Cited pages 2, 39, 40, 55, 62, 63, 64 and 65.)
- [Tierny 2008b] J. Tierny, J.P Vandeborre and M. Daoudi. *Enhancing 3D Mesh Topological Skeletons with Discrete Contour Constrictions*. Visual Computer, vol. 28, no. 1, pages 41–55, 2008. (Cited pages 47, 56 and 57.)
- [Tierny 2009] J. Tierny, J.P Vandeborre and M. Daoudi. *Partial 3D Shape Retrieval by Reeb Pattern Unfolding*. Computer Graphics Forum - Eurographics Association, vol. 28, no. 1, pages 41–55, 2009. (Cited pages 2, 55, 70, 71, 72, 73, 83, 84, 85 and 87.)
- [Tung 2005a] T. Tung and F. Schmitt. *The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes*. International Journal of Shape Modeling, vol. 11, pages 91–120, 2005. (Cited pages 54 and 56.)
- [Tung 2005b] T. Tung and F. Schmitt. *The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes*. International Journal of Shape Modeling, vol. 11, pages 91–120, 2005. (Cited page 70.)
- [Turk 1999] G. Turk and J. O.Brien. *Shape transformation using variational implicit functions*. Computer Graphics, vol. 33, pages 335–342, 1999. (Cited page 23.)

- [Valette 2009] S. Valette, R. Chaine and R. Prost. *Progressive Lossless Mesh Compression Via Incremental Parametric Refinement*. Eurographics Symposium on Geometry Processing, vol. 28, no. 5, pages 1301–1310, 2009. (Cited page 98.)
- [Váša 2014] L. Váša, S. Marras, K. Hormann and G. Burnett. *Compression Dynamic Meshes with Geometric Laplacians*. Computer and Graphics Forum, vol. 33, no. 2, pages 145–154, 2014. (Cited page 94.)
- [Wanf 2007] S. Wanf, Y. Wang, M. Jin, X. David Gu and D. Samaras. *Conformal Geometry and Its Applications on 3D Shape Matching, Recognition, and Stitching*. IEEE Transactions On Pattern Analysis and Machine Intelligence, vol. 29, no. 7, pages 1209–1220, 2007. (Cited pages 32, 33 and 34.)
- [Wang 2006] S. Wang, Y. Wang, M. JIN, X. GU and D. Samaras. *3D surface matching and recognition using conformal geometry*. pages 2453–2460. IEEE conf. in Computer Vision and Pattern Recognition, 2006. (Cited page 74.)
- [Yang 2002] J. Yang, C. Kim and S.U. Lee. *Compression of 3D triangle mesh sequences based on vertex-wise motion vector prediction*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 12, pages 1178–1184, 2002. (Cited page 92.)
- [Yang 2004] J.H. Yang, C.S. Kim and S.U. Lee. *Progressive compression of 3D dynamic sequence*. pages 1975–1978. Proceedings of the International Conference on Image Processing ICIP, 2004. (Cited page 93.)
- [Zhou 2004] K. Zhou, H. Bao and J. Shi. *3D surface filtering using spherical harmonics*. Computer Aided Design, vol. 36, no. 4, pages 363–375, 2004. (Cited page 28.)