



Numéro d'ordre: 4218

UNIVERSITÉ DES SCIENCES ET TECHONOLOGIES DE LILLE
LABORATOIRE D'INFORMATIQUE FONDAMENTALE DE LILLE

THÈSE

Présentée en vue d'obtenir le grade de Docteur,
spécialité Informatique
par
Julien Tierny

REEB GRAPH BASED 3D SHAPE MODELING AND APPLICATIONS

Thèse soutenue le 2 Octobre 2008 devant le jury composé de :

Mme	Sophie Tison	Professeur, USTL	(Présidente)
M.	Atilla Baskurt	Professeur, INSA Lyon	(Rapporteur)
M.	Bruno Lévy	Directeur de Recherche, INRIA Nancy Grand Est	(Rapporteur)
M.	Claude Labit	Directeur de Recherche, INRIA Rennes	(Examinateur)
M.	Anuj Srivastava	Professeur, Florida State University	(Examinateur)
M.	Mohamed Daoudi	Professeur, TELECOM Lille 1	(Directeur)
M.	Jean-Philippe Vandeborre	Maître de Conférences, TELECOM Lille 1	(Co-encadrant)

UNIVERSITÉ DES SCIENCES ET TECHONOLOGIES DE LILLE
Laboratoire d'Informatique Fondamentale de Lille
UMR USTL/CNRS 8022 – Bat. M3
Cité Scientifique – 59655 Villeneuve d'Ascq Cedex – FRANCE
Tel: +33 (0)3 28 77 85 41 – Fax: +33 (0)3 28 77 85 37

*This manuscript is dedicated
to all those who supported me along the years.*

“ Any problem which is non-linear in character, which involves more than one coordinate system or more than one variable, or where structure is initially defined in the large, is likely to require considerations of topology and group theory for its solution. In the solution of such problems classical analysis will frequently appear as an instrument in the small, integrated over the whole problem with the aid of topology or group theory. ”

Marston Morse, 1934 (Mor34)

FOREWORD

THIS manuscript reviews my research work during the last three years (from October 2005 to October 2008) at the Computer Science Laboratory of Lille (LIFL) for the obtention of a Ph.D. in Computer Science.

This work has been supervised by Dr. Jean-Philippe Vandeborre and Pr. Mohamed Daoudi (Institut TELECOM). It has been mainly funded through a French Ministry for Research Ph.D. grant and partially funded by the European Network of Excellence *Delos* (No. 507618) and through an ANR grant (Agence Nationale de la Recherche, No 07-MDCO-015).

During this period, I also held a teaching assistant position (*Moniteur de l'Enseignement Supérieur*) at the Computer Science department (UFR IEEA) of the University of Sciences and Technologies of Lille.

This manuscript presents most of the published results of this research work (chapters 5 and 6: (TVDo6b, TVDo6a, TVDo8a), chapter 7: (TVDo7a, TVDo8c) and chapter 8: (TVDo7b, TVDo8b).

The reading of this manuscript only requires basic background in Computer Science and algorithmic; most of the mathematical notions are introduced in a dedicated chapter.

ACKNOWLEDGEMENTS

First of all, I would like to thank my two thesis advisors for their trust, and more specifically for having made me discover scientific research and the two research fields of top excitement (!) that are Shape Analysis and Computer Graphics. During these three years, Pr. Mohamed Daoudi has been a mentor to me, even more than at a pure scientific level. I would like to express my gratitude to Dr. Jean-Philippe Vandeborre for his outstanding human qualities and its unconditional encouragements which definitely helped me *mutate* into a grown-up researcher.

Then, I would like to thank the members of the committee and especially the reviewers of the manuscript for having accepted this significant task (in the middle of the summer!): M. Atilla Baskurt (Professor at INSA Lyon) – who assessed his interest in my work since the very beginning of my thesis, presented me to the community and who finally accepted to review my manuscript despite his time consuming responsibilities – and M. Bruno Lévy (INRIA Research Director) – who also showed early his interest in my work, afforded me his trust in several occasions and who finally also kindly accepted to sacrifice his precious time for the manuscript reviewing. I also thank the committee members Mme Sophie Tison (Professor and head of the Computer Science Laboratory of the University of Lille 1), M. Claude Labit (INRIA Research Director) and M. Anuj Srivastava (Professor at Florida State University)¹. All these people made me the honor to be present for this special day despite their highly charged agendas and I'm sincerely grateful for that.

A special thank goes to my research buddies who were, for some of them, crazy enough to be my roommates the time of a conference (in alphabetical order, sorry): Grégoire Aujay, Sébastien Barbier, Robert Bargmann, Christian Boucheny, Flavien Bridault, Jérôme Champavere, Mélanie Cornillac, Tarik Filali-Ansary, Emmanuel Filiot, Arnaud Fontaine, Olivier Gauwin, Franck Hétroy, Samuel Hym, Pierre Kraemer, Linda Lu Lin, Sylvain Mongy, Basile Sauvage, François Septier, Slawek Staworko,

¹ I would like to specially thank Pr. Anuj Srivastava and Dr. Franck Hétroy for the support they brought to my career in several occasions.

Thierry Urruty, Bruno Vallet, Johann Vandromme and Samuel Vidal. If you need to remember only one name for the posterity, remember that of Emmanuel Filiot ;)

I thank my family, my friends, people at TELECOM Lille 1 and LIFL for their never decreasing warm encouragements. I also thank (my) Sophie for her love in general and for her kind support during the dissertation time.

Finally, I would like to thank all the contributors to the GNU related projects for their idealogic commitment and for the excellence of their software.

AUTHOR'S PUBLICATIONS

INTERNATIONAL PUBLICATIONS

Book chapters

- Stefano Berretti, Mohamed Daoudi, Alberto Del Bimbo, Tarik Filali-Ansary, Pietro Pala, **Julien Tierny** and Jean-Philippe Vandeborre, “*3D object indexing*”, chapter of “*3D object processing: compression, indexing and watermarking*”, Ed. Wiley, 2008.

Journals

- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi, “*Partial 3D shape retrieval by Reeb pattern unfolding*”, **Computer Graphics Forum** (Eurographics Association), Ed. Blackwell. To appear.
- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi, “*Enhancing 3D mesh topological skeletons with discrete contour constraints*”, **The Visual Computer** (International Journal of Computer Graphics), Ed. Springer, Volume 24, Number 3, pp. 155-172.

Conferences

- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi, “*Fast and precise kinematic skeleton extraction of 3D dynamic meshes*”, **IEEE International Conference on Pattern Recognition 2008**. To appear.
- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi, “*Reeb chart unfolding based 3D shape signatures*”, **Eurographics 2007**, pp. 13-16.

- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi,
"Topology driven 3D mesh hierarchical segmentation",
IEEE Shape Modeling International 2007, pp. 215-220.
- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi,
"3D mesh skeleton extraction using topological and geometrical analyses",
Pacific Graphics 2006, pp. 85-94.
- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi,
"Invariant high level Reeb graphs of 3D polygonal meshes",
IEEE 3DPVT 2006, pp. 105-112.

Invited conferences

- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi,
"Geometry flavored topological skeletons: applications to shape handling, understanding and retrieval",
Second DELOS Conference, 2007.
- Mohamed Daoudi, Tarik Filali-Ansary, **Julien Tierny** and Jean-Philippe Vandeborre,
"3D mesh models: view-based indexing and structural analysis",
First DELOS Conference 2007, Lecture Notes in Computer Science,
pp. 298-307.

Software

- **Julien Tierny**
"SINAMIS Is Not A Mesh Indexing System",
Benchmarking tools and partial 3D shape retrieval system implementation (paper: *"Partial 3D shape retrieval by Reeb pattern unfolding"*), binary release.
<http://www.lifl.fr/~tierny/sinamis.html>.

LOCAL PUBLICATIONS

Conferences

- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi,
"Signatures de formes 3D par dépliage de cartes de Reeb",
CORESA 2007, pp. 242-246.

- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi,
“*Graphes de Reeb de haut niveau de maillages polygonaux 3D*”,
CORESA 2006, pp. 172-177.

Workshops

- **Julien Tierny**, Jean-Philippe Vandeborre and Mohamed Daoudi,
“*Analyse topologique et géométrique de maillages 3D pour l'extraction de squelette*”,
AFIG 2006 (French chapter of Eurographics), pp. 1-8.

CONTENTS

FOREWORD	vii
ACKNOWLEDGEMENTS	ix
AUTHOR'S PUBLICATIONS	xi
CONTENTS	xv
LIST OF FIGURES	xix
1 INTRODUCTION	1
1.1 SCIENTIFIC CHALLENGES	3
1.2 CONTRIBUTIONS	4
1.3 OUTLINE	6
1.4 FRENCH INTRODUCTION	7
1.4.1 Défis scientifiques	7
1.4.2 Contributions	9
1.4.3 Plan du manuscrit	11
2 3D SHAPE RAW REPRESENTATIONS	13
2.1 VOLUME VS BOUNDARY REPRESENTATIONS	15
2.2 IMPLICIT SURFACE REPRESENTATIONS	17
2.3 EXPLICIT SURFACE REPRESENTATIONS	18
2.3.1 Parametric surfaces	18
2.3.2 Surface meshes	19
2.4 FOCUS ON SURFACE MESHES	21
2.4.1 Design and acquisition of surface meshes	21
2.4.2 Surface mesh storage file formats	23
2.5 EXTRINSIC REPRESENTATION RELATED ISSUES	24
2.5.1 Representation variability	24
2.5.2 Applicative implications	25
2.6 CHAPTER CONCLUSION	26
2.7 FRENCH CHAPTER ABSTRACT	27

3	3D SHAPE MODELING STATE OF THE ART	29
3.1	MOTIVATIONS AND TAXONOMY	31
3.2	GEOMETRY MODELING	32
3.2.1	Spectral modeling	32
3.2.2	Conformal geometry based modeling	37
3.2.3	Riemannian geometry based modeling	41
3.3	TOPOLOGY MODELING	45
3.3.1	Curve skeletons	45
3.3.2	Surface mesh segmentation	49
3.3.3	Differential topology based modeling	54
3.4	CHAPTER CONCLUSION	57
3.5	FRENCH CHAPTER ABSTRACT	58
NOTATIONS		59
4	THEORETICAL BACKGROUND	61
4.1	SMOOTH SETTING	63
4.1.1	Differentiable manifolds	63
4.1.2	Elements of Morse theory	71
4.1.3	Reeb graphs	75
4.2	DISCRETE SETTING	79
4.2.1	Manifold simplicial decompositions	79
4.2.2	Morse theory in the discrete setting	84
4.2.3	Reeb graphs in the discrete setting	87
4.3	CHAPTER CONCLUSION	88
4.4	FRENCH CHAPTER ABSTRACT	89
5	ENHANCED TOPOLOGICAL SKELETONS	91
5.1	SCIENTIFIC ISSUES	93
5.1.1	Intrinsic and expressive PL Morse function	94
5.1.2	Geometry aware Reeb graph computation algorithm	95
5.2	INTRINSIC AND EXPRESSIVE PL MORSE FUNCTION	96
5.2.1	Feature vertex extraction	96
5.2.2	PL Morse function definition	102
5.3	ENHANCED TOPOLOGICAL SKELETON COMPUTATION	109
5.3.1	Concept of Reeb graph in the discrete setting	109
5.3.2	Reeb graph computation algorithm	113
5.3.3	Data-structures and shape modeling	115
5.3.4	Time complexity	117
5.3.5	Experiments and results	120

5.4	CHAPTER CONCLUSION	125
5.5	FRENCH CHAPTER ABSTRACT	126
6	APPLICATION TO CONTROL SKELETON EXTRACTION	129
6.1	CONTROL SKELETON PARADIGM	131
6.1.1	Motivations	131
6.1.2	Related work	132
6.2	AUTOMATIC CONTROL SKELETON EXTRACTION BY CONTOUR CONTRACTION AND CONTOUR CONSTRICTION COMPUTATION .	133
6.2.1	Rigid segment definition by contour contraction	133
6.2.2	Joint definition by contour constriction computation	135
6.3	EXPERIMENTS AND RESULTS	138
6.3.1	Parameter setting	140
6.3.2	Computation times	141
6.3.3	Robustness	142
6.3.4	Discussion	143
6.4	APPLICATION TO SHAPE POSE EDITING	144
6.5	CHAPTER CONCLUSION	145
6.6	FRENCH CHAPTER ABSTRACT	145
7	APPLICATION TO PARTIAL SHAPE RETRIEVAL	147
7.1	SCIENTIFIC CONTEXT	149
7.1.1	Motivations	149
7.1.2	Related work	150
7.1.3	Problem statement and method overview	151
7.2	REEB CHART UNFOLDING BASED GEOMETRY SIGNATURE	154
7.2.1	Reeb chart topology	154
7.2.2	Reeb chart unfolding	157
7.3	REEB PATTERNS IN REEB GRAPHS	161
7.3.1	Reeb patterns and structural distortion	161
7.3.2	Reeb pattern topology	163
7.4	PARTIAL SIMILARITY COMPUTATION BASED ON REEB PATTERN UNFOLDING	167
7.4.1	Reeb pattern combination enumeration	168
7.4.2	Expansion of common sub-graphs	168
7.4.3	Intra-Reeb pattern node-to-node matching	170
7.4.4	Partial similarity estimation	170
7.5	EXPERIMENTAL STUDY	171
7.5.1	SHREC 2007 partial shape retrieval benchmark	172

7.5.2	Experimental setup	174
7.5.3	Performance evaluation	176
7.5.4	Robustness evaluation	178
7.5.5	Discussion	179
7.6	APPLICATION TO MODELING BY EXAMPLE	180
7.7	CHAPTER CONCLUSION	180
7.8	FRENCH CHAPTER ABSTRACT	182
8	APPLICATION TO SEMANTIC-ORIENTED SEGMENTATION	185
8.1	SCIENTIFIC CONTEXT	187
8.1.1	Related work	187
8.1.2	Motivations	187
8.1.3	Topology driven 3D mesh segmentation	188
8.1.4	Chapter overview	188
8.2	TOPOLOGY DRIVEN HIERARCHICAL SEGMENTATION	189
8.2.1	Semantic-oriented heuristics	189
8.2.2	Fine segmentation computation	190
8.2.3	Segmentation hierarchy	195
8.2.4	Experiments and results	197
8.3	DYNAMIC SURFACE MESH ANALYSIS	203
8.3.1	Related work	203
8.3.2	Motion characterization	205
8.3.3	Surface decomposition based on motion analysis	206
8.3.4	Early experiments	208
8.4	CHAPTER CONCLUSION	211
8.5	FRENCH CHAPTER ABSTRACT	213
9	CONCLUSION	215
9.1	SUMMARY OF CONTRIBUTIONS	215
9.2	OPEN PROBLEMS AND DIRECTIONS	218
9.3	FRENCH CONCLUSION	221
9.3.1	Résumé des contributions	221
9.3.2	Problèmes ouverts et directions de recherche	225
INDEX		229
BIBLIOGRAPHY		231

LIST OF FIGURES

1.1 Ubiquity of 3D data in nowadays technological society. From left to right and top to bottom: Valence hospital MRI system, facial authentication system (©Aurora), computer- aided design of a cellphone (©Dassault Systems), simula- tion of an earthquake wave propagation (MSB*03), digital acquisition of Michelangelo's David (Stanford University Michelangelo project), animated picture promotional poster (Blender Foundation), screen-shot of a video game ("The legend of Zelda: Twilight Princess", ©Nintendo).	2
2.1 Example of "famous" 3D shapes: the Cyberware horse (2.1(a)), the Utah teapot (2.1(b)) and the Stanford bunny (2.1(c)).	15
2.2 Cutting the Stanford bunny 3D shape: cutting a volume representation (2.2(b)) and a boundary representation (2.2(c)).	15
2.3 Successive slices of a volume representation of a human head obtained by magnetic resonance imaging, showing the inner components of the brain.*	16
2.4 3D modeling and animation (Blender 2.4(a)) and computer- aided design (CATIA V5 ©2.4(b)) systems are exam- ples of software using parametric surfaces as core shape representation.*	19
2.5 Bi-torus represented by a surface mesh (rendered in wire- frame mode).	20
2.6 Lab's Minolta VI-300 laser range scanner.	22
2.7 Standard layout for surface mesh storage with VRML v2.0. .	23
2.8 Trivial transformations applied on the same 3D shape re- sults in different surface mesh representations: affine trans- formation 2.8(b), near-isometric transformation 2.8(c) and sampling modification 2.8(d).	25

3.1	Common signal processing filters (low pass, enhancement and band-exaggeration) applied on the manifold harmonics of a surface mesh (VLo8).	32
3.2	Spectral modeling applied to texture filtering (VL07).	34
3.3	Spectral modeling applied to shading filtering (VL07).	35
3.4	Shape pose editing through Laplacian geometry modeling (LSLCOo5).	35
3.5	Shape blending through Laplacian geometry modeling (SCOL [*] o4).	35
3.6	A uniform grid and its image under a <i>conformal</i> map (angle-preserving). [*]	36
3.7	Conformal mapping of a face surface S to its canonical domain D . From left to right: original textured surface, conformally mapped texture, canonical domain (u, v) coordinate system, conformal coordinate system $\mathbf{S}(u, v)$ (GY03).	37
3.8	Conformal representation of a brain surface. From left to right: $\lambda(u, v)$, same $\lambda(u, v)$ displayed on the canonical domain (the sphere), $\kappa_H(u, v)$, same $\kappa_H(u, v)$ displayed on the canonical domain (GVo4).	38
3.9	Chart atlas based conformal planar parameterization and texture mapping (LPRM02). From left to right: local conformal coordinate systems $\mathbf{S}(u, v)$, chart unfolding in the canonical domain D and corresponding texture mapping on the Stanford bunny 3D shape.	39
3.10	Conformal factor $\lambda(u, v)$ repartition for humanoid 3D shapes represented by surface meshes of different topology (genus-1 and genus-0) (BCG08).	40
3.11	Displacement interpolation and extrapolation (KMP07). The input blue surfaces (1^{st} and 4^{th}) have been interpolated through near-isometric transformations (in green, 2^{nd} and 3^{rd} surfaces) by geodesic computation in shape space. This displacement has then been extrapolated in shape space to produce the final surface (in red, 5^{th} one).	43
3.12	Interactive shape pose editing by shape space exploration (KMP07): the user draws a path (in green) on a 2D-representation of the shape space (middle row, right) and the corresponding shapes (dots in 2D) are interactively computed (in green in bottom row).	44

3.13	Medial axis of a 2D planar shape represented by a boundary contour in bold (ABE07).	46
3.14	A surface mesh, its medial surface and its curve skeleton (DJ06).	47
3.15	Shape pose editing through medial surface topology modeling (YBS07).	48
3.16	Examples of <i>low-level</i> surface mesh segmentation (3.16(a) (LPRM02), 3.16(b) (BM03), 3.16(c) (CSAD04)).	50
3.17	Examples of <i>high-level</i> surface mesh segmentation (3.17(a) (KLT03), 3.17(b) (KLT05), 3.17(c) (LZ07)).	50
3.18	Feature points extracted with the techniques presented in (KLT05) (3.18(a)) and (LZ07) (3.18(b)) respectively.	51
3.19	Adjacency graphs abstracted from the segmentation may not preserve exactly the topology of the surface (KLT05) (two handles and only one loop in the graph).	53
3.20	Surfaces including topological handles and the Reeb graphs corresponding to the colored functions (PSBMo7).	55
4.1	Example of <i>transition function</i> $\phi_{ij} = \phi_j \circ \phi_i^{-1}$ for a 2-manifold (Lee03).	65
4.2	A 2-manifold with one boundary component (Lee03).	67
4.3	Coffee-dunked doughnut analogy to Morse theory (Har98).	72
4.4	Georges Reeb (right) in Oberwolfach, 1953.	75
4.5	A smooth compact 2-manifold \mathbb{M} and the Reeb graph $R(f)$ of its <i>height</i> function f , both embedded in \mathbb{R}^3	77
4.6	Example of geodesic between two vertices of a triangulated surface of the Stanford bunny 3D shape.	82
4.7	The star $St(v)$ of a vertex v (left), its link $Lk(v)$ (middle) and the opposite angles α_i and β_i of the edge (v, v_i) (right).	82
4.8	The discrete mean curvature (left) and the discrete Gaussian curvature (right) computed on a triangulated surface.	83
4.9	A regular vertex and some critical vertices. From left to right: regular vertex ($n_{Lk^-(v)} = n_{Lk^+(v)} = 1$), a minimum vertex ($n_{Lk^-(v)} = 0, n_{Lk^+(v)} = 1$), a maximum vertex ($n_{Lk^+(v)} = 0, n_{Lk^-(v)} = 1$), a 1-saddle ($n_{Lk^-(v)} = n_{Lk^+(v)} = 2$) and a 2-saddle ($n_{Lk^-(v)} = n_{Lk^+(v)} = 3$).	85
4.10	The three possible unfoldings (right) of a 2-saddle (left) into two 1-saddles, with the level lines of the new saddles.	86

5.1	Shape preserving transformations (affine 5.1(b), near-isometric 5.1(c), under-sampling 5.1(d)) of a triangulated surface and the resulting <i>height functions</i> and their critical vertices.	94
5.2	Some PL functions and their critical vertices.	96
5.3	Critical vertices due to the sampling of a non-differentiable function.	98
5.4	Source vertices v_1 (5.4(a)) and v_2 (5.4(b)) of a triangulated surface (most geodesic-distant vertices).	99
5.5	\mathcal{F}_1 , \mathcal{F}_2 and \mathcal{F} sets on a triangulated surface.	100
5.6	f PL function on two triangulated surfaces and its <i>extra</i> critical vertices.	102
5.7	General perturbation strategy of an arbitrary PL function f to a simple Morse function.	103
5.8	f_1 (top) and f (bottom) critical vertex repartition on the Cyberware horse triangulated surface.	105
5.9	Visual comparison between the <i>height</i> PL Morse function (top) and the f PL Morse function (bottom) after shape preserving transformations (affine, near-isometric and under-sampling).	108
5.10	Discrete level line $f^{-1}(f(v))$ around a minimum vertex of f (red).	110
5.11	Discrete level lines (in red) on a triangulated surface.	111
5.12	Discrete sub-level sets (light grey) and discrete contours (red) of a regular vertex (left) and of non-degenerate critical vertices (from left to right: minimum, maximum and 1-saddle) on their star.	112
5.13	The first three iterations of algorithm 3 on an arbitrary distinctly valued PL Morse function (one iteration per row).	112
5.14	Connecting the equivalence classes in the Reeb graph: discrete contour configuration (top row) and corresponding connection in the Reeb graph (bottom row). From left to right: minimum vertex (case 1), regular vertex (case 2), maximum vertex (case 3), splitting 1-saddle (case 4(a)) and merging 1-saddle (case 4(b)).	114
5.15	Feature vertices of several triangulated surfaces.	120
5.16	f PL Morse function computation time (including feature vertex extraction and perturbation) with regard to the number of triangles of the input triangulated surface.	122

5.17	Enhanced topological computation time with regard to the number of triangles of the input triangulated surface.	123
5.18	Symbolic embedding of the enhanced topological skeleton of f on several triangulated surfaces.	124
5.19	f PL Morse function invariance and robustness towards shape-preserving transformations (affine and near-isometric), under-sampling and random noise and the symbolic embedding of the resulting enhanced topological skeleton.	125
6.1	Control skeleton driven shape pose editing of a 3D character (Sébastien Dominé, NVIDIA).	131
6.2	Control skeleton segment positioning by contour contraction.	134
6.3	Several curvature measurements on the hand triangulated surface: discrete mean curvature κ_H (left), discrete Gaussian curvature κ_G (middle) and curvature index κ_I (KvD92) (right).	136
6.4	g (left) and \hat{g} (middle) with regard to f (X axis) for several edges of $\mathcal{R}(f)$ and the corresponding surface regions (right).	137
6.5	Control skeleton after contour contraction (left), contour constrictions (middle) and control skeleton after contour constriction computation (right).	139
6.6	Control skeleton of other triangulated surfaces.	140
6.7	Control skeleton robustness towards triangulated surface degradations: random noise (middle) and under-sampling (right).	142
6.8	Zooming in the alien triangulated surface: feature vertex extraction implies the creation of segments in the control skeleton for each finger and toe.	143
6.9	Control skeleton of the hand triangulated surface (middle left), articulated skeleton (middle right) and resulting pose edition (right).	144
7.1	Feature vertices and f functions (left) and symbolic embeddings of the related enhanced topological skeletons (right) of several triangulated surfaces.	155
7.2	Segmentation of a hand triangulated surface model into its Reeb charts.	156

7.3	Reeb chart geometry similarity estimation process: each chart \mathcal{C}_i is mapped under ϕ_i to its canonical planar domain \mathcal{D} and its stretching signature λ_{ϕ_i} is computed with regard to the area distortion introduced by ϕ_i	157
7.4	Disk-like Reeb chart unfolding signature computation.	158
7.5	Example of <i>stretching signatures</i> for altered versions of the thumb chart and other primitive charts.	159
7.6	Annulus-like Reeb chart unfolding signature computation. .	160
7.7	Structural distortion on two visually similar and topology equivalent Reeb patterns.	162
7.8	Enumeration of the disk-like Reeb charts of a Reeb pattern. Left: Original Reeb pattern P . Right: Reeb pattern P' after the closure of the unique boundary component of P . Disk-like Reeb charts are in blue.	164
7.9	Enumeration of the annulus-like Reeb charts of a Reeb pattern. Left: Original Reeb pattern P . Right: Reeb pattern P' after the closure of the unique boundary component of P . Annulus-like Reeb charts are in red.	165
7.10	Structural signatures of a genus-0 and a genus-1 Reeb pattern.	167
7.11	Expansion example: for a given combination of topology equivalent Reeb patterns $((P_1, P_2)$ and (P_3, P_4)), the expansion algorithm tries to expand as much as possible the common sub-graphs (in bold in the second row).	168
7.12	Expansion process from two topology equivalent Reeb patterns \mathcal{P}_1 and \mathcal{P}_2 . Steps 1, 2 and 3 are depicted by blue arrows.	169
7.13	Reeb chart (bright colors) and Reeb pattern (dark colors) matching between a boy and a centaur. Unmatched charts are black.	171
7.14	SHREC 2007 data-set snapshot: one class per row: ants, armadillos, bearings, birds, bustes, chairs, cups, fishes, four-legs, glasses, hands, humans, mechanics, octopuses, planes, pliers, springs, tables, teddies and vases.	172
7.15	SHREC 2007 query-set snapshot.	173
7.16	A query from the SHREC 2007 query-set (a centaur) and the top-7 results retrieved by our system.	174
7.17	A genus-1 query from the SHREC 2007 query-set and the top-7 results retrieved by our system.	175

7.18	Another query from the SHREC 2007 query-set (an Armadillo composed with a plane) and the top-7 results retrieved by our system. Notice Armadillos have been retrieved despite near-isometric transformations.	175
7.19	Average Normalized Discounted Cumulative Gain (NDCG) vectors for Reeb pattern unfolding (RPU), (BMSFo6) (ERG) and (CDS*05) (CORNEA) on the SHREC 2007 data-set.	177
7.20	Contribution of the Reeb chart unfolding signature to the performances.	177
7.21	Random noise on a SHREC 2007 query.	178
7.22	Robustness evaluation of RPU with a noisy version of the SHREC 2007 query-set.	179
7.23	Modeling a <i>cow-horse</i> by example: the user selects on the query, then on the retrieved results (second row) the Reeb patterns to be exchanged.	181
7.24	Modeling a new woman 3D shape by example, composing both synthetic and scanned data.	181
8.1	Initial geodesic PL function to feature vertices (top left), curvature index κ_T (top middle) and PL function with the curvature term (top right).	191
8.2	Segmentation abstract graph and resulting segmentation.	193
8.3	Segmentation abstract graph (after perception heuristics based simplification) and resulting segmentation.	195
8.4	Segmentation abstract graph (after simplification) and the hierarchy of resulting decompositions (coarsest to finest).	196
8.5	Finest level of hierarchy for various 3D shapes.	198
8.6	Comparison between the hierarchical decompositions obtained with the algorithm of Katz et al. (KLT05) (top row) and our algorithm (bottom row).	200
8.7	Compatible hierarchies of decomposition for two similar objects.	200
8.8	Finest decompositions of the hand triangulated surface after affine and near-isometric transformations, under-sampling and random noise.	201
8.9	Comparison to other topology driven segmentation algorithms (left: (ZMT05), middle: (LKAo6)).	202

8.10 Example frames of the <i>horse</i> dynamic triangulated surface (top row) and its quadratic edge-length deviation (bottom row).	206
8.11 Elastic boundaries for several dynamic triangulated surfaces (reported on the first frame).	207
8.12 Examples of dynamic triangulated surfaces and the corresponding segmentation abstract graphs.	209
8.13 Comparison with the functional dynamic 3D shape decomposition of (LWCo6) (left) and of (dATTS08) (middle).	209
8.14 Animation reverse engineering: a dynamic triangulated surface is generated (top row) from a predefined skeleton of a static mesh (top row, left). Edge-length deviation (bottom row) and resulting decomposition (bottom row, right). Elastic node location error: 0.44%	211

INTRODUCTION

“Un dessin vaut mieux qu’un long discours.”
French popular proverb.

As underlined by this popular proverb (“*a picture is worth a thousand words*”), graphical models played an important role in the development of human communication and society, from the very first cave paintings, to the first ancient logograms (like the Egyptian hieroglyphs), to the modern typography, drawings, pictures, movies and nowadays virtual three-dimensional worlds.

With the rapid recent development of 3D technologies, dealing with the acquisition (with laser range scanners for example), the rendering (with accelerated graphical processing units) and the distribution of 3D data on the Web (with collaborative “*Web 2.0*” portals¹), 3D data become a major interactive graphical experience, yielding new applications having an important impact on nowadays society (see figure 1.1) like medical imaging, security systems, computer-aided design (most of manufactured products are now designed in 3D), scientific simulation, cultural heritage, film industry, video games, etc.

This 3D related technological activity now leads to a so-called “*3D data big bang*” through the exponential growth of professional, scientific, medical or personal collections of 3D shapes. This continuous source of 3D content imposes the invention of computational techniques for its efficient processing: edition, comparison and retrieval, compression, watermarking, etc. Beyond these recently formulated problems, a common fundamental issue is that of the efficient computational representation of either synthetic or acquired 3D shapes.

In this thesis, we formulate the problem of 3D shape efficient computational modeling, propose solutions by exploring differential topol-

¹See Dassault Systems collaborative 3D web portal: <http://www.3dvia.com>.



Figure 1.1 – Ubiquity of 3D data in nowadays technological society. From left to right and top to bottom: Valence hospital MRI system, facial authentication system (©Aurora), computer-aided design of a cellphone (©Dassault Systems), simulation of an earthquake wave propagation (MSB*03), digital acquisition of Michelangelo's David (Stanford University Michelangelo project), animated picture promotional poster (Blender Foundation), screen-shot of a video game ("The legend of Zelda: Twilight Princess", ©Nintendo).

ogy techniques and address some of the above problems dealing with 3D shape processing.

1.1 SCIENTIFIC CHALLENGES

As discussed in the following chapters, existing solutions for the raw digital representation of either synthetic or acquired 3D shapes suffer from high variability towards shape-preserving transformations (like affine or isometric transformations) or representation degradations (like random noise or sampling variation). As a consequence, shape processing tasks might lead to distinct results considering a 3D shape before and after these canonical transformations. The first scientific issue to be addressed by 3D shape modeling is thus the definition of intrinsic shape descriptions, enabling their consistent processing.

Moreover, for the technical reasons underlined in the following chapters, raw digital 3D shape representations mainly come in a discrete flavor. Another scientific issue of 3D shape modeling is then to transform these discrete digital representations into symbolic representations (mathematical shape abstractions) in order to enable efficient calculus and processing. This step is often referred to as shape abstraction.

The two scientific challenges of 3D shape modeling can then be summarized as follows:

- The definition of computational methods for 3D shape intrinsic characterization;
- The definition of computational methods for 3D shape symbolic (or expressive) abstraction.

Intrinsic and expressive 3D shape modeling is a fundamental requirement for the resolution of 3D shape processing problems. Thus, in addition to the problem of intrinsic and expressive 3D shape modeling, we focus in this thesis of three shape processing research problems related to Computer Graphics and Computer Vision, to assess the validity of the proposed shape modeling approach.

Abstraction models for shape handling

In 3D modeling systems, 3D shapes come most of the time in a raw discrete digital representation which is hard to handle for interactive use. An open research problem is the definition of abstraction models (as well as

automatic methods for their computation) that enable a direct intuitive and interactive handling for user shape edition, like deformation or pose edition.

Partial 3D shape retrieval

To help human users browsing the continuously growing collections of 3D shapes, efficient content-based retrieval techniques have to be proposed, towards purely shape based *Google*-like systems. A particular challenging problem is that of shape retrieval according to partial similarity. In this context, the system is expected to retrieve, automatically and in interactive processing times, the entries of a 3D shape collection which share visually similar sub-parts with an input query shape, even if they visually differ globally. This subject has recently focused the shape modeling research community due to its inherent difficulty with regard to the canonical global similarity estimation problem.

Semantic-oriented shape decomposition

Finally, the automatic decomposition of a 3D object into its constitutive functional parts is a research problem whose resolution would definitely contribute to perception related shape processing tasks (like geometry lossy compression) and human user geometry enhancing tasks (like shape painting or texture mapping).

1.2 CONTRIBUTIONS

In this thesis, we bring contributions to each of the following problems.

Intrinsic and expressive 3D shape modeling

First, we explore differential topology to propose a shape modeling technique based on Reeb graphs. In this context, the input shape, represented by a triangulated surface, is automatically and intrinsically abstracted into a graph structure, abstracting geometry related problems into graph problems (well known by computer scientists). We propose a new Reeb graph computation technique which provides output data-structures (referred to as *enhanced topological skeletons*) more compliant with the Reeb graph definition in the smooth setting than previous approaches, enabling not only the study of the topology of the shape but also of its geometry. To guarantee this symbolic representation's invariance and expressive properties,

we propose a computation technique for the definition of an intrinsic and *expressive* piecewise linear function (based on the notion of shape feature points) which drives the graph construction. To assess the validity of the modeling approach, we propose to use it to address the following research problems.

Abstraction models for shape handling

In the context of 3D shape handling, we propose a solution based on a skeletal embedding of the Reeb graph. We take advantage of the “*geometry-aware*” version of Reeb graphs we propose to define geometrical calculus on the Reeb graph. Then, a control skeleton is automatically embedded in the Euclidean 3D space by contour contraction and contour constriction computation, whatever the pose of the input shape is. Such a control skeleton helps human users to intuitively modify the pose of 3D shape by grabbing and rotating the branches of the skeleton, as illustrated in a presented interactive shape pose edition application.

Partial 3D shape retrieval

In the context of partial 3D shape retrieval, we propose an automatic fast and robust partial similarity estimation solution which takes advantage of the theoretical properties of Reeb graphs to improve both the 3D shape description and comparison step. Extensive quantitative experiments show the performance superiority of our approach with regard to the methods competing in the 2007 edition of the international SHape REtrieval Contest (SHREC 2007) partial retrieval track.

Semantic-oriented shape decomposition

Finally, in the context of automatic *semantic-oriented* shape decomposition, we propose a fast and robust solution based on the formerly introduced enhanced Reeb graphs. With such a modeling approach, we move the segmentation problem to a more abstract level, where human perception theory related heuristics are easily integrable. Then, to assess the validity of this approach, we discuss the qualitative properties of the resulting decompositions according to the evaluation criteria enumerated in a recent survey. Moreover, we present early results for the functional decomposition of a 3D shape based on Reeb graphs which exploits the temporal information provided by time-varying 3D data representing 3D shape an-

imations. Early qualitative and quantitative experiments valid the pertinence of this approach with regard to existing solutions.

1.3 OUTLINE

The rest of this manuscript is laid out as follows.

First, in chapter 2, we introduce the existing solutions for raw 3D shape representation and detail the technical reasons for the widespread use of surface meshes. We also show the variability issues of this representation, motivating research for more sophisticated techniques.

Then, in chapter 3, we survey recent techniques for intrinsic and symbolic 3D shape modeling and give arguments for the use of differential topology based techniques.

Chapter 4 concentrates all the theoretical preliminaries of the following contributions.

Chapter 5 details our intrinsic and expressive 3D shape modeling strategy and provides algorithms, data-structures and experimental results.

The rest of the manuscript is dedicated to the application of our modeling strategy to the previously mentioned shape processing problems.

Chapter 6 details our strategy for automatic control skeleton extraction from triangulated surfaces, provides experimental results and presents an application to interactive shape pose edition.

Chapter 7 details our approach for partial 3D shape retrieval. Extensive experimental results are provided and an application to interactive modeling by example is proposed.

Chapter 8 presents our strategy for the automatic functional decomposition of a 3D shape, both from human perception theory related heuristics and from the analysis of time-varying 3D shapes. Experiments are presented and discussed and an application to animation reverse engineering is proposed.

Finally, chapter 9 concludes this manuscript by summarizing the contributions of this thesis, enumerates remaining open problems and propose directions for future research.

1.4 FRENCH INTRODUCTION

Comme souligné par le proverbe "*Un dessin vaut mieux qu'un long discours*", les modèles graphiques ont joué un rôle important dans le développement de la communication et de la société humaines, depuis les premières peintures pariétales préhistoriques, aux premiers logogrammes (comme les hiéroglyphes égyptiens), à la typographie moderne, la peinture, la photographie, le cinéma et aux mondes virtuels tridimensionnels actuels.

Avec le développement récent des technologies 3D, traitant de l'acquisition (avec les scanners lasers par exemple), du rendu (avec les cartes d'accélération graphique) et de la distribution de données 3D sur le Web (avec les portails collaboratifs "*Web 2.0*"), les données 3D constituent à présent une expérience graphique interactive majeure, aboutissant à de nouvelles applications ayant un impact fort sur la société actuelle, comme l'imagerie médicale, la sécurité informatique, la conception assistée par ordinateur (la plupart des objets manufacturés sont aujourd'hui conçus en 3D), la simulation scientifique, la préservation du patrimoine culturel, l'industrie du cinéma, les jeux vidéos, etc.

Cette activité technologique liée à la 3D conduit à présent à un phénomène baptisé le "*big bang 3D*" au travers de la croissance exponentielle des collections professionnelles, scientifiques, médicales ou personnelles de formes 3D. Cette source continue de contenu 3D impose l'invention de nouvelles techniques informatiques pour en assurer un traitement efficace : pour l'édition, la comparaison et l'indexation, la compression, le tatouage, etc. Au delà de ces problèmes formulés récemment, une difficulté fondamentale commune est celle de la représentation informatique efficace des formes 3D, aussi bien acquises que synthétiques.

Dans cette thèse, nous formulons le problème de la modélisation informatique efficace de forme 3D, proposons des solutions en explorant la topologie différentielle et traitons certains des problèmes susmentionnés liés au traitement de forme 3D.

1.4.1 Défis scientifiques

Les solutions existantes pour la représentation brute numérique des formes 3D (acquises ou synthétiques) souffrent de grande variabilité face aux transformations préservant la forme (comme les transformations affines ou isométriques) ou aux dégradations de la représentation (comme le bruit aléatoire ou la variation d'échantillonnage). Par conséquent, les tâches de traitement de forme peuvent aboutir à des résultats distincts si

l'on considère une forme 3D avant et après ces transformations canoniques. Le premier problème scientifique auquel est confronté la modélisation de forme 3D est donc celui de la définition de représentations de forme intrinsèques, rendant possible des traitements cohérents.

De plus, pour des raisons techniques évoquées dans les chapitres suivants, les représentations brutes de formes 3D sont principalement discrètes. Un autre problème scientifique de la modélisation de formes 3D est donc la définition de transformations de ces représentations numériques discrètes en représentations symboliques (abstractions mathématiques de forme) pour rendre utilisables des techniques de calcul existantes pour leur traitement.

Les deux défis scientifiques de la modélisation de formes 3D peuvent donc être résumés comme suit :

- La définition de méthodes informatiques pour la caractérisation intrinsèque de forme 3D ;
- La définition de méthodes informatiques pour l'abstraction symbolique (ou expressive) de formes 3D.

La modélisation intrinsèque et expressive de formes 3D est un pré-requis à la résolution des problèmes de traitement de forme 3D. Ainsi, en plus du problème de la modélisation de formes 3D, nous nous intéressons également dans cette thèse à trois problèmes de recherche en traitement de formes 3D, liés à l'Informatique Graphique et à la Vision par Ordinateur, ce pour attester de la validité de notre approche de modélisation.

Abstraction pour la manipulation de formes

Dans les systèmes de modélisation 3D, les formes 3D sont représentées de manière brute, sous une forme discrète numérique qu'il est difficile de manipuler pour un usage interactif. La définition de modèles d'abstraction de formes (ainsi que de techniques automatiques pour leur calcul) permettant une manipulation directe et intuitive pour l'édition de forme par un utilisateur (déformation ou édition de pose) est un problème de recherche résolument ouvert.

Indexation 3D par similarité partielle

Pour aider les utilisateurs dans leur parcours des collections de formes 3D, des techniques efficaces pour l'indexation et la recherche par le contenu doivent être proposées, dans le but de fournir des systèmes *à la*

Google basés uniquement sur des considérations de forme. En particulier, l'indexation et la recherche de formes 3D selon des critères de similarité partielle constituent un problème de recherche ouvert et difficile. Dans ce contexte, le système doit retrouver, automatiquement et dans des temps interactifs, les entrées d'une collection de formes 3D qui partagent des sous-parties visuellement similaires avec une forme présentée en requête, ce même si les formes diffèrent globalement. Ce sujet a récemment attiré l'attention de la communauté de recherche en modélisation 3D, notamment en raison de sa difficulté inhérente en comparaison avec le problème d'estimation de similarité globale.

Décomposition de forme orientée sémantique

Enfin, la décomposition automatique d'une forme 3D en ses parties constitutives fonctionnelles est un problème de recherche dont la résolution contribuerait indéniablement aux tâches de traitement de forme liées à la perception (comme la compression avec pertes) ou encore aux tâches interactives d'enrichissement de formes 3D (comme le coloriage de formes ou le placage de textures).

1.4.2 Contributions

Dans cette thèse, nous apportons des contributions à chacun des problèmes suivants.

Modélisation intrinsèque et expressive de formes 3D

Tout d'abord, nous explorons la topologie différentielle et proposons une technique de modélisation basée sur les graphes de Reeb. Dans ce contexte, la forme en entrée, représentée par une surface triangulée, est automatiquement et intrinsèquement abstraite sous la forme d'un graphe ; les problèmes géométriques sont alors abstraits sous la forme de problèmes sur les graphes, largement étudiés en Informatique. Nous proposons pour cela une nouvelle technique de calcul de graphes de Reeb qui fournit en sortie des structures de données plus fidèles à la définition des graphes de Reeb dans le continu, rendant possible non seulement l'étude de la topologie de la forme mais aussi de sa géométrie. Pour garantir l'invariance de cette représentation symbolique et son expressivité, nous proposons une technique pour le calcul d'une fonction linéaire par morceaux invariante et expressive (basée sur la notion de point caractéristique d'une forme) utilisée pour guider la construction du graphe.

Pour témoigner de la validité de notre approche de modélisation, nous proposons de l'utiliser dans les problèmes de recherche suivants.

Abstractions pour la manipulation de formes

Dans le contexte de la manipulation de formes 3D, nous proposons une solution basée sur un plongement du graphe de Reeb sous la forme d'un squelette. Nous tirons parti de la version des graphes de Reeb que nous avons proposée pour définir des méthodes de calcul géométrique sur les graphes de Reeb. Ensuite, un squelette de contrôle est automatiquement plongé dans l'espace euclidien 3D par contraction de contour et par calcul de constrictions de contour, ce quelque soit la pose de la forme en entrée. Un tel squelette de contrôle permet ensuite à un utilisateur de modifier intuitivement la pose d'une forme 3D en articulant les branches du squelettes, comme illustré par un exemple d'application à l'édition interactive de pose d'objets 3D déformables.

Indexation par similarité partielle

Dans le cadre de l'indexation par similarité partielle de formes 3D, nous proposons une technique d'estimation de similarité partielle automatique, rapide et robuste. En particulier, nous tirons parti des propriétés théoriques des graphes de Reeb pour améliorer à la fois les étapes de description et de comparaison de formes 3D. De nombreuses expériences quantitatives montrent la supériorité de notre approche face aux méthodes participant à l'édition 2007 du concours international d'indexation de formes 3D (SHREC 2007) dans la catégorie indexation par similarité partielle.

Décomposition de forme orientée sémantique

Enfin, dans le contexte de la décomposition automatique *orientée sémantique* de forme, nous proposons une solution rapide et robuste basée sur les graphes de Reeb améliorés précédemment introduits. Avec cette approche de modélisation, nous déplaçons le problème de segmentation à un niveau plus abstrait, où les heuristiques basées sur la théorie de la perception humaine sont facilement intégrables. Ensuite, pour témoigner de la validité de cette approche, nous discutons les propriétés qualitatives des décompositions obtenues selon les critères d'évaluations mentionnés dans un état de l'art récent. De plus, nous présentons des résultats préliminaires sur la décomposition fonctionnelle d'une forme 3D basée sur les

graphes de Reeb, qui exploite l'information temporelle fournie par des données 3D variant dans le temps représentant des animations de formes 3D. Des expériences préliminaires qualitatives et quantitatives valident la pertinence de notre approche par rapport aux solutions existantes.

1.4.3 Plan du manuscrit

Le reste du manuscrit est structuré comme suit. Tout d'abord, dans le chapitre 2, nous introduisons les solutions existantes pour la représentation brute de formes 3D et détaillons les raisons techniques de l'utilisation majoritaire des surfaces maillées. Nous montrons également les problèmes de variabilité de cette représentation, motivant la recherche de techniques de modélisation plus sophistiquées. Ensuite, dans le chapitre 3, nous dressons l'état de l'art des techniques récentes pour la modélisation intrinsèque de formes 3D et donnons des arguments pour l'utilisation de techniques basées sur la topologie différentielle. Le chapitre 4 concentre tous les préliminaires théoriques des contributions présentées ensuite dans ce manuscrit.

Le chapitre 5 détaille notre stratégie de modélisation intrinsèque et expressive de formes 3D, décrit les algorithmes et structures de données associées et présente des résultats expérimentaux.

Le chapitre 6 détaille notre stratégie pour l'extraction automatique de squelettes de contrôle à partir de surfaces triangulées, fournit des résultats expérimentaux et présente une application à l'édition interactive de pose d'objets 3D déformables.

Le chapitre 7 détaille notre approche pour l'indexation 3D par similitude partielle. De nombreux résultats expérimentaux sont fournis et une application à la *modélisation par l'exemple* est proposée.

Le chapitre 8 présente notre stratégie pour la décomposition fonctionnelle automatique de formes 3D, à la fois à partir d'heuristiques basées sur la théorie de la perception humaine et à partir de l'analyse de formes 3D variant dans le temps. Des résultats expérimentaux sont présentés et discutés et une application à l'ingénierie inverse d'animations est proposée.

Enfin, le chapitre 9 conclue ce manuscrit en résumant les contributions apportées par cette thèse, énumère certains problèmes demeurant ouverts et propose quelques directions de recherche future.

Chaque chapitre sera agrémenté d'un résumé en français.

3D SHAPE RAW REPRESENTATIONS

2

CONTENTS

2.1	VOLUME VS BOUNDARY REPRESENTATIONS	15
2.2	IMPLICIT SURFACE REPRESENTATIONS	17
2.3	EXPLICIT SURFACE REPRESENTATIONS	18
2.3.1	Parametric surfaces	18
2.3.2	Surface meshes	19
2.4	FOCUS ON SURFACE MESHES	21
2.4.1	Design and acquisition of surface meshes	21
2.4.2	Surface mesh storage file formats	23
2.5	EXTRINSIC REPRESENTATION RELATED ISSUES	24
2.5.1	Representation variability	24
2.5.2	Applicative implications	25
2.6	CHAPTER CONCLUSION	26
2.7	FRENCH CHAPTER ABSTRACT	27

THIS chapter introduces existing solutions for the digital representation of 3D shapes, insisting on their specificity with regard to the acquisition/creation context and their target applications.

This short survey concludes with a special emphasis on surface meshes and highlights the main drawbacks of this widely used representation for shape processing tasks, motivating research for more sophisticated shape modeling techniques.

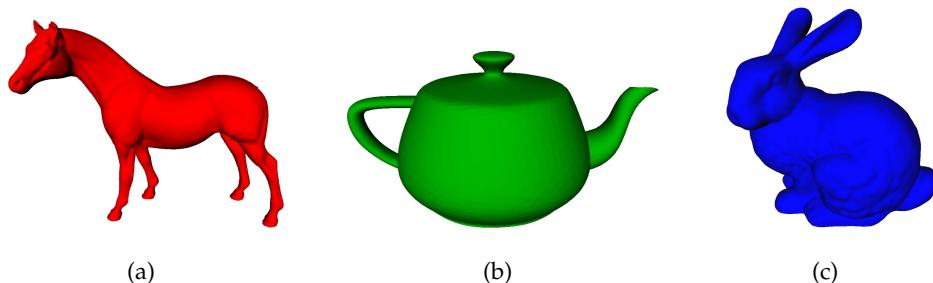


Figure 2.1 – Example of “famous” 3D shapes: the Cyberware horse (2.1(a)), the Utah teapot (2.1(b)) and the Stanford bunny (2.1(c)).

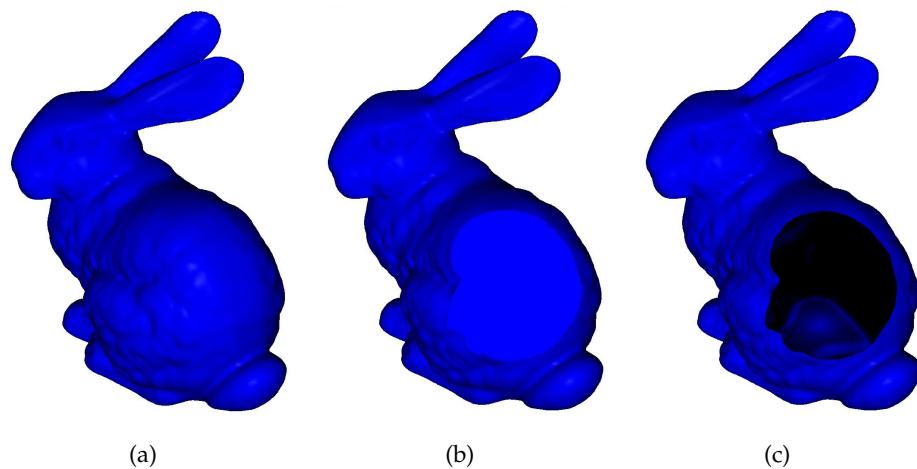


Figure 2.2 – Cutting the Stanford bunny 3D shape: cutting a volume representation (2.2(b)) and a boundary representation (2.2(c)).

2.1 VOLUME VS BOUNDARY REPRESENTATIONS

The computational representation of a scene in 3D related applications (video games, film industry, cultural heritage, scientific simulation, computer-aided design, security systems, medical imaging, etc.) requires the modeling of the objects composing the scene. The attributes of these digital objects can involve their shape, their material composition, their lighting characteristics and others.

Focusing on the shape of a digital object, several computational representations can be considered. They can be classified into two main categories: volume and boundary representations. In the first case 3D shapes are defined with regard to the volume of the object while in the second case, only the external boundary of the object intervenes in the shape representation, as illustrated in figure 2.2 (the inside of the object is hollow for boundary representations).

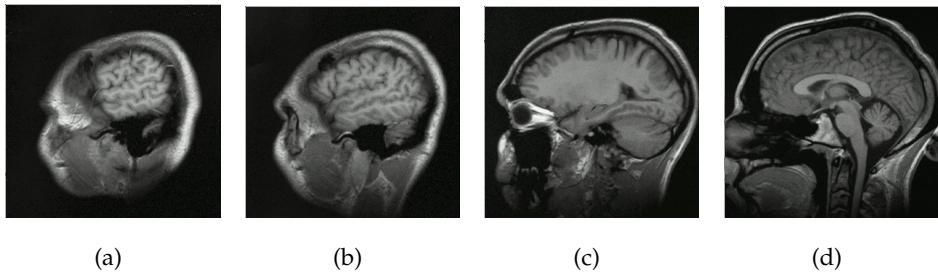


Figure 2.3 – Successive slices of a volume representation of a human head obtained by magnetic resonance imaging, showing the inner components of the brain.*

The *voxel* grid model is a representative example of volume representations. In this context, the Euclidean 3D space is subdivided into uniform volume elements (“voxels”, the 3D analogue of pixels) which represent some numerical quantity (color, density, temperature). The enumeration of the voxels occupied by the object enables the definition of its shape. Voxel grids are particularly suited for the acquisition of 3D data which are not directly “visible” from an external observator, like in geological or medical imaging (see figure 2.3¹). Volume-based representations have two main advantages. First, they can efficiently model non-visible object’s characteristics like possible cavities (like the interior of the teapot in figure 2.1 for instance). Moreover, in the context of physical modeling (for scientific simulation, medical imaging or security computer vision for example), they enable the enhancement of the object’s digital representation with precise physical characteristics of its inner material (density, temperature, etc.).

Boundary representations on the other hand just represent the external “visible” boundary of an object, modeled by a surface. As a direct consequence, boundary representations require less memory footprint than volume based representations for an equivalent level of detail. Moreover, as the appearance of an object depends largely on its exterior, in most of Computer Graphics and Computer Vision applications, the external boundary of the object is a sufficient representation of its shapes. Furthermore, boundary surfaces are far easier to render or to acquire (with laser scanners for example). For these reasons, the boundary representation is a popular representation of 3D shapes and thus have largely drawn the shape modeling research community’s interest. As an illustration, 3D shape research repositories ([prior3](#), [gamo3](#), [aimo6](#)) are mainly composed

¹Pictures whose caption includes the character * are released under the terms of the GNU Free Documentation License.

of boundary representations. As a consequence, in the framework of this manuscript, even though most of the introduced concepts are extensible to volume representations, we will exclusively focus on boundary surfaces as 3D shape representations.

Boundary representations can be classified into two categories: implicit and explicit surfaces. We introduce them in the following sections with a special emphasis on explicit surfaces. For a larger survey of surface representations in geometric modeling, we defer the reader to (HGoo, Ilio5).

2.2 IMPLICIT SURFACE REPRESENTATIONS

Implicit surfaces are defined according to a particular mathematical form (BB97). Given a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, the implicit surface \mathbf{S} associated to f is the set of points \mathbf{p} of coordinates (x, y, z) which satisfy the following equation:

$$f(x, y, z) = 0 \Leftrightarrow \mathbf{S} = f^{-1}(0) \quad (2.1)$$

Such surfaces are referred to as implicit in the sense that the function f does not explicitly describe the surface, but only implies its existence. The following function defines an example of trivial implicit surface (a sphere of radius r):

$$f(x, y, z) = x^2 + y^2 + z^2 - r^2 \quad (2.2)$$

Implicit surfaces can be built by compositing simple primitives like spheres or ellipsoids (with trivial f functions). *Blobby Molecules* (Bli82) or *Soft Objects* (WW86) are examples of such primitives.

Because of their compact mathematical description and their good differentiable properties, implicit surfaces are particularly suited for simulating physical processes (TPBF87) or for modeling smooth objects (TO99).

For this latter reason, implicit surfaces have shown to be interesting candidates for shape representation in the context of 3D computer-aided design or 3D modeling and animations systems.

However, their efficient display on computer (thanks to the Graphics Processing Unit) without conversion to an explicit form is still an open research issue (KSNo8). Moreover, because implicit surfaces are abstract mathematical concepts, they cannot be used directly to model acquired geometry of real life objects without consequent processings (SSP94).

2.3 EXPLICIT SURFACE REPRESENTATIONS

On the contrary to implicit surfaces, explicit surface representations are explicitly defined by some particular mathematical form. We consider two types of explicit surfaces: parametric surfaces and surface meshes.

2.3.1 Parametric surfaces

Given a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, a parametric surface \mathbf{S} is the set of points \mathbf{p} of coordinates (x, y, z) lying on the image by f of its base domain Ω_f parameterized by (u, v) :

$$\mathbf{S} = \{f(u, v) = (f_x(u, v), f_y(u, v), f_z(u, v)) \quad / \quad (u, v) \in \Omega_f \subset \mathbb{R}^2\} \quad (2.3)$$

The following function defines an example of trivial parametric surface (a sphere of radius r):

$$f(u, v) = (f_x(u, v), f_y(u, v), f_z(u, v)), \quad u \in [-\frac{\pi}{2}, \frac{\pi}{2}], v \in [-\pi, \pi] \quad (2.4)$$

$$f_x(u, v) = r \times \cos(u) \times \cos(v) \quad (2.5)$$

$$f_y(u, v) = r \times \cos(u) \times \sin(v) \quad (2.6)$$

$$f_z(u, v) = r \times \sin(u) \quad (2.7)$$

Examples of common parametric surfaces are super-quadratics, Bézier and B-spline surfaces. For instance, Bézier surfaces have been introduced by the french engineer Pierre Bézier in 1972 with the aim of making the creation of surfaces easier and more intuitive for car designers. In this context, the design of a surface is determined by the definition a set of $(n + 1) \times (m + 1)$ control points $\mathbf{k}_{i,j}$ ($u \in [0, 1], v \in [0, 1]$):

$$f(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) \mathbf{k}_{i,j} \quad (2.8)$$

where $B_i^n(u)$ is a Bernstein polynomial:

$$B_i^n(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{(n-i)} \quad (2.9)$$

Nowadays, non-uniform rational B-splines (NURBS, a generalization of Bézier's work) are massively used as a fundamental tool for several surface editing operations in computer-aided design and modeling software (see figure 2.4) since they enable designers to intuitively and rapidly define complex shaped smooth surfaces by just positioning a network of control points.

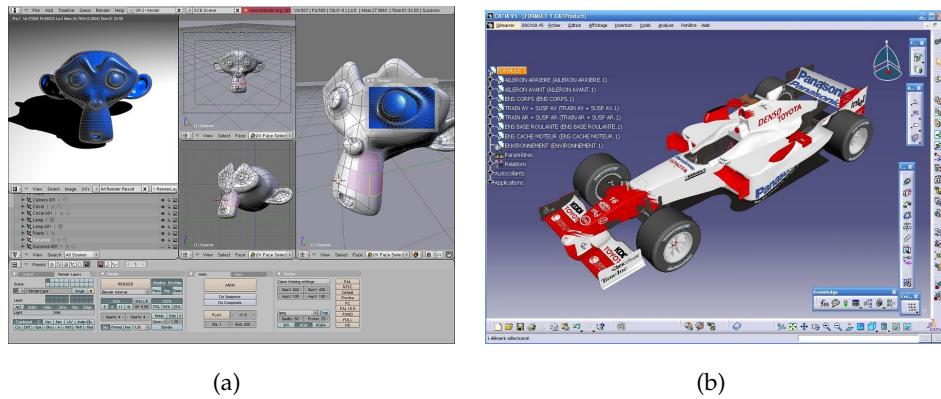


Figure 2.4 – 3D modeling and animation (Blender 2.4(a)) and computer-aided design (CATIA V5 ©2.4(b)) systems are examples of software using parametric surfaces as core shape representation.*

However, like implicit surfaces, parametric surfaces are abstract mathematical concepts. Thus, if they are very useful in the context of computer-aided design, they are inappropriate to represent acquired geometry of real life objects (of a priori unknown geometry and topology). Moreover, their rendering requires a post-process which consists in sampling f base domain to produce a surface mesh.

2.3.2 Surface meshes

A surface mesh is a piecewise linear approximation of a surface. The explicit representation is given by the definition of an arbitrary number of sample points \mathbf{p} of coordinates (x, y, z) lying on the surface. Then the surface is locally approximated by polygon primitives (like triangles or quadrangles) which connects the sample points into a graph structure, denoted as a polygon mesh M . The polygon primitives are denoted as faces f , the faces' segments as edges e and the edges' extremities as vertices v (one vertex per sample point \mathbf{p}):

$$M = \{V, E, F\} \quad (2.10)$$

$$F = \{f_1, f_2, \dots, f_{N_F}\} \quad (2.11)$$

$$E = \{e_1, e_2, \dots, e_{N_E}\} \quad (2.12)$$

$$V = \{v_1, v_2, \dots, v_{N_V}\} \quad (2.13)$$

For technical reasons related to rendering (see following remarks about rendering), the considered primitives are most of the time triangles (see figure 2.5). The surface mesh is then denoted as a triangulation (we defer the reader to chapter 4 page 61 for a formal definition of triangulations).

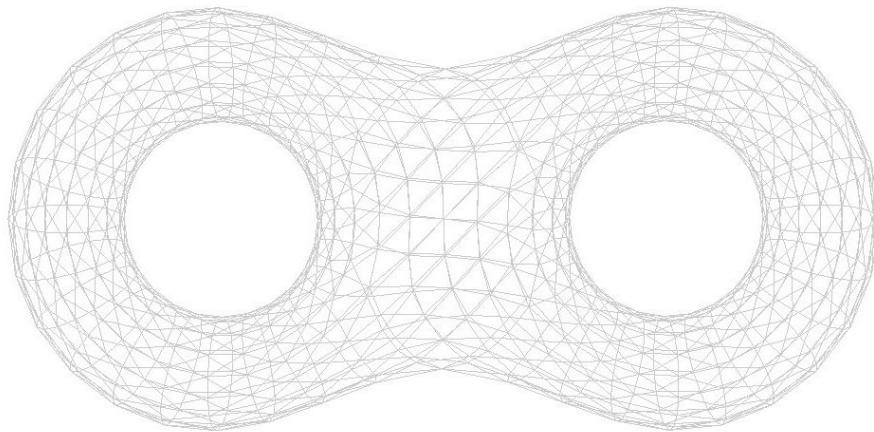


Figure 2.5 – Bi-torus represented by a surface mesh (rendered in wireframe mode).

Surface meshes are definitely the most popular surface representation in 3D related applications. First, current hardware rendering computer devices (Graphics Processing Units) are designed to render surfaces made explicit in the form of a polygon mesh. In particular, as only 3 points are necessary for the definition of a plane in the Euclidean 3D space, rendering systems systematically subdivide complex polygon primitives into triangles before processing them in their pipeline.

Thus, any computer-aided design software has to convert surfaces into polygon meshes for their interactive rendering. Surface piecewise linear approximation is particularly straightforward for parametric surface representations (defined by a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$): the polygon mesh is the image by f of a uniform grid lying on its base domain, as illustrated in figure 2.4(a) where a NURBS based surface is rendered into a polygon mesh form. Consequently, as surface piecewise linear approximation is a must feature of 3D modeling software, these systems also export surfaces (for interoperability with other systems) in the form of surface meshes.

Furthermore, in the context of geometry acquisition (with laser range scanners), acquired point clouds can be converted rapidly into surface meshes (see section 2.4.1 for more details).

Consequently, surface mesh is a surface representation which is both convenient in the context of geometry design, acquisition and interactive rendering. Moreover, surface meshes can be seen as the smallest “common denominator” of all explicit surface based 3D shape representations, thanks to the facility of their conversion to polygon meshes.

For these reasons, surface meshes have now become a *de facto* standard for shape storage and exchange in 3D related applications. This

3D shape representation has consequently focused the shape modeling research community's interest, as testified by its quasi omni-presence in research shape repositories (prio3, gamo3, aimo6). Consequently, in the framework of this manuscript, we will exclusively consider 3D shapes represented in the form of an explicit surface mesh (and particularly triangulations).

2.4 FOCUS ON SURFACE MESHES

2.4.1 Design and acquisition of surface meshes

Surfaces meshes can be first created in two distinct contexts: shape computer-aided design and shape acquisition.

From shape design to surface meshes

Computer-aided design systems represent surfaces either in an implicit or an explicit form.

In the case of implicit surfaces, the *polygonization* step may be computationally expensive since the surface points' locations cannot be directly deduced from the surface definition. This problem is similar to that of the extraction of surface meshes from volume representations. The *marching cubes* algorithm (LC87) is a standard solution. It first subdivides the 3D space into a uniform voxel grid and identifies voxels intersecting the surface. Then, depending on the type of voxel intersection, specific strategies are employed to compute surface approximating triangles. Unfortunately, since voxels have constant size, a trade-off has to be made as for the resolution of the grid between the output's expected level of detail and the extraction computation time.

On the contrary to implicit surfaces, parametric surfaces present the advantage to explicitly capture in their definition the surface points' locations in space. Thus, a quadrangle mesh can be extracted straightforwardly considering the image by f of a uniform grid defined on its base domain. Quadrangles can eventually be divided into triangle couples if a triangulation is needed.

From shape acquisition to surface meshes

Many technical solutions have been proposed for the acquisition of 3D geometry from real life objects. Even if Computer Vision practitioners have



Figure 2.6 – Lab's Minolta VI-300 laser range scanner.

made a significant effort to develop less expensive alternative techniques based on passive sensors, laser range scanners (see figure 2.6) are now a standard solution for 3D geometry acquisition.

The acquisition process is composed of several sub-steps. Most of laser scanners use an optical camera and laser rays to measure by triangulation the distance to sample target points on the object's boundary to produce *depth images*. Depth images are 2D views of the object accompanied with the depth information (the distance from the object to the scanner) for each pixel. Several depth images have to be taken from multiple viewpoints to recover the whole object. Then, point clouds are generated for each depth image and the resulting point cloud sets must then be registered for alignment (using the Iterative Closest Point algorithm (BMK92) for example). The intermediate output of the acquisition is thus a point cloud, a set of sample points lying on the boundary surface of the object.

Surface mesh automatic reconstruction from point sets is an active research topic which is far beyond the scope of this manuscript. We defer the reader to (MM99) for a detailed survey and to (KBHo06, SLS*o6, BHGS06) for recent automatic techniques. In practice, interactive techniques based on the canonical Delaunay triangulation are mostly employed and can be considered as part of the acquisition operation.

```
#VRML v2.0 utf8
Shape{
  geometry IndexedFaceSet {
    coord Coordinate {
      point [
        # sample point coordinate (x,y,z) list
      ]
    }
    coordIndex [
      # face list: vertex indices (face separator: "-1")
    ]
  }
}
```

Figure 2.7 – Standard layout for surface mesh storage with VRML v2.0.

2.4.2 Surface mesh storage file formats

An abundant quantity of specification file formats have been provided for the storage of surface meshes, most of them in an ASCII form. Here are few examples: the 3D Object File Format (OFF), the Wavefront OBject format (OBJ), the Stanford University PoLYgon format (PLY), the GNU Triangulated Surface format (GTS) or the Virtual Reality Modeling Lan-gage (WRL). Their storage strategies are very similar. First, the vertices' positions in 3D (sample points (x, y, z) coordinates) are listed in an unorganized way. Then, the polygon primitives (the faces) are also defined by an unorganized list. Each entry of this list defines a face by the indices of its vertices (indexed by their order of appearance in the sample point list).

Only one of these formats have been subject to international standardization: the Virtual Reality Modeling Language (WRL file extensions). VRML has been originally specified based on the Open Inventor API paradigm developed by Silicon Graphics Inc. and has been first proposed to the International Organization for Standardization (ISO) in 1994 by the Web3D Consortium in order to provide a standard technique for modeling 3D interactive experiences over the Web. Its successor (X3D, based on XML) has been approved for standardization in 2005 but has still less success than the second version of VRML (proposed in 1997). Among its numerous features, VRML version 2 includes surface mesh modeling.

Its integration to the hierarchical scene description philosophy of VRML is the following. 3D objects are specified in a “Shape” dedicated

node. This node type has several attributes, including its material appearance (for lighting modeling) and its geometry. The “geometry” attributes can be valued with pre-defined shape primitive nodes or with an “IndexedFaceSet” node. This latter node has two main attributes which are the coordinates of the sample points (“coord” field, valued with a “Coordinate” node) and the face specification (“coordIndex” valued with an array of vertex integer indices, using the “-1” joker value for face separation). A standard layout for surface mesh storage with VRML v2.0 is shown in figure 2.7.

2.5 EXTRINSIC REPRESENTATION RELATED ISSUES

2.5.1 Representation variability

The major problem of existing *raw* surface representations (implicit or explicit) is the fact that they are extrinsic surface descriptions (BJ85). Thus they shall be affected by important representation variability with regard to shape canonical transformations. This observation is the main motivation of the shape modeling community research effort.

Either implicit or explicit representations are extrinsic in the sense they do not represent the surface geometrical structure but just its embedding in the Euclidean 3D space. For example, considering a parametric surface, modifying the surface embedding by a translation will result in a different parametric surface definition, through an addition of the translation components to the expressions of f_x , f_y and f_z (see equation 2.3). The same remark also goes to surface meshes where the sample points’ locations vary (x , y , z coordinates). This simple example shows that extrinsic surface representations (including surface meshes) suffer from variability under trivial transformations like affine transformations.

This issue gets even more concerning when more sophisticated transformations are considered like isometric transformations, elastic deformations or surface smoothing for example. Figure 2.8 illustrates this observation by showing a given surface after common transformations, resulting in distinct surface mesh representations. Particularly to surface meshes, different point sampling strategies will also lead to totally different representations for a same surface, as shown in figure 2.8(d) where the number of sample points is twice less than in figure 2.8(a).

This variability in surface representation is a major issue for shape

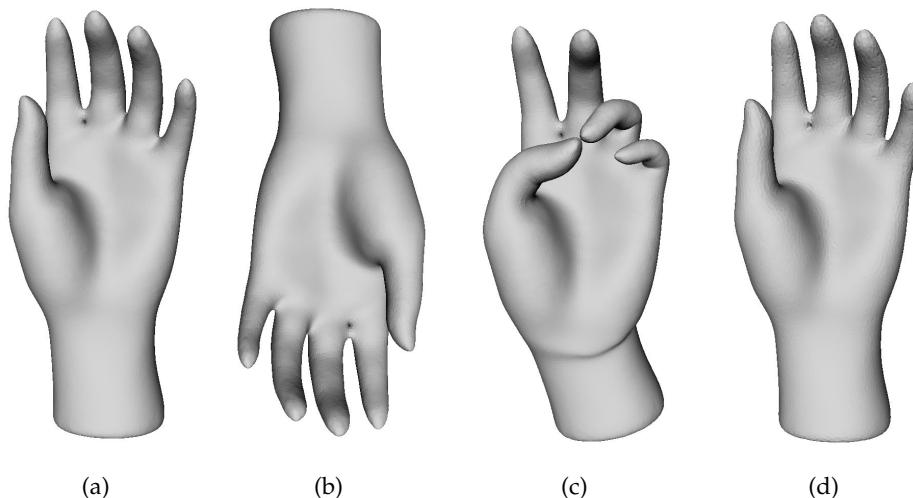


Figure 2.8 – Trivial transformations applied on the same 3D shape results in different surface mesh representations: affine transformation 2.8(b), near-isometric transformation 2.8(c) and sampling modification 2.8(d).

processing tasks, where surfaces should have consistent computational representations despite such transformations.

2.5.2 Applicative implications

From an applicative point of view, variability in surface representation shall match certain expectations and shall not be a function of the surface embedding but of its intrinsic characteristics. Efficient intrinsic representations should be invariant to application specific transformation groups and should only express variability towards other pre-defined ones. Here are few examples:

- **Shape comparison:** In the case of deformable objects (having possible distinct poses, like the hand in figure 2.8(a)), shape intrinsic representations for shape comparison should be invariant to isometric transformations (including affine transformations). In the case of elastic objects, the shape representation should be invariant to elastic deformations. As most of real life objects are not elastic, most of the time, intrinsic representations should efficiently express the quantity of elasticity of considered transformations. Moreover, representations should be robust towards surface degradation, like noise or sampling variation. Finally, similar shapes should have similar surface representations. Shape comparison can be viewed as the

most demanding application in term of shape intrinsic representation computation.

- **Shape watermarking:** in the context of shape copyright protection, the intrinsic representation at the origin of the surface watermarking should be invariant to affine and isometric transformations and should be particularly robust to surface local and global noise.
- **Shape deformation:** in the case of computer-aided editing of shape pose (for animation), the shape representation should also be invariant to the actual pose of the input surface. Moreover, it should efficiently describe the remarkable shape features in order to help the designer in his/her task.

These few examples well illustrate the intrinsic surface representation applicative requirements. First, intrinsic surface representations should be invariant to a pre-defined set of transformation groups. Second, they should be robust towards surface degradation (like noise or sampling variation). Third, they should provide an expressive shape representation.

The definition and computation of such intrinsic surface descriptions from raw extrinsic descriptions (in particular from surface meshes) is the main goal of the 3D shape modeling research community. An important research effort has been provided the last decade in this direction and will be surveyed in the following chapter.

2.6 CHAPTER CONCLUSION

In this chapter, we presented existing solutions for the computational representation of 3D shapes, in particular representing the object surface boundary. We showed surface mesh is the smallest common denominator of surface based representations (either implicit or explicit) and is the most popular shape representation for storage and exchange, thanks to its versatility towards computer-aided design, shape acquisition or interactive rendering.

However, surface mesh is an extrinsic shape representation and suffers from important variability with regard to shape canonical transformations, which is a major issue for most of shape processing tasks, where surfaces should have consistent computational representations despite such transformations.

In the following chapter, we survey last decade's research effort towards the definition and computation of intrinsic and convenient surface representations for 3D shape processing tasks.

2.7 FRENCH CHAPTER ABSTRACT

Dans ce chapitre, nous avons présenté les solutions existantes pour la représentation informatique de formes 3D. Celles-ci peuvent être classées en deux catégories: les représentations volumiques et surfaciques. En particulier, les représentations surfaciques peuvent elles mêmes être classées en deux groupes: les surfaces implicites et explicites. Les surfaces maillées appartiennent à ce dernier groupe et constituent la représentation la plus couramment utilisée pour le stockage ou l'échange, grâce à sa polyvalence envers la conception assistée par ordinateur, l'acquisition de forme ou le rendu interactif.

Cependant, les surfaces maillées sont une représentation extrinsèque de forme et souffrent d'une importante variabilité face à des transformations de forme standards, ce qui constitue un problème majeur pour le traitement de formes 3D, où les surfaces doivent être représentées en mémoire de manière cohérente malgré ces transformations.

Dans le chapitre suivant, nous dressons donc l'état de l'art concernant la définition et le calcul de représentations intrinsèques de surface pour le traitement de formes 3D.

3

3D SHAPE MODELING STATE OF THE ART

CONTENTS

3.1	MOTIVATIONS AND TAXONOMY	31
3.2	GEOMETRY MODELING	32
3.2.1	Spectral modeling	32
3.2.2	Conformal geometry based modeling	37
3.2.3	Riemannian geometry based modeling	41
3.3	TOPOLOGY MODELING	45
3.3.1	Curve skeletons	45
3.3.2	Surface mesh segmentation	49
3.3.3	Differential topology based modeling	54
3.4	CHAPTER CONCLUSION	57
3.5	FRENCH CHAPTER ABSTRACT	58

This chapter reviews recent advances in term of intrinsic modeling of 3D shapes represented by surface meshes. This survey is split into two main categories to underline the complementarity of the under-laying concepts: geometry and topology based modeling. Three representative classes of techniques are presented for each category.

In particular, we give a special emphasis on topology modeling for surface structural representation, showing its interest from both an applicative and a more fundamental point of view.

3.1 MOTIVATIONS AND TAXONOMY

We showed in the previous chapter that the commonest computational representation of 3D shapes is the surface mesh, mainly because of its versatility towards computer-aided design, shape acquisition or interactive rendering. However, this surface representation is extrinsic and suffers from high variability towards standard shape transformations such as affine or isometric transformations. Moreover, surface sampling strategies for mesh generation are most of the time also extrinsic.

For these reasons, it is first necessary to define computational surface representations that are intrinsic to the surface in order to perform further shape processing. This is the subject of 3D shape modeling research.

The intrinsic characteristics of a surface can be interpreted from two different (but complementary) angles of view: geometry and topology.

Geometry and topology can be seen as two dual concepts related to shape as melody and rhythm are related to music. The topology of a surface is the set of its characteristics that do not vary under smooth transformations. Consequently, the topology of a surface is an intrinsic property. However, the topology is insufficient to fully describe a surface as any two surfaces of identical topology can be transformed from one to the other by a smooth transformation (homeomorphism). The geometry of a surface on the other hand is precisely the set of its intrinsic characteristics that vary under smooth transformations.

The geometry of a surface is often referred to as its shape. However, from a practical point of view, as we will show in this chapter, fundamental limitations of geometry-based shape modeling are related to topology and can be possibly addressed by topology-based modeling. Moreover, psychological research (Bie87) claims that the perceived object structure is an important element in human shape understanding. Thus, as topology covers surface structure description, it is a good candidate for *expressive* shape modeling in the context of shape understanding applications.

In this chapter, we review 3D shape modeling techniques for surface mesh intrinsic description according to the above classification. First, we present some representative geometry based modeling techniques and then examples of topology related methods. In particular, thanks to its relation to shape structure (which is particularly interesting for the applicative problems addressed later on this document), we will focus on topology modeling, which is the core of the research presented in the rest of this manuscript.

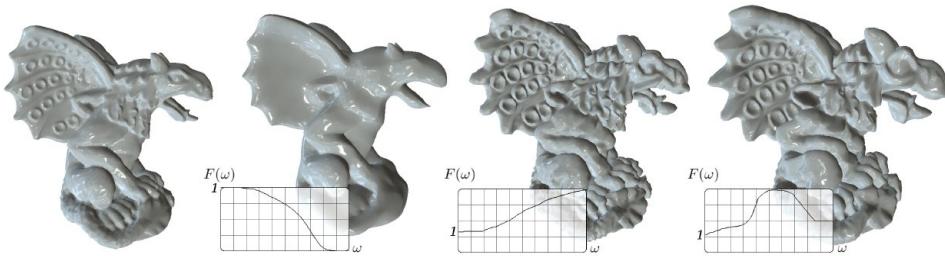


Figure 3.1 – Common signal processing filters (low pass, enhancement and band-exaggeration) applied on the manifold harmonics of a surface mesh (VLo8).

3.2 GEOMETRY MODELING

3.2.1 Spectral modeling

“Can one hear the shape of a drum?” (Kac66)

Signal processing formalism

Even if the general answer to this surprising question is “no” (BCDS94) (a given spectrum does not uniquely describe a surface), for most of cases, the answer is “yes”. Consequently, a significant research effort (Tau95, Lévo6, VLo8, ZvKDo8) has been dedicated recently to the transposition of signal processing models (like in acoustics) to the geometry analysis of *manifolds* (mathematical concept of *curved space* which generalizes the notion of surface; we defer the reader to chapter 4 page 61 for a formal definition of manifolds), as illustrated in figure 3.1 in the context of geometry filtering.

In particular, the original idea of spectral modeling is to infer intrinsic geometrical surface characteristics by computing its *spectral transform*. In acoustics, a spectral transform can be used to determine the energy reparation of a sound among harmonics. To do so, the temporal sound signal is commonly decomposed through the canonical Fourier Transform into a sum of sinusoidal signals, called basis functions (which form an orthonormal basis) and the spectrum of the signal is computed by considering the phase and the amplitude of the basis functions.

In the case of surfaces, the definition of proper function bases requires caution. In particular, the fundamental requirement of a function basis for spectral decomposition is orthonormality. In the case of the canonical Fourier Transform, the employed sinusoidal functions are the eigenfunctions of the $\partial^2/\partial t^2$ operator (where t is time). The extension of the $\partial^2/\partial t^2$ operator to manifolds is the Laplace operator Δ (see next subsection). As other Hermitian operators, the Laplacian’s eigenfunctions,

denoted as *manifold harmonics*, are orthonormal and thus can be used as a function basis for spectral transform computation (Lévo6, VLo8). The eigenfunctions and eigenvalues of the Laplacian are all the pairs (H^k, λ_k) that satisfy:

$$-\Delta H^k = \lambda_k H^k \quad (3.1)$$

On a closed curve, the eigenfunctions H^k of the Laplace operator define the function basis of the Fourier Transform. On a square, they correspond to the function basis of the Discrete Cosine Transform, used in JPEG compression framework.

To consider the Laplace operator on surface meshes, one can interpret the geometry of the surface as a linear combination of basis functions applied on the n sample points \mathbf{p} of coordinate (x, y, z) of the surface mesh:

$$\mathbf{p} = \sum_{i=1}^n \mathbf{p}_i \Phi^i \quad (3.2)$$

Computing the manifold harmonic transform of the function \mathbf{p} means projecting \mathbf{p} from the Φ^i basis (geometry) onto the H^k basis (frequency) through the inner product (we defer the reader to (VLo8) for practical details). Thus, the manifold harmonic transform maps a point \mathbf{p} to a set of vectors $[\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \dots, \tilde{\mathbf{p}}_m] \in \mathbb{R}^3$ with m the number of computed harmonics:

$$\tilde{\mathbf{p}}_k = \langle \mathbf{p}, H^k \rangle \quad (3.3)$$

To apply the operations processed in the spectral domain back to the geometry, an inverse manifold harmonic transform also needs to be introduced:

$$\mathbf{p} = \sum_{k=1}^m \tilde{\mathbf{p}}_k H^k \quad (3.4)$$

To make a full analogy with acoustics, in the case of a temporal sound signal, the H^k functions (the eigenfunctions of the Laplace operator) in equation 3.4 are sinusoidal functions and the $\tilde{\mathbf{p}}_k$ vectors are the Fourier coefficients.

As the spectral description of the surface mesh is intrinsic to its geometrical structure (a vertex location is a function of its direct neighbors' locations in the geometry basis Φ^i), it is a good candidate for several shape processing tasks (like geometry filtering as illustrated in figure 3.1).

In the next subsections, we focus on the Laplacian operator for surface meshes and show its utility on its own, besides spectral analysis, for shape modeling applications.



Figure 3.2 – Spectral modeling applied to texture filtering (VL07).

Laplacian operator on surface meshes

The Laplace operator played an important role in physics for the understanding of wave propagation, heat flow, electrostatics or fluid mechanics.

In the case of manifolds, the definition of the Laplacian enables the introduction of differential coordinates (Soro6), which reflects the curved nature of the manifold. In particular, the geometry basis functions Φ^i are defined with regard to differential coordinates (leading to the linear combination in equation 3.2). Several formulations have been proposed for the discretization of the Laplace operator (and thus for the geometry function basis definition) and are characterized by distinct properties (MDSBo2, Soro6, VL07, ZvKDo8).

Applications

Spectral and Laplacian based modeling covers a broad band of applications. A straightforward application is shape compression, where only the first coefficients of the manifold harmonics can be considered for shape representation (KG00). In the context of shape copyright protection, Ohbuchi et al. (OMT02) hide watermarks in the spectral domain of the surface mesh. As the spectral representation of the surface is intrinsic to its geometry, it is also a good representation for shape comparison (EK03, RWP05, JZ06, JZ07, Rus07). Moreover, as shown by Vallet and Lévy (VL07), spectral modeling can also be considered with input func-

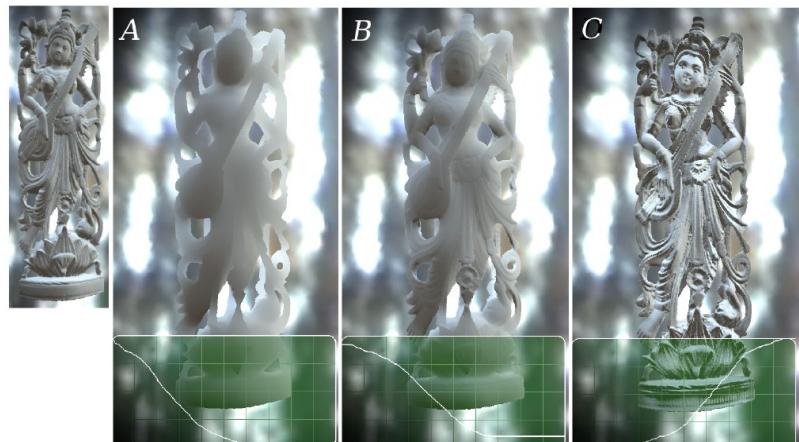


Figure 3.3 – Spectral modeling applied to shading filtering (VL07).

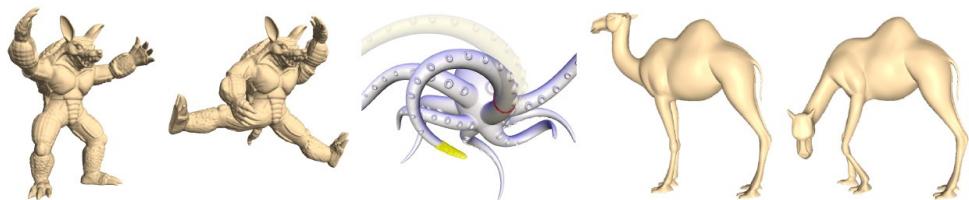


Figure 3.4 – Shape pose editing through Laplacian geometry modeling (LSLCO05).

tion basis related to other information than geometry, like texture (figure 3.2) or normals (for shading, figure 3.3).

The Laplacian surface representation (through differential coordinates) has shown to be an interesting shape description on its own, especially for constrained shape edition. For example, in the framework of shape deformation, Lipman et al. (LSLCO05) proposed an intuitive system where the user modifies the cartesian coordinates of a surface portion (in yellow in the center of figure 3.4). Then the shape deformation is automatically transferred by computing the new cartesian coordinates of the surface portion delimited by the red ring (center of figure 3.4) by solving a linear system based on the Laplace operator. The same reasoning can be applied for

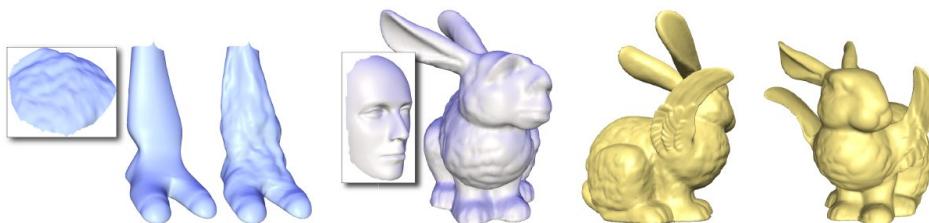


Figure 3.5 – Shape blending through Laplacian geometry modeling (SCOL*o4).

shape blending, where the differential coordinates of the blended shape is a linear combination of the differential coordinates of the two input shapes (figure 3.5). For a larger overview of applications of spectral and Laplacian based 3D shape modeling, we defer the reader to (Soro6, ZvKDo8).

Discussion

As pointed out by Zhang et al. (ZvKDo8), the spectral modeling of manifolds still needs theoretical investigations in order to have a better interpretation of existing and future operators. Moreover, it has been shown that all the natural properties of the Laplacian in the smooth setting cannot be preserved in the discrete setting with only one operator (WMKG07), which explains the diversity of existing discrete Laplacian formulations (Soro6, VL07, ZvKDo8). Furthermore, as pointed out by Dyer et al. (DZMCo7), spectral surface representations may suffer from variability when different yet similar shapes are considered and the most robust discrete Laplacians in this sense are also the most expensive to compute. Finally, as underlined in the introduction of this section, the spectrum of a surface does not uniquely describe it since the existence of distinct *iso-spectral* surfaces (having possibly distinct eigenfunctions yet) have been demonstrated (BCDS94). This observation motivates research for the definition of unique intrinsic shape descriptions, like conformal geometry based surface representations.

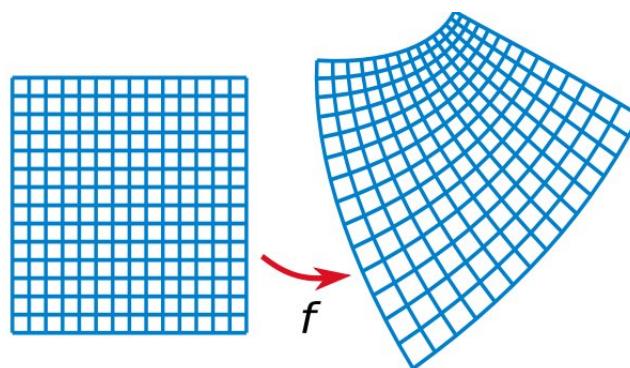


Figure 3.6 – A uniform grid and its image under a conformal map (angle-preserving).*



Figure 3.7 – *Conformal mapping of a face surface S to its canonical domain D . From left to right: original textured surface, conformally mapped texture, canonical domain (u, v) coordinate system, conformal coordinate system $S(u, v)$ (GY03).*

3.2.2 Conformal geometry based modeling

Conformal surface representation

Conformal geometry based modeling roots in parameterization analysis (FHo4). The purpose of conformal geometry modeling is to recover a parametric representation of the surface which is intrinsic and unique. A one-to-one mapping f from a surface \mathbf{S} to another surface \mathbf{S}' is *conformal* if the angle of intersection of every pair of intersecting arcs on \mathbf{S}' is the same as that of the corresponding pre-images on \mathbf{S} at the corresponding point (FHo4), as illustrated in figure 3.6 where grid arcs remain locally orthogonal. Conformal maps have many nice theoretical properties and have been used in several problems dealing with surface characterization (GY03).

Let f be a conformal map from a surface \mathbf{S} (see figure 3.7, left image) to a *canonical domain* \mathbf{D} (a surface with the same topology and having constant Gaussian curvature¹, see figure 3.7 middle left image). Let (u, v) be a coordinate system on the canonical domain \mathbf{D} (see figure 3.7, middle right image) and $\mathbf{S}(u, v)$ its pre-image by f on \mathbf{S} (see figure 3.7, right image). Then the following equations hold (GVo4):

$$\frac{\partial \mathbf{S}}{\partial u} \times \frac{\partial \mathbf{S}}{\partial v} = \lambda(u, v)n(u, v) \quad (3.5)$$

$$\Delta \mathbf{S}(u, v) = \kappa_H(u, v)n(u, v) \quad (3.6)$$

where Δ is the Laplace (manifold) operator, $n(u, v)$ is the function that maps a point of $\mathbf{S}(u, v)$ to its surface normal vector, and $\kappa_H(u, v)$ is the mean curvature. $\lambda(u, v)$ is called the *conformal factor* function (or also the *stretching factor*). The tuple (λ, κ_H) is the *conformal representation* of $\mathbf{S}(u, v)$.

Figure 3.7 shows an example of conformal parameterization of a facial surface and illustrates angle-preservation (intersecting arcs' angles are

¹We defer the reader to chapter 4 page 61 for a formal definition of Gaussian curvature

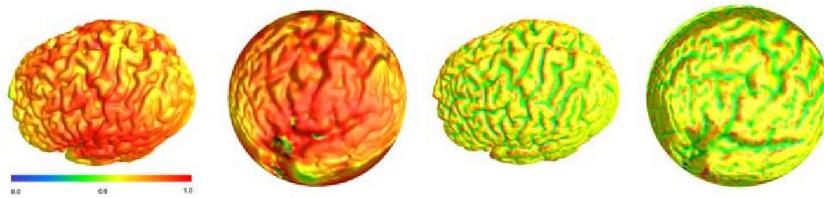


Figure 3.8 – Conformal representation of a brain surface. From left to right: $\lambda(u,v)$, same $\lambda(u,v)$ displayed on the canonical domain (the sphere), $\kappa_H(u,v)$, same $\kappa_H(u,v)$ displayed on the canonical domain (GVo4).

preserved on the canonical domain). Figure 3.8 shows a brain surface and its conformal representation.

Intuitively, equations 3.5 and 3.6 show that conformal flattening (mapping to a canonical domain) can be fully characterized by the input curvature of a surface $\kappa_H(u,v)$ and by the stretch $\lambda(u,v)$ one has to apply to conformally map the surface to the canonical domain. Conformal surface representations have several interesting properties.

First, the uniformization theorem (Abi81) states that any manifold surface can be conformally mapped to a *canonical domain* (a surface with the same topology and having constant Gaussian curvature). Thus a conformal representation of a surface can always be computed.

Second, and most important, given the *conformal representation* (λ, κ_H) of a surface \mathbf{S} and the boundary condition $\mathbf{S}(u,v)|_{\partial D}$, \mathbf{S} can be uniquely and completely reconstructed (see theorem 1 of (GVo4)). From an applicative point of view, this property is very attractive. For comparison, in the case of spectral modeling, a given frequency representation can potentially correspond to different (iso-spectral) surfaces.

Third, an important property of conformal representation is its stability. If the shape of a surface \mathbf{S} is slightly perturbed, then its conformal representation (λ, κ_H) will be perturbed only slightly (GVo4).

Fourth, isometric transformations are also conformal (FH04), which implies that conformal representations are robust towards usual shape transformations like isometries (WWJ*07, BCG08), as shown in figure 3.10 where the conformal factor for couples of near-isometric surfaces is represented.

As a conclusion, conformal surface representations can always be computed, they are unique, stable and invariant to isometric transformations. Thus, they are good candidates for intrinsic and unique 3D shape modeling.

Conformal maps can be computed through a global optimization pro-

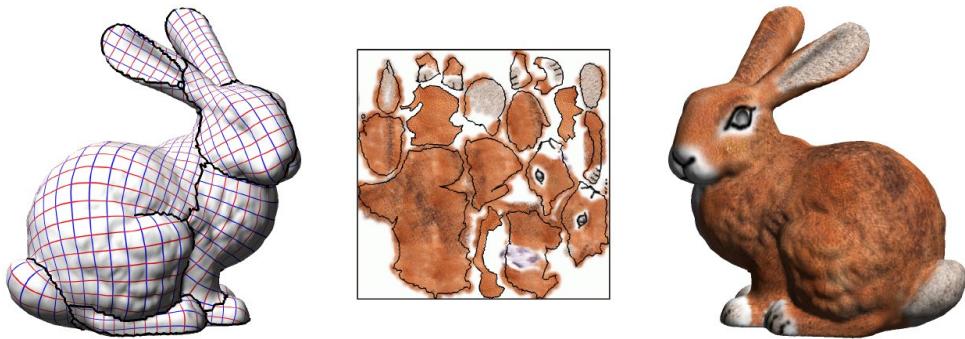


Figure 3.9 – *Chart atlas based conformal planar parameterization and texture mapping (LPRM02).* From left to right: local conformal coordinate systems $S(u,v)$, chart unfolding in the canonical domain D and corresponding texture mapping on the Stanford bunny 3D shape.

cess. A popular way for computing them is to solve the Cauchy-Riemann equation using the technique proposed by Lévy et al. (LPRM02), which efficiently computes a unique solution (conformal map) of this problem. Then $\lambda(u,v)$ computation is straightforward. It can be approximated by the ratio between the area of a local neighborhood on the surface (like a triangle in the case of triangle meshes) and the area of its image on the canonical domain (LGQ06). $\kappa_H(u,v)$ can be computed with existing techniques for curvature computation in the discrete setting (MDSB02).

Applications

To our knowledge, conformal geometry has mainly been used to address two shape modeling related research problems: surface parameterization and surface comparison. It has first drawn surface parameterization specialists' attention. As conformal maps are angle-preserving, they are particularly interesting in the context of texture mapping. Indeed, it is more convenient for an artist to draw geometry textures in a planar domain. Then, thanks to conformal geometry, the designed texture can be mapped back onto the geometry, guaranteeing no (angular) distortion in the shape of the drawings, as illustrated in figure 3.9. We defer the reader to (FHo4) for a comprehensive survey about planar parameterization and to (ZRS05, ZLS07, DG07, BCCB08) and (YBS04, SLMB05, RLL^{*}06, SPP08) for respectively linear and non-linear recent techniques.

As they are intrinsic, stable and unique, conformal representations are particularly interesting candidates for surface description in the context of 3D shape comparison. Gu and Vemuri (GV04) consider the distance computation between a surface S_1 and S_2 as an optimization problem, whose

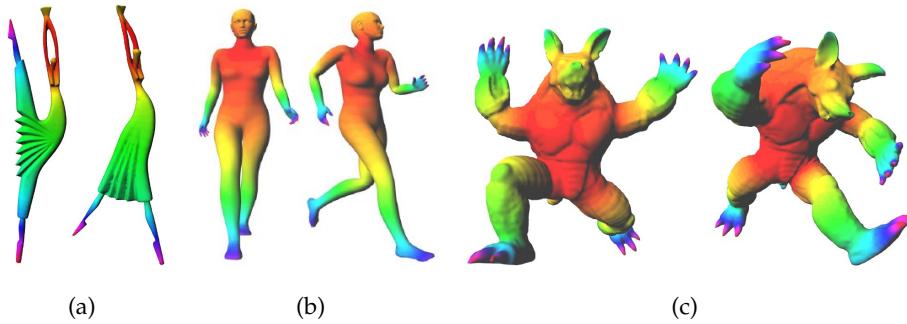


Figure 3.10 – Conformal factor $\lambda(u, v)$ repartition for humanoid 3D shapes represented by surface meshes of different topology (genus-1 and genus-0) (BCGo8).

solution is a diffeomorphism f^* that maps D_1 to D_2 while minimizing the following matching energy:

$$\begin{aligned} E(f^*) = & \int_{D_1} ||\lambda_1(u, v) - \lambda_2(f^*(u, v))||^2 dudv \\ & + \int_{D_1} ||\kappa_{H1}(u, v) - \kappa_{H2}(f^*(u, v))||^2 dudv \end{aligned} \quad (3.7)$$

Then the distance between S_1 and S_2 is given by $E(f^*)$. Similar strategies have been proposed later in the context of brain (GWC*04) or face (WWJ*06, WWJ*07) surface comparison. Recently, noticing that curvature computation in the discrete setting is particularly noisy and not very useful in practice, Ben-Chen and Gotsman propose to represent the surface only by its conformal factor for fast shape comparison in the context of shape retrieval (BCGo8).

Discussion

The main limitation of conformal representations is related to topology. Indeed, the uniformization theorem (Abi81) guarantees that any surface can be conformally mapped to a surface with the same topology and with constant curvature. This implies that conformal representations are convenient for the representation of constant topology surfaces. For example, in the context of shape comparison, surfaces will be required to have the same topology to be efficiently compared. Figure 3.10 shows that the repartition of the conformal factor function $\lambda(u, v)$ highly varies when the topology of considered surfaces vary (even if the represented shapes are similar): in the case of genus-0 surfaces (figures 3.10(b) and 3.10(c)) low λ values (in orange) are located in the center of the shape while for genus-1 surfaces (figure 3.10(a)), they are located around the topological handle. As a consequence, conformal factor based representations will

exhibit high distance between surfaces of distinct topology even if the underlying shapes are similar.

As pointed out by Wang et al. (WWJ^{*}06, WWJ^{*}07), the generalization of conformal representations to arbitrary topology surfaces seems to be a challenging direction of research. Ricci flows, recently introduced in 3D shape modeling by Gu et al. (GWK^{*}07), look to be a promising solution. Nevertheless, topology independent conformal representation seems to be a harder problem whose solution would certainly benefit shape modeling. Some solution elements can be thought however considering topology-based methods for surface decomposition, providing a full topology control of the input surfaces, as shown later on this document.

3.2.3 Riemannian geometry based modeling

Geometry modeling in shape space

Riemannian geometry modeling in shape space is certainly the most abstract (but still intuitive) geometry modeling paradigm presented so far. In this framework, a surface is represented as a point of an abstract curved-space (*manifold* space) of high dimension, called *shape space*, representing all the possible embeddings of the input surface. Then, shape analysis and processing are achieved by reasoning on this abstract space. This kind of approaches allows great shape modeling flexibility through the explicit definition of the transformation groups that need to be expressed or ignored.

Shape spaces have been largely studied by the mathematical community (Ken84, MM06) but only few authors have investigated it in computer science so far, mainly for shape deformation analysis (CEF98, KSMJ04). In particular, Klassen et al. (KSMJ04) introduced a computational approach to the study of spaces of curves. An extension of this work has been proposed for disk topology surfaces in the context of face comparison (SSJD08) while Kilian et al. (KMP07) presented a framework for surfaces of arbitrary topology.

In this latter work, the shape space S related to a surface \mathbf{S} represents all the possible embeddings of \mathbf{S} in \mathbb{R}^3 (KMP07). To perform Riemannian computation (of lengths, angles, curvatures, etc.) on the manifold shape space, S must be equipped with a *metric*.

Intuitively, for a given surface \mathbf{S} on S , the introduced metric enables to give a distance evaluation between two tangent directions on S . In particular, the definition of a metric is closely related to that of inner products

on the tangent spaces of \mathbb{S} . At this stage, the flexibility of Riemannian based geometry modeling allows to define specific inner products (and thus specific metrics) endowing shape modeling with applicative-specific constraints. The definition of the metric through that of the inner products determines the structure of the manifold shape space \mathbb{S} . For example, in the context of as-isometric-as-possible surface metamorphosis, Kilian et al. (KMP07) define an inner product between two tangent vectors such that two isometric surfaces have a null distance on \mathbb{S} . Consequently, near-isometric surfaces will have close locations on \mathbb{S} . Moreover, such a metric will efficiently express elastic deformations between shapes.

Once the shape space \mathbb{S} is equipped with a metric, one can perform calculus on it, like displacement measurement. In particular, it is possible to define the amount of energy $E(\phi)$ required to perform a displacement ϕ on \mathbb{S} with regard to the introduced metric. As shown in (Spi79), the displacement energy $E(\phi^*)$ is minimal if ϕ^* corresponds to a *geodesic* in \mathbb{S} . A geodesic is a path (along the manifold) of minimal length between two points of the manifold (we defer the reader to chapter 4 page 61 for a formal definition of geodesics). In a shape space \mathbb{S} , a geodesic from a surface \mathbf{S} to a surface \mathbf{S}' is consequently the optimal displacement of \mathbf{S} on \mathbb{S} that smoothly transforms \mathbf{S} into \mathbf{S}' . Here, the notion of optimality is given by the length of the displacement ϕ . In other words, a geodesic transforms *naturally* \mathbf{S} into \mathbf{S}' according to the geometrical structure of \mathbb{S} , whose intrinsic transformation properties have been made explicit through the mathematical definition of its metric.

The computation of geodesics is a must-feature of Riemannian geometry modeling techniques. They can be approximated through a global optimization process. For example, Kilian et al. (KMP07) propose a multi-resolution approach based on a Newton method for the minimization of $E(\phi)$.

Once the measurement of displacement on \mathbb{S} as well as geodesics can be computed, the input surface \mathbf{S} is efficiently modeled among all its possible embeddings represented and structured by \mathbb{S} . Then, a large class of shape processing problems can be considered.

Applications

On its own, the modeling of an input surface \mathbf{S} on a shape space \mathbb{S} is not particularly useful. As the geometrical structure of \mathbb{S} is inferred from the length of geodesics between surfaces, geometry modeling in shape

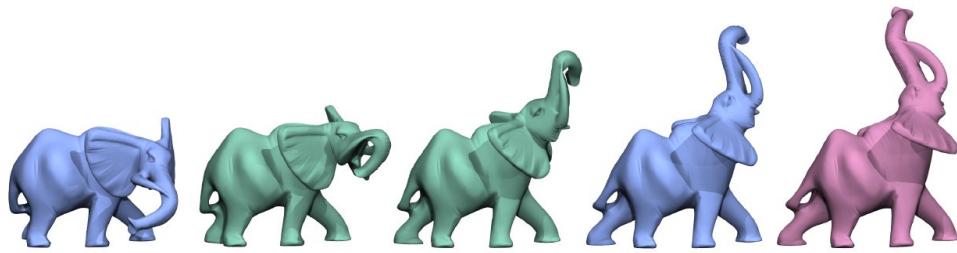


Figure 3.11 – Displacement interpolation and extrapolation (KMP07). The input blue surfaces (1^{st} and 4^{th}) have been interpolated through near-isometric transformations (in green, 2^{nd} and 3^{rd} surfaces) by geodesic computation in shape space. This displacement has then been extrapolated in shape space to produce the final surface (in red, 5^{th} one).

space is more convenient in applications where several input surfaces are considered.

For example, in the context of 3D face surface comparison, Srivastava et al. (SSJDo08) propose to compute geodesics between couples of faces and consider geodesics' lengths as the geometrical distances between the faces. Thanks to an appropriate metric definition, facial expression (elastic transformations) are particularly efficiently expressed: low distances are given between distinct expressions of a same individual while higher distances are given for distinct individuals.

Kilian et al. (KMP07) present an interesting panorama of applications based on the concepts of displacement interpolation and extrapolation.

Given two input surfaces \mathbf{S}_1 and \mathbf{S}_2 (blue surfaces in figure 3.11) accompanied with compatible coordinate systems $\mathbf{S}_1(u, v)$ and $\mathbf{S}_2(u, v)$, shape interpolation (green surfaces in figure 3.11) is performed by computing the geodesic $\phi^* : \mathbf{S}_1 \times [0, 1] \rightarrow \mathbf{S}$ such that:

$$\phi^*(\mathbf{S}_1(u, v), 1) = \mathbf{S}_2(u, v), \quad (u, v) \in \mathbf{D}_{\mathbf{S}_1}, \quad \mathbf{D}_{\mathbf{S}_1} = \mathbf{D}_{\mathbf{S}_2} \quad (3.8)$$

Then, shape extrapolation is determined by solving the Euler-Lagrange equation of energy to compute additional tangent vector fields that continue the geodesic (red surface in figure 3.11).

These two fundamental tools (interpolation and extrapolation) enables the development of interesting pose editing applications (like shape pose transfer by parallel transport or interactive shape pose editing by shape space exploration, see figure 3.12), providing not only the new embedding of the input surface, but also a family of surfaces smoothly joining the input surface to the output surface in shape space.

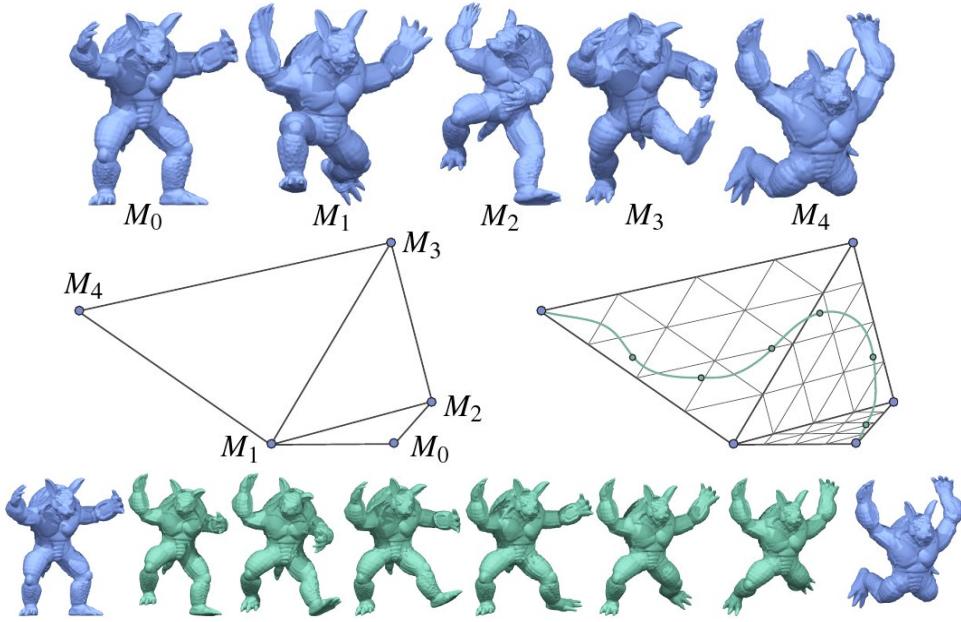


Figure 3.12 – Interactive shape pose editing by shape space exploration (KMP07): the user draws a path (in green) on a 2D-representation of the shape space (middle row, right) and the corresponding shapes (dots in 2D) are interactively computed (in green in bottom row).

Discussion

Although only few contributions have been brought so far to 3D shape modeling using Riemannian analysis, geometry modeling in shape space seems to be a very promising direction of research. However, this framework is affected by several important limitations.

First of all, as implicitly mentioned in equation 3.8, the computation of geodesics in shape space between a surface \mathbf{S}_1 and \mathbf{S}_2 first requires a faithful cross-parameterization $f : \mathbf{S}_1 \rightarrow \mathbf{S}_2$ that relevantly maps a point $\mathbf{S}_1(u, v)$ to a point $\mathbf{S}_2(u, v)$. This strong requirement is a major drawback of geometry modeling in shape space which considerably limits its applicative impacts.

In the context of as-isometric-as-possible surface deformation, Kilian et al. (KMP07) bypass this issue by considering input surfaces represented by isomorphic *compatible* surface meshes, guaranteeing a near-isometric mapping f between them. Such compatible meshes are pre-generated using a marker assisted correspondence algorithm (SP04).

Srivastava et al. (SSJD08), in the framework of face comparison, use a single-marker correspondence heuristic that automatically and robustly detects the tip of the nose. Then facial surfaces are parameterized using a curvilinear coordinate system based on the level sets and the gradient of

the geodesic distance (along the surface) to the latter marker. Two input surfaces to compare are then equipped with their own coordinate system $\mathbf{S}_1(u, v)$ and $\mathbf{S}_2(u, v)$. Finally, a function f^* which optimally maps the two coordinate systems ($f^*(\mathbf{S}_1(u, v)) = \mathbf{S}_2(u, v)$) is computed through an optimization process that takes level sets geometrical characteristics into account. This solution seems to be a valid hypothesis as the nose tip can be considered as a faithful reference point for faces.

However, in the general case, the problem of finding a relevant parameterization between the input surfaces remains an open issue.

Moreover, another major limitation resulting from this problem is the fact that the input surfaces must be homomorphic so that such a parameterization is guaranteed to even exist. This implies that geometry modeling in shape space, like conformal geometry, can only consider surfaces of constant topology, which highly reduces its applicative impacts.

This limitation is a common drawback of most of geometry based 3D shape modeling paradigms. As we will see further on this literature review, some solution elements can be thought however by considering differential topology based surface decomposition techniques that make topology edition possible for topology oblivious cross parameterization computation.

3.3 TOPOLOGY MODELING

The topology of a surface is the set of its characteristics that are invariant under any smooth transformation. Consequently, this intrinsic surface description is of major interest for 3D shape modeling applications.

Literally, *topology* is the study (*logos*) of places (*topos*). In other words, topology studies the fine and global structure of a space. Thus, the aim of topology modeling is to provide an intrinsic structural representation of the surface which efficiently captures its topological characteristics.

A significant research effort has been provided the last couple of decades for the structural description of 3D shapes represented as surfaces. We here survey the most significant types of topology related techniques.

3.3.1 Curve skeletons

A common paradigm for surface structural description is the shape skeleton, a 1-dimensional abstract representation of the surface structure. Such

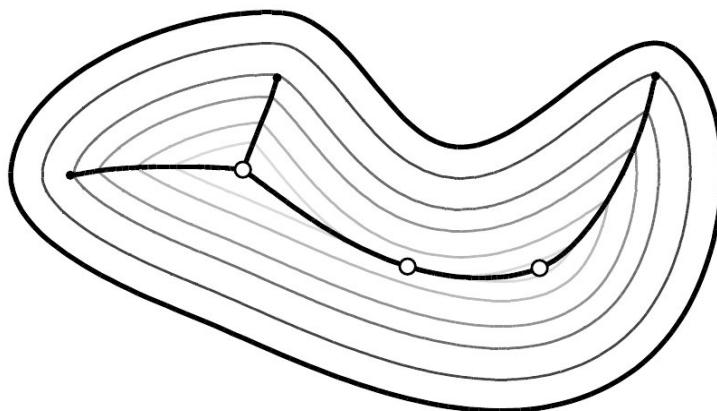


Figure 3.13 – Medial axis of a 2D planar shape represented by a boundary contour in bold (ABE07).

a high-level description enables to decompose, describe and recognize 3D shapes and has shown its importance in several application fields, like shape comparison, visualization, animation, compression, etc.

The curve skeleton is a modern variant of shape skeletons inspired by the medial axis, which has been extensively studied in the past in the case of 2D planar contours (Blu67, ABE07).

In 2D, the medial axis of a shape is a set of curves defined as the locus of points that have at least two closest points on the boundary of the shape (ABE07), as shown in figure 3.13. In the case of 3D shapes represented by surface meshes, the corresponding concept is the medial surface (GG90), because in addition to curves, it can also contain surface patches (CSM07). However, 1-dimensional representations are reported to be more convenient from an applicative point of view (DGB03, DJ06, CSM07, ATC*08). Consequently, most authors consider a pruned version of the medial surface called the curve skeleton, as shown in figure 3.14.

Let $\mathbf{V} \subset \mathbb{R}^3$ be a subset of the Euclidean 3D space bounded by a surface \mathbf{S} representing a 3D shape. The medial surface $\mathbf{M} \subset \mathbf{V}$ is the set of centers of the maximal spheres inscribed in \mathbf{V} :

$$\begin{aligned} \mathbf{M} = \{ \mathbf{p} \in \mathbf{V} | \exists \mathbf{p}_1, \mathbf{p}_2 \in \mathbf{S}, \mathbf{p}_1 \neq \mathbf{p}_2, \\ ||\overrightarrow{\mathbf{p} \mathbf{p}_1}|| = ||\overrightarrow{\mathbf{p} \mathbf{p}_2}|| = \min_{\mathbf{p}_k \in \mathbf{S}} \{ ||\overrightarrow{\mathbf{p} \mathbf{p}_k}|| \} \end{aligned} \quad (3.9)$$

The medial surface \mathbf{M} consists of surface elements, called *sheets*, and of 1-dimensional elements, called curves (GK04).

In equation 3.9, let \mathbf{p}_1 and \mathbf{p}_2 be the *boundary points* of \mathbf{p} . Moreover, let $f_B : \mathbf{M} \rightarrow \mathcal{P}(\mathbf{S})$ be the *boundary transform*, which maps each point $\mathbf{p} \in \mathbf{M}$ to its boundary point set. A point $\mathbf{p} \in \mathbf{M}$ lies on the curve skeleton \mathbf{C} of

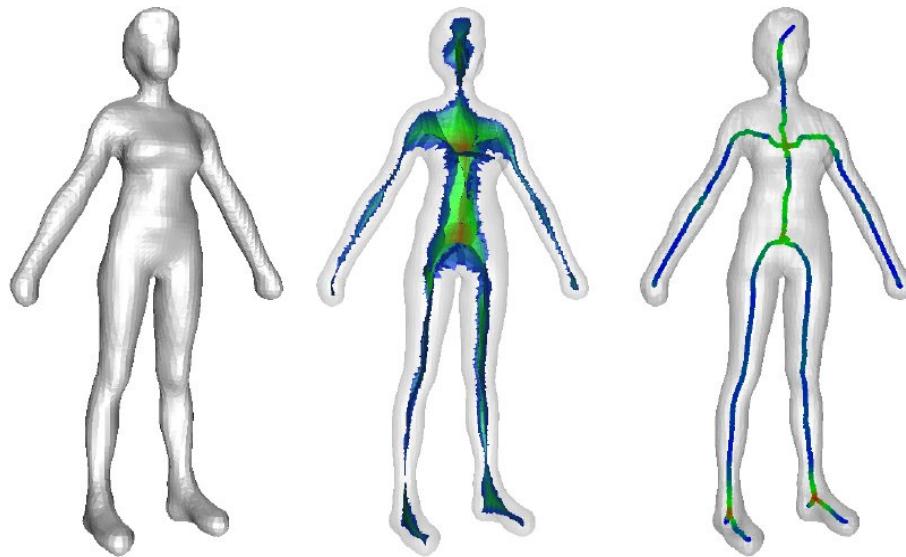


Figure 3.14 – A surface mesh, its medial surface and its curve skeleton (DJ06).

S if there exist at least two geodesics in **S** linking each couple of points of $f_B(\mathbf{p})$ (DJ06, RT07)². Here, the existence of several geodesics between the boundary points of \mathbf{p} guarantees that \mathbf{p} is *centered* in **V**.

An important number of curve skeleton extraction techniques can be found in the literature, either by reasoning on the volume bounded by the surface mesh (using volume thinning (MWLo2, WLo8) or distance field computation (MW003, CSYB05)) or by reasoning directly on the surface mesh (using active surfaces (SLSD07) or laplacian contraction (ATC*08) for example). We defer the reader to (CSM07) for a comprehensive survey.

Applications

Curve skeleton computation benefits to a wild spectrum of applications (CSM07). In medical visualization, curve skeletons can be used to specify collision free paths for virtual endoscopy navigation for example.

The high-level nature of the description provided by curve skeletons is particularly interesting for applications investigating the nature of the object represented by the 3D shape. In particular, in the framework of 3D shape recognition and retrieval, curve-skeletons have been used as high-level bias for surface comparison (SSGD03, BI04, CDS*05). In this context,

²We remind a geodesic between two points of a surface is the shortest path along the surface linking them (we defer the reader to chapter 4 page 61 for a formal definition of geodesics).

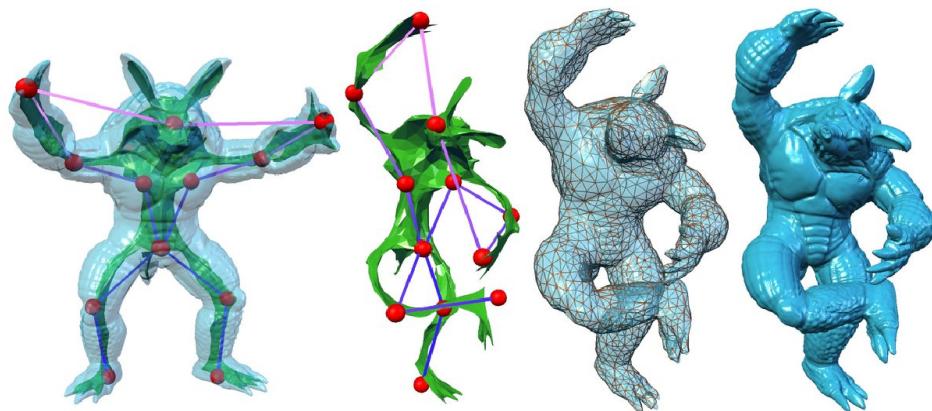


Figure 3.15 – *Shape pose editing through medial surface topology modeling (YBS07)*.

a graph structure is abstracted from the skeleton and graph matching techniques are then employed to measure similarity between surfaces.

Curve skeletons can also be employed in the framework of shape edition through metamorphosis (BTSG00) or shape pose edition for animation (Blo02, WML*06, YBS07, ATC*08).

Discussion

According to Cornea et al. (CSM07), curve-skeleton extraction techniques can be classified in four groups: thinning, distance-field, general field and geometry based extractions (see respectively (WLo08), (BS00), (CSYB05) and (ATC*08) for representative techniques).

Following this taxonomy, it turns out that some of these algorithm classes can guarantee some description features and some cannot. For example, Cornea et al. (CSM07) report that thinning and general field based techniques in general cannot reconstruct the surface from the representation, which is a major drawback. Indeed, an efficient shape modeling technique should always enable the reconstruction from the description of the input surface. Such a feature is needed for example for surface comparison, pose edition or metamorphosis.

Moreover, curve skeletons are inspired from medial axis. Like that of medial axis, the definition of medial surfaces (see equation 3.9) implies it is inherently sensitive to perturbation of the input surface (like surface noise). Consequently, the lack of stability of the extracted description towards surface perturbation is also a commonly admitted drawback of curve skeleton extraction techniques, resulting in the apparition of an important number of additional skeleton branches in the presence of noise.

In their taxonomy, Cornea et al. (CSMo07) state that only general field based methods provide satisfactory robustness towards slight surface perturbations.

In practice, most of existing techniques define a post-process pruning strategy aiming at suppressing irrelevant branches of the skeletons. However, pruning may often be too aggressive to guarantee that the final skeleton preserves the input surface topology, which is also a major drawback for topology characterization.

3.3.2 Surface mesh segmentation

Surface mesh segmentation consists in partitioning a surface mesh into segments of homogeneous characteristics. Then a structural representation can be abstracted from the adjacency graph of the extracted partition.

Shamir (Shao08) formulates the segmentation process as a partitioning optimization problem whose aim is to find a disjoint partition P of the mesh into P_0, P_1, \dots, P_k segments such that a criterion function $f_C(P)$ is minimized (or maximized) under a set of constraints C (Shao08).

Although this formulation gives a good intuition of the problem, in practice however, the segmentation problem is rarely formalized this way. First, to prevent combinatorial explosion, global optimization processes are avoided. Second, most of the time, the applicative constraints are hard to make explicit as a constraint functional (f_C).

Surface mesh segmentation techniques can be classified into two categories (AKM^{*}06, Shao08): *low-level* and *high-level* approaches (or *surface-type* and *part-type* (Shao08)).

Low-level approaches aim at decomposing the mesh into segments of homogeneous geometrical properties (see figure 3.16 for examples). These techniques mostly serve in a divide-and-conquer strategy to make the resolution of global geometry processing problem easier. For example, in the context of surface planar parameterization, the objective of the segmentation will be to extract surface segments with disk topology whose geometry enables low distortion unfolding (LPRM02). In the context of compression of computer-aided engineering designed models, Lavoué et al. (LDB05) propose to analyse the curvature tensor to decompose the mesh into segments of homogeneous curvature. Surface re-meshing and simplification are other contexts of development of such techniques (BM03, CSAD04). However, low-level segmentation approaches are most of the time application-specific solutions, due to the low-level geometrical

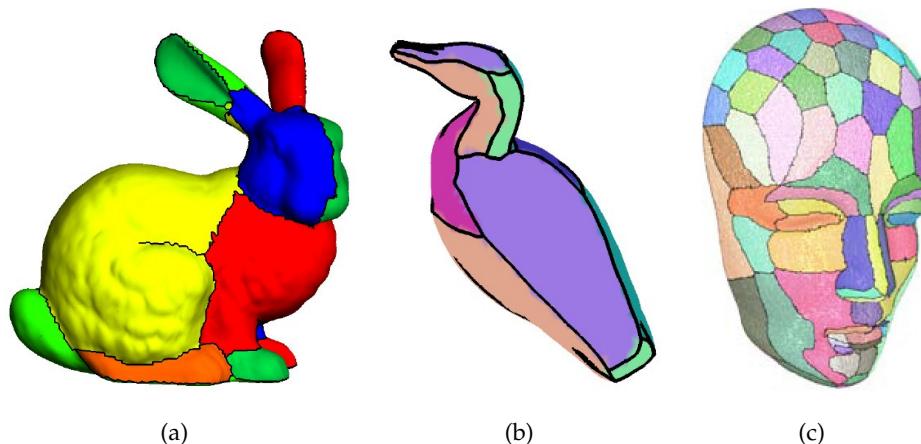


Figure 3.16 – Examples of low-level surface mesh segmentation (3.16(a) (LPRM02), 3.16(b) (BM03), 3.16(c) (CSAD04)).

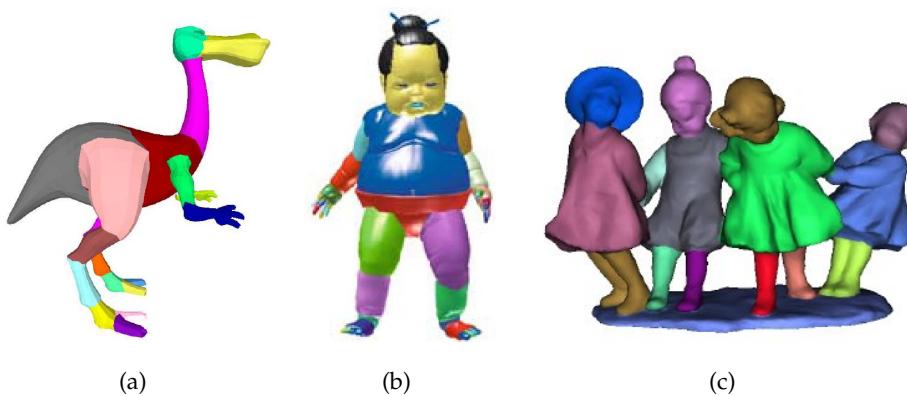


Figure 3.17 – Examples of high-level surface mesh segmentation (3.17(a) (KT03), 3.17(b) (KL05), 3.17(c) (LZ07)).

constraints they express, and thus are difficult to re-use in other applicative contexts.

High-level approaches aim at decomposing the mesh into segments of homogeneous perceptual interest (see figure 3.17 for examples). Using heuristics based on human perception theory (Bie87), these approaches have the ambition to infer a *meaningful* decomposition of the 3D shape without a priori knowledge. Such a high-level surface description benefits shape understanding or perception related applications in general, such as shape recognition or compression for example.

The main challenge of high-level segmentation techniques is to infer high-level notions from low-level measurements (such as curvature or geodesic distances). Then, clustering-based algorithm are mostly employed to derive the decomposition from the latter measurements.

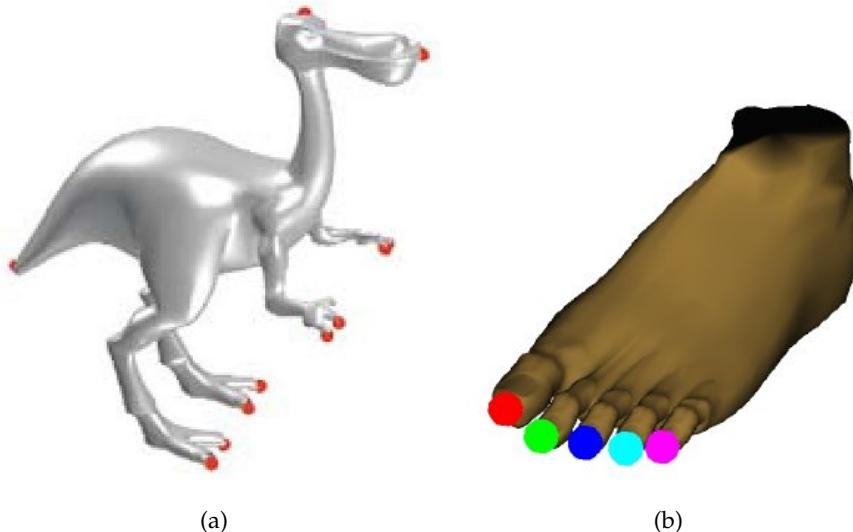


Figure 3.18 – Feature points extracted with the techniques presented in (KLT05) (3.18(a)) and (LZ07) (3.18(b)) respectively.

For example, Katz and Tal (KT03) introduces a face-to-face metric based on geodesic distances along the surface and normal angular distances. The geodesic component of this distance guarantees robustness against affine and near-isometric transformations. The angular distance component helps in placing segment boundaries along the most concave parts of the surface (Bie87). Then, this metric is computed for each pair of faces of the mesh and the decomposition is achieved using a variant of the canonical *k-means* clustering algorithm (DH73).

Podolak et al. (PSG*06) propose to drive the segmentation through the notion of perceived symmetry. The decomposition process is also based on a *k-means* algorithm which considers the planar-reflective transform values of each vertex as an input.

In a follow-up work, Katz et al. (KLT05) introduce the notion of *feature points* as reference points to help the decomposition process. The feature points are the points located at the extremity of prominent components of the surface and thus can be considered as points of particular geometrical and perceptual interest (see figure 3.18). Liu and Zhang (LZ07) also base their algorithm on automatic feature point extraction but propose a slightly different computation.

These few examples give a good preview of the kind of notions authors propose to infer from low-level measurements. We still defer the reader to (AKM*06, Shao08) for complete surveys of segmentation techniques.

Applications

Low-level segmentation approaches are mostly ad-hoc solutions specifically designed to serve a divide-and-conquer strategy, to make the resolution of a global geometry processing problem easier. Consequently, their transposition to other contexts may often be inappropriate, which limits their applicative impact. Compression, texture mapping, surface re-meshing or subdivision are typical applications using low-level segmentation (KGoo, LPRM02, BM03, CSAD04, LDB05).

On the other hand, *high-level* segmentation approaches try to decompose the surface according to criteria that are general enough to be used in several applications, mostly dealing with shape understanding. As these approaches segment the surface mesh into segments of homogeneous perceptual interest, the related adjacency graph (see figure 3.19) provides a structural description of the surface which captures a certain *visual semantic* (AKM^{*}06), motivating their use for shape understanding related problems.

Surface metamorphosis is an example of such an application, where the mapping between the input surfaces has to rely on perceptual hypothesis so the metamorphosis from a surface to the other is visually coherent. To this end, Shlafman et al. present a morphing framework based on high-level surface segmentation (STK02). Character animation (KTo3) and collision detection for character animation (LTZ01) are two other examples of applications where high-level segmentation has shown to be particularly interesting. Finally, these approaches have also been used as a bias to drive graph-matching based shape comparison by perceptual criteria in shape retrieval (ZTS02).

Discussion

As proposed by Katz and Tal (KTo3) in the framework of skeleton-driven shape pose edition, a structural representation of the surface can be abstracted from the segmentation by considering the adjacency graph of the extracted segments (see figure 3.19). Indeed, thanks to their heuristics based on visual perception, high-level segmentation techniques consequently provide *expressive* surface structural descriptions.

High-level segmentation based structural description has several interesting properties. First, on the contrary to certain curve-skeleton extraction techniques, the input surface can always be reconstructed from the structural adjacency graph, since the surface is represented by a set

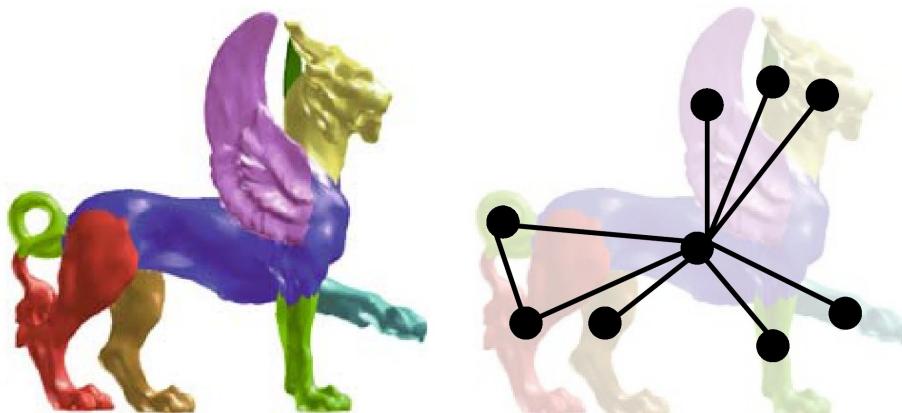


Figure 3.19 – *Adjacency graphs abstracted from the segmentation may not preserve exactly the topology of the surface (KLT05)* (two handles and only one loop in the graph).

of surface patches (segments). Since it is obtained from decomposition, this graph has an explicit connection to the surface segments. Second, most of high-level segmentation techniques base their analysis on geodesic distances, which guarantees the robustness of the representation towards standard transformations (affine and isometric).

Although these techniques may introduce interesting perceptual heuristics, it is still hard to evaluate their relevance since no ground-truth validation process has been proposed so far, like it has been done for 2D image segmentation. Consequently, there is no measurable guarantee that resulting segmentations express a semantic decomposition.

However, even with a faithful semantic decomposition obtained from a human operator, there is still no guarantee that the resulting adjacency graph exactly preserves the surface topology. Indeed, high-level segmentation techniques' decomposition criteria are perceptual and not topological. For example, figure 3.19 shows a genus-2 surface (with two topological handles) whose adjacency graph has only one loop. The topological handle in green is composed of only one segment and thus does not introduce an additional loop in the graph. Consequently, the presence of this topological handle cannot be inferred by reasoning on the adjacency graph. Thus, the adjacency graph of high-level segmentation does not provide an efficient computational representation of topological handles, which is a major topological invariant of a surface. Like for most of curve-skeleton extraction techniques, high-level segmentation based topology modeling may not preserve exactly the topology of the input surface, which prevents complete topology intrinsic representation and which can be detrimental to further geometry processing tasks.

3.3.3 Differential topology based modeling

Differential topology is a field of mathematics that studies the topological characteristics of manifolds through the use of differential geometry tools (Hir76). An example of celebrated result from differential topology is the hairy ball theorem, which states that, due to its topological characteristics, there is no non-vanishing continuous tangent vector field on the sphere (EG79).

In particular, Morse theory (Mor25, Mil63) is a branch of differential topology that investigates the relations between the topological invariants of manifolds and the properties of a specific class of smooth real functions defined on them, called the Morse functions. Morse theory provides a powerful framework for direct topology characterization of manifolds and has shown its utility on its own in mathematics for the demonstration of results on geodesics, in quantum field theory and more recently in shape modeling.

We here give intuitions regarding Morse theory related tools for shape modeling mainly to underline their applicative interest, but we defer the reader to the next chapter (page 61) for formal definitions and properties.

Morse theory in shape modeling

Two main tools derived from Morse theory have shown to be particularly useful in the context of shape modeling for Computer Graphics or Computer Vision, namely the Morse-Smale complex and the Reeb graph.

The Morse-Smale complex is a cellular decomposition of a Morse (Smale) function defined over a manifold, where cell boundaries are located on the steepest ascent/descent gradient lines of the function, which intersect at the points where the function's gradient vanishes (EHZ03). Thanks to their roots in Morse theory, Morse-Smale complexes efficiently capture manifolds' topology and have been used in several computer graphics problems, like terrain modeling (BEHP04), chart-atlas generation (NGHo4) or surface re-meshing (DBG*06).

The Reeb graph (Ree46) is another representation of Morse functions defined on manifolds, which contracts the connected components of Morse functions' level sets to points (PSBM07). In the case of surfaces, this compact representation, when embedded in \mathbb{R}^3 , forms a skeleton of the 3D shape, as shown in figure 3.20, often called *topological skeleton*. In particular, in the case of closed surfaces embedded in \mathbb{R}^3 , it has been shown that the number of loops in the Reeb graph exactly equals the number



Figure 3.20 – Surfaces including topological handles and the Reeb graphs corresponding to the colored functions (PSBM07).

of topological handles of the surface (CMEH^{*}03). Generally speaking, in number of situations, its roots in Morse theory enable to derive easily further relations between the Reeb graph and the manifold’s topology, as we will see later on this manuscript. Moreover, Reeb graphs show great flexibility in term of invariant surface representation, since their geometrical invariance characteristics can be made explicit through the definition of the under-laying Morse function.

Consequently, the Reeb graph is a particularly interesting candidate for 3D shape intrinsic structural modeling: its skeleton nature gives a good overview of the global surface structure, its geometrical invariance can be defined explicitly and it efficiently captures surface topological invariants.

Reeb graphs were first introduced in shape modeling for surface encoding by Shinagawa et al. (SKK91). Ever since, they have been developed in number of flavors and have shown their utility in a large number of computer graphics or computer vision problems, like surface compression (BMS00), surface parameterization (SF02, PSF04, ZMT05), surface quadrangulation (HA03b) iso-surface extraction (vKvOB^{*}97), re-meshing (WDS00) and simplification (CSdPo04, WHDS04), surface shading (BHK03), shape retrieval (HSKK01, TS05), character animation (AHLDo07), etc. We defer the reader to (BGSF08) for a recent survey on the applications of Reeb graphs in shape modeling.

When topology modeling rescues geometry modeling

“Any problem which is non-linear in character, which involves more than one coordinate system or more than one variable, or where structure is initially defined in the large, is likely to require considerations of topology and group theory for its solution. In the solution of such problems classical analysis will

frequently appear as an instrument in the small, integrated over the whole problem with the aid of topology or group theory.” (Mor34)

As suggested by this citation of Marston Morse, topology shape modeling can be of great help in purely geometric problems when geometry shape modeling meets its limitations.

For a certain class of purely geometric problems, even when no particular care needs to be given to the surface topology, differential topology has shown to be useful. This is the case for example in surface planar parameterization (PSFo4, ZMT05), chart-atlas generation (NGHo4) or surface re-meshing (DBG*06).

However, as we have seen in this chapter, geometry modeling techniques meet limitations in several problems when arbitrary topology surfaces are considered and specifically when topology variation needs to be efficiently taken into account. For example, in the context of conformal geometry modeling based surface comparison, input surfaces are required to share the same topology. The same remark goes to shape modeling in shape space, where surfaces represented in a given shape space also have to share the same topology.

Thanks to their roots in Morse theory, Reeb graphs provide a faithful solution to surface topology understanding and control. For example, as the number of loops in a Reeb graph equals the number of topological handles of the related surface, geometrical detection of topological handles can be reduced to a combinatorial problem, which consists in the identification of loops in a graph. This example shows that number of geometry related problems can be reduced thanks to the Reeb graph to graph problems which have been already massively studied in Computer Science.

Extending this idea, beyond topology understanding, Morse theory tools such as Reeb graphs permit topology control and edition. Recently, several authors (BPJ07, LGQ08) have investigated differential topology for genus-oblivious surface cross-parameterization, that is the computation of a point-to-point mapping between surfaces of possibly different topology. For example, the first proposed solution was based on Reeb graphs (BPJ07). Such topology oblivious cross parameterization techniques seem to be promising directions of research for the automatic mapping computation between surfaces of possibly different topology. Such a result would be of great interest to overcome conformal or Riemannian geometry modeling’s main limitations.

As a consequence, differential topology modeling and more specifically Reeb graphs are tools of great potential for intrinsic surface structural representation, complete topology characterization and topology control for geometry processing. For these latter reasons, we will focus in this thesis on the development of Reeb graph based modeling techniques.

3.4 CHAPTER CONCLUSION

In this chapter, we reviewed state of the art techniques for the intrinsic modeling of 3D shapes computationally represented as surface meshes. We took the decision to split this survey into two main categories, respectively geometry and topology based modeling, to underline the complementarity of these two concepts.

First, we reviewed three representative classes of geometry modeling techniques: spectral modeling, conformal geometry modeling and Riemannian geometry modeling. We showed the main advantages of these techniques and showed that most of their limitations are due to topology considerations.

Second, we reviewed three representative classes of topology modeling techniques for surface structural description: curve-skeleton extraction, surface mesh segmentation and differential topology based modeling. In particular, we showed curve-skeleton and segmentation related techniques provided interesting structural descriptions from an applicative point of view but sometimes failed in completely characterizing the topology of the input surface.

On the contrary, Reeb graph based modeling, thanks to its solid theoretical foundations rooting in Morse theory, provides both an expressive structural description of the surface based on a skeletal representation, a complete characterization of the surface's topology and further topology control facilities able to overcome geometry modeling techniques' topology related limitations. For these reasons, due to Reeb graph's potential, we will mainly focus in this thesis in the development of Reeb graph based modeling techniques.

In the following chapter, we introduce the theoretical preliminaries. Then, we present the contributions of this thesis and show their utility in three research problems related to Computer Graphics and Computer Vision.

3.5 FRENCH CHAPTER ABSTRACT

Dans ce chapitre, nous avons dressé l'état de l'art de la modélisation intrinsèque de formes 3D représentées en mémoire sous la forme de surfaces maillées. Nous avons divisé cette étude en deux catégories principales, respectivement la modélisation géométrique et topologique, pour mettre en évidence la complémentarité de ces deux concepts.

Premièrement, nous avons passé en revue trois classes représentatives de techniques de modélisation géométrique: les modélisations spectrale, conforme et Riemannienne. Nous avons détaillé les avantages respectifs de ces types d'approche et nous avons montré que la plupart de leurs limitations concernent des considérations de topologie.

Deuxièmement, nous avons passé en revue trois classes représentatives de techniques de modélisation topologique: l'extraction de squelettes-courbes, la segmentation de surfaces maillées et la topologie différentielle. En particulier, nous avons montré que les techniques liées à l'extraction de squelettes-courbes et la segmentation proposaient des descriptions structurelles intéressantes d'un point de vue applicatif mais échouaient parfois à capturer fidèlement tous les invariants topologiques de la surface en entrée.

Au contraire, la modélisation par graphe de Reeb, grâce à ses fondements théoriques éprouvés, ayant pour origine la théorie de Morse, fournit à la fois une description structurelle invariante et expressive, une caractérisation complète de la topologie de la surface ainsi que des facilités de contrôle de topologie permettant de surmonter les limitations intrinsèques de la modélisation géométrique liées à la topologie. Pour ces raisons, nous nous intéresserons principalement dans cette thèse au développement de techniques de modélisation basées sur les graphes de Reeb.

Dans le chapitre suivant, nous introduisons les préliminaires théoriques. Ensuite, nous présentons les contributions de cette thèse et montrons leur utilité dans trois problèmes de recherche en rapport avec l'Informatique Graphique et la Vision par Ordinateur.

NOTATIONS

SMOOTH SETTING

\mathbb{X}	Topological space
\mathbb{R}^d	Euclidean space of dimension d
\mathbb{H}^d	Half space of \mathbb{R}^d
\mathbb{M}	d -manifold
$\partial\mathbb{M}$	Boundary of \mathbb{M}
g, b	Genus and number of boundary components of \mathbb{M}
$\chi(\mathbb{M})$	Euler characteristic of \mathbb{M}
$f, f^{-1}(b)$	Function $f : \mathbb{A} \rightarrow \mathbb{B}$ and level set of $b \in \mathbb{B}$ by f
κ_H, κ_G	Mean and Gaussian curvatures
$R(f)$	Reeb graph of f

DISCRETE SETTING

\mathcal{K}	Simplicial complex
σ, τ	d -simplex, face of a d -simplex
v, e, t	Vertex (0-simplex), edge (1-simplex) and triangle (2-simplex)
\mathcal{S}	Triangulated surface
$\partial\mathcal{S}$	Boundary of \mathcal{S}
g, b	Genus and number of boundary components of \mathcal{S}
$\chi(\mathcal{S})$	Euler characteristic of \mathcal{S}
$Lk(v), St(v)$	Link and star of v
f	Piecewise linear function $f : \mathcal{K} \rightarrow \mathbb{R}$
$Lk^-(v), Lk^+(v)$	Lower and upper link of v relatively to f
$Lk_d(v), Lk_d^-(v)$	d -simplices of $Lk(v)$ and $Lk^-(v)$
κ_H, κ_G	Discrete mean and Gaussian curvatures
$\mathcal{R}(f)$	Reeb graph of f

4

THEORETICAL BACKGROUND

CONTENTS

4.1	SMOOTH SETTING	63
4.1.1	Differentiable manifolds	63
4.1.2	Elements of Morse theory	71
4.1.3	Reeb graphs	75
4.2	DISCRETE SETTING	79
4.2.1	Manifold simplicial decompositions	79
4.2.2	Morse theory in the discrete setting	84
4.2.3	Reeb graphs in the discrete setting	87
4.3	CHAPTER CONCLUSION	88
4.4	FRENCH CHAPTER ABSTRACT	89

THIS chapter concentrates the theoretical preliminaries of the presented contributions. It is subdivided into two parts, to reflect the way surface related problems are apprehended in Computer Science. First, smooth concepts are presented. Then, their approximations in the discrete setting, more tangible for computers, are introduced and techniques for their computation are presented.

We invite the impatient readers having sufficient background to directly jump to the important definitions (in boxes).

We defer the reader to the books by Milnor (Mil63), Gramain (Gra71), do Carmo (dC92), Fomenko and Kunii (FK97), Edelsbrunner (Ede01) and Lee (Lee03) for further readings.

4.1 SMOOTH SETTING

First, we introduce through a set of definitions the notion of *manifolds* which generalizes that of surfaces and some of its properties. To describe these concepts, a number of intermediary necessary definitions are presented along the discussion. We invite the impatient readers having sufficient background to directly jump to the important definitions (in boxes).

4.1.1 Differentiable manifolds

Definitions

Definition 1 (Topology) A topology on a set \mathbb{X} is a collection \mathcal{T} of subsets of \mathbb{X} having the following properties:

- The sets \emptyset and \mathbb{X} are in \mathcal{T} ;
- The union of any sub-collection of \mathcal{T} is in \mathcal{T} ;
- The intersection of a finite sub-collection of \mathcal{T} is in \mathcal{T} .

Definition 2 (Topological space) A set \mathbb{X} for which a topology \mathcal{T} is defined is called a topological space.

For example, the space of real numbers \mathbb{R} is a topological space.

Definition 3 (Open set) A subset $\mathbb{A} \subset \mathbb{X}$ of the topological space \mathbb{X} is an open set of \mathbb{X} if it belongs to \mathcal{T} .

Definition 4 (Closed set) A subset $\mathbb{B} \subset \mathbb{X}$ of the topological space \mathbb{X} is a closed set of \mathbb{X} if its complement $\mathbb{X} - \mathbb{B}$ is open.

Intuitively, open sets are subsets of topological spaces which do not contain their boundaries. For example, considering the space of real numbers \mathbb{R} , $(-\infty, 0) \cup (1, +\infty)$ and $[0, 1]$ are complements and respectively open and closed sets.

Property 1 (Open sets)

- The set \emptyset is open;
- The union of any number of open sets is open;
- The intersection of a finite number of open sets is open.

These properties follow from the definition of topology.

Definition 5 (Function) A function $f : \mathbb{A} \rightarrow \mathbb{B}$ associates each element of the topological space \mathbb{A} with a unique element of the topological space \mathbb{B} . A function is also called map.

Definition 6 (Continuous function) A function $f : \mathbb{A} \rightarrow \mathbb{B}$ is continuous if for each open subset $\mathbb{C} \in \mathbb{B}$, the set $f^{-1}(\mathbb{C})$ is an open subset of \mathbb{A} .

Definition 7 (Bijection) A function $f : \mathbb{A} \rightarrow \mathbb{B}$ is a bijection if for each element $b \in \mathbb{B}$ there is exactly one element $a \in \mathbb{A}$ such that $f(a) = b$. f is said to be bijective. It is also said to be one-to-one (injective) and onto (surjective).

Definition 8 (Homeomorphic spaces) Two topological spaces \mathbb{A} and \mathbb{B} are homeomorphic if and only if there exists a continuous bijection $f : \mathbb{A} \rightarrow \mathbb{B}$ with a continuous inverse $f^{-1} : \mathbb{B} \rightarrow \mathbb{A}$. \mathbb{A} and \mathbb{B} are said to be homeomorphic.

Definition 9

(Manifold) A topological space \mathbb{M} is a d -manifold if every element $m \in \mathbb{M}$ has an open neighborhood \mathbb{N} homeomorphic to \mathbb{R}^d .

An intuitive description of a d -manifold is that of a curved space, which has locally the structure of an Euclidean space of dimension d , but which has a more complicated global structure. Manifolds are often seen as a gluing of Euclidean spaces. Moreover, d -manifolds can also be seen as a continuous piling of $(d - 1)$ -manifolds. Notice that consequently to this definition, Euclidean spaces fall as special cases of manifolds.

This straightforward definition then follows:

Definition 10 (Surface) A surface is a 2-manifold.

As we are concerned in this thesis with the topology modeling of surfaces, if omitted, the manifolds' dimension (noted d) is meant to be 2.

There are many different notions of manifold, with more or less structure, yielding in a manifold categorization. In the context of Morse theory, we will particularly focus on smooth compact manifolds.

Definition 11 (Covering) A collection \mathcal{C} of subsets of a topological space \mathbb{X} is a covering of \mathbb{X} if the union of elements of \mathcal{C} is equal to \mathbb{X} .

Definition 12 (Compact manifold) A d -manifold \mathbb{M} is compact if every open covering \mathcal{C} of \mathbb{M} contains a finite sub-collection that is also a covering of \mathbb{M} .

The following property and definitions will be particularly useful for the setup of Morse theory:

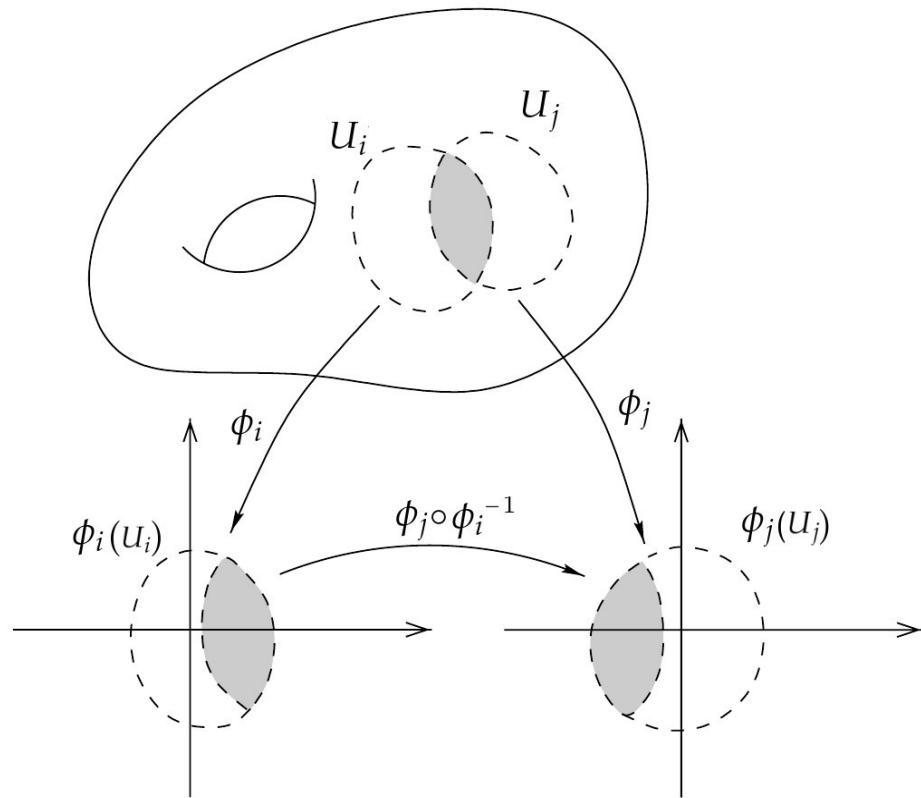


Figure 4.1 – Example of transition function $\phi_{ij} = \phi_j \circ \phi_i^{-1}$ for a 2-manifold (Lee03).

Property 2 (Compact manifolds) Any continuous real-valued function is bounded on a compact d -manifold.

Definition 13 (Smooth function) A function $f : \mathbb{A} \rightarrow \mathbb{B}$ is smooth if its partial derivatives of all orders are continuous functions.

Definition 14 (Smooth manifold, figure 4.1) Let \mathbb{M} be a d -manifold and $\{(U_i, \phi_i)\}$ an open covering of \mathbb{M} ($\mathbb{M} = \cup U_i$) along with homeomorphisms $\phi_i : U_i \rightarrow \mathbb{R}^d$. Let $\forall i, j, \phi_{ij} : \phi_i(U_i \cap U_j) \rightarrow \phi_j(U_i \cap U_j)$ be the function defined by the formula $\phi_{ij} = \phi_j \circ \phi_i^{-1}$. If every function ϕ_{ij} (called transition function) is a smooth function, then \mathbb{M} is a smooth d -manifold. A smooth d -manifold is also said to be differentiable.

3D shape modeling problem description

To give a formal description of the problem addressed by 3D shape modeling research, we then introduce the notion of *embedding*.

Definition 15 (Injection) A function $f : \mathbb{A} \rightarrow \mathbb{B}$ is an injection if for each couple of elements

$a_1, a_2 \in \mathbb{A}$ such that $a_1 \neq a_2$, $f(a_1) \neq f(a_2)$. f is said to be injective. It is also said to be one-to-one.

Definition 16 (Embedding) A function $f : \mathbb{A} \rightarrow \mathbb{B}$ is an embedding of \mathbb{A} in \mathbb{B} if f is injective and continuous and if \mathbb{A} and $f(\mathbb{A})$ are homeomorphic.

To give an intuitive example, the application of a sticking plaster on somebody's wound can be seen as an embedding of the surface of the plaster on the surface of the body. Even if the application stretches the plaster, it is still homeomorphic to any unused sticking plaster of same type. Since the sticking plaster does not recover the whole body, the application is injective.

An embedding of \mathbb{A} in \mathbb{B} is an *extrinsic* representation of \mathbb{A} since it depends on the embedding function f . Consequently the embedding of a manifold is a biased representation of that manifold.

Back to our problem, surfaces of the real world are 2-manifolds embedded in the *real space*, the Euclidean 3-dimensional space. Consequently, in 3D shape modeling, all considered surfaces, either in acquisition or computed-aided design contexts, are embedded in \mathbb{R}^3 through an unknown embedding function $f : \mathbb{S} \rightarrow \mathbb{R}^3$ and the only tangible input surface representation is the result of this embedding.

For example, some usual shape transformations, such as affine transformations (like translations or rotations in \mathbb{R}^3), preserve the surface geometrical and topological characteristics while altering its embedding representation in \mathbb{R}^3 . The same remark goes to isometric transformations (transformations that preserve the length of curves defined along the surface), which is the concept corresponding to the articulation of deformable 3D shapes.

Consequently, to consistently describe surfaces despite these usual embedding transformations (which correspond to usual real world surface non-altering transformations), 3D shape modeling research aims at computing intrinsic surface representations, given one of their embedding in \mathbb{R}^3 .

Topological invariants

In the following, we introduce several topology notions (boundary components, genus, Euler characteristic, etc.) in order to further characterize manifolds.

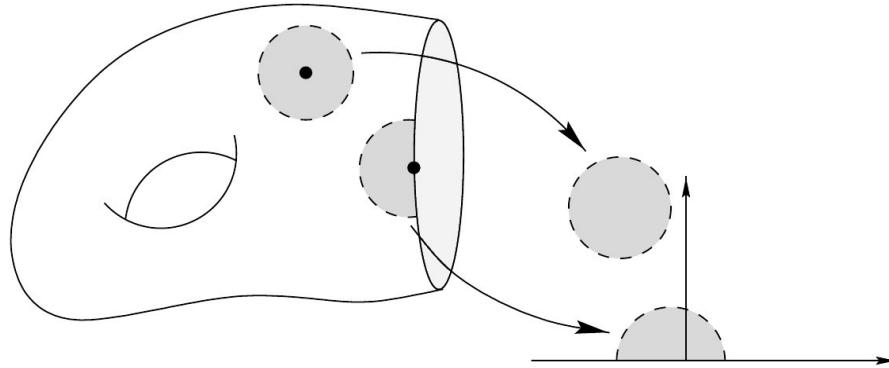


Figure 4.2 – A 2-manifold with one boundary component (Lee03).

Definition 17 (Half space) *The space $\mathbb{H}^d \subset \mathbb{R}^d = \{h = (h^1, \dots, h^d) : h^d \geq 0\}$ is called the half space of \mathbb{R}^d .*

Definition 18 (Manifold with boundary) *A topological space \mathbb{M} is a d-manifold with boundary if every element $m \in \mathbb{M}$ has an open neighborhood \mathbb{N} homeomorphic either to \mathbb{R}^d or to the half space \mathbb{H}^d .*

Definition 19 (Manifold's boundary) *The boundary of a d-manifold \mathbb{M} with boundary, noted $\partial\mathbb{M}$, is the complement in \mathbb{M} of the subspace of \mathbb{M} , called the interior of \mathbb{M} , composed of all the elements $m \in \mathbb{M}$ such that m has an open neighborhood \mathbb{N} homeomorphic to \mathbb{R}^d .*

Definition 20 (Closed manifold) *A d-manifold \mathbb{M} without boundary is called a closed manifold.*

Property 3 (Manifolds' boundary) *The boundary $\partial\mathbb{M}$ of a d-manifold \mathbb{M} with boundary is a closed $(d - 1)$ -manifold.*

Definition 21 (Connected manifold) *A d-manifold \mathbb{M} is disconnected if it is the union of disjoint non-empty open sets. Otherwise, \mathbb{M} is said to be connected.*

Definition 22 (Connected components) *The maximal connected subsets of a d-manifold \mathbb{M} are called its connected components.*

Definition 23 (Boundary components, figure 4.2) *The connected components of the boundary $\partial\mathbb{M}$ of a d-manifold \mathbb{M} with boundary are called the boundary components of \mathbb{M} .*

The following concepts are defined in the special case of 2-manifolds. They can also be extended to higher dimensions, though their introduction requires more work.

Definition 24 (Genus of a surface) *The genus g of a 2-manifold \mathbb{M} is the maximal number of non-intersecting connected closed 1-manifolds (curves) embedded on \mathbb{M} that cut \mathbb{M} (creating artificial boundary components) while still preserving the number of connected components of \mathbb{M} .*

For example, a sphere sectioned at its equator is cut into two disconnected half-spheres. On the contrary, cutting a torus on the inside of its topological handle opens the torus but does not separate it into several connected pieces. Consequently, the genus of a torus equals one while that of a sphere equals zero. Intuitively, the genus of a 2-manifold corresponds to the number of its topological handles. The genus of the 2-manifold represented in figure 4.2 equals one.

Definition 25 (Normal vector field) *The normal vector field of a 2-manifold \mathbb{M} is a function $N : \mathbb{M} \rightarrow \mathbb{R}^3$ that maps each element $m \in \mathbb{M}$ to a vector orthogonal to the tangent plane of \mathbb{M} in m . $N(m)$ is called the normal vector of \mathbb{M} in m .*

Definition 26 (Orientable surface) *A 2-manifold is said to be an orientable surface if its normal vector field is non null for every element $m \in \mathbb{M}$ and if it is a continuous function.*

Intuitively, an orientable 2-manifold is a surface for which it is possible to distinguish an inner and an outer side. Examples of non-orientable surfaces are the Möbius strip and the Klein bottle.

Property 4 (Compact orientable surfaces) *Any compact 2-manifold that is either orientable or has a boundary can be embedded in \mathbb{R}^3 .*

Either in acquisition or computer-aided design contexts, surfaces represent the boundary of real world objects and are consequently orientable surfaces that can be embedded in \mathbb{R}^3 . Thus, all the surfaces considered later on this manuscript are orientable.

It follows that the topology of a compact, smooth, connected and orientable 2-manifold can be fully characterized by two pieces of information: its genus g and its number b of boundary components. This statement is captured into the celebrated Euler relation:

Definition 27

(Euler relation I) *The Euler characteristic $\chi(\mathbb{M})$ of a compact, smooth, connected and orientable 2-manifold \mathbb{M} is a topological invariant given by the following expression, where g stands for the genus of \mathbb{M} and b for its number of boundary components:*

$$\chi(\mathbb{M}) = 2 - 2g - b \quad (4.1)$$

Some geometrical invariants

In the following, we introduce some geometrical invariants massively used in shape modeling. For the sake of conciseness, we will only define them in the case of a compact, smooth, connected and orientable 2-manifold¹.

In particular, we will consider local homeomorphisms of an open covering $\{(U_i, \phi_i)\}$ of the 2-manifold \mathbb{M} to subsets of \mathbb{R}^2 $\phi_i : U_i \rightarrow \mathbb{R}^2$, already introduced in definition 14 and the local coordinate system in \mathbb{R}^2 , denoted (u, v) , forming local parameterizations of \mathbb{M} .

Given an embedding of the 2-manifold \mathbb{M} in \mathbb{R}^3 , let $\mathbf{S}_i(u, v)$ be the embedding function that maps a point p of $\phi_i(U_i)$ with coordinate $(u, v) \in \mathbb{R}^2$ to a point of the embedded surface in the Euclidean 3D space \mathbb{R}^3 with coordinate (x, y, z) .

The partial derivatives of \mathbf{S}_i are then given by:

$$\frac{\partial \mathbf{S}_i}{\partial u}(u, v) = [\frac{\partial x}{\partial u}(u, v), \frac{\partial y}{\partial u}(u, v), \frac{\partial z}{\partial u}(u, v)] \quad (4.2)$$

$$\frac{\partial \mathbf{S}_i}{\partial v}(u, v) = [\frac{\partial x}{\partial v}(u, v), \frac{\partial y}{\partial v}(u, v), \frac{\partial z}{\partial v}(u, v)] \quad (4.3)$$

Definition 28 (First fundamental form) *The first fundamental form of a 2-manifold \mathbb{M} , often noted ds^2 , is given by the following expression:*

$$ds^2 = dw^T G dw \quad (4.4)$$

where G is the first fundamental form matrix:

$$G = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{S}_i}{\partial u} \cdot \frac{\partial \mathbf{S}_i}{\partial u} & \frac{\partial \mathbf{S}_i}{\partial u} \cdot \frac{\partial \mathbf{S}_i}{\partial v} \\ \frac{\partial \mathbf{S}_i}{\partial v} \cdot \frac{\partial \mathbf{S}_i}{\partial u} & \frac{\partial \mathbf{S}_i}{\partial v} \cdot \frac{\partial \mathbf{S}_i}{\partial v} \end{bmatrix} \quad (4.5)$$

and $dw = (du, dv)^T$.

The first fundamental form depicts intrinsic geometrical characteristics of the 2-manifold \mathbb{M} . It enables to infer some metric properties of \mathbb{M} from

¹Moreover, also in a conciseness purpose, we introduce these notions by reasoning on the manifold's embedding: a more comprehensive study requires the introduction of the concepts of Riemannian manifold and of metrics on tangent spaces.

its embedding in \mathbb{R}^3 . In other words, it enables intrinsic measurements on a surface, like the computation of the length of curves embedded on it.

In particular, let $\gamma(t) = (x^1(t), x^2(t))$ be a smooth parametric curve embedded on a 2-manifold \mathbb{M} .

Definition 29 (Christoffel symbols) *The Christoffel symbols are the functions:*

$$\Gamma_{jk}^i = \sum_{l=1}^2 \frac{1}{2} g^{il} \left(\frac{\partial g_{kl}}{\partial x^j} + \frac{\partial g_{il}}{\partial x^k} - \frac{\partial g_{jk}}{\partial x^l} \right) \quad (4.6)$$

where g_{ij} are the coefficients of the first fundamental form matrix G and g^{ij} are those of its inverse G^{-1} .

Definition 30 (Geodesic on a surface) *A smooth parametric curve $\gamma(t) = (x^1(t), x^2(t))$ embedded on a connected 2-manifold \mathbb{M} is a geodesic if it is the solution of the following system:*

$$\frac{d^2 x^i}{dt^2} + \sum_{l,k=1}^2 \Gamma_{lk}^i \frac{dx^l}{dt} \frac{dx^k}{dt} = 0, \quad i = 1, 2 \quad (4.7)$$

Nota Bene

In the rest of this manuscript, as we will mainly deal with geodesics in the discrete setting (on triangulated surfaces), we will not exploit this analytic definition of geodesics. We invite the reader to simply interpret a geodesic on a 2-manifold \mathbb{M} as the *shortest path* (minimal length) along \mathbb{M} between two points of it.

Geodesic measurements are intrinsic characteristics. In particular, the length of geodesics is invariant against affine and isometric transformations (the latter corresponding to the notion of articulation of deformable 3D shapes). Consequently, the measure of geodesics along surfaces is a common tool in 3D shape modeling. Moreover, their computation in the discrete setting is quite straightforward.

Definition 31 (Second fundamental form) *The second fundamental form of a 2-manifold \mathbb{M} , often noted l , is given by the following expression:*

$$l = dw^T Q dw \quad (4.8)$$

where Q is the second fundamental form matrix:

$$Q = \begin{bmatrix} n \cdot \frac{\partial^2 S_i}{\partial u^2} & n \cdot \frac{\partial^2 S_i}{\partial u \partial v} \\ n \cdot \frac{\partial^2 S_i}{\partial v \partial u} & n \cdot \frac{\partial^2 S_i}{\partial v^2} \end{bmatrix} \quad (4.9)$$

and n is the unit normal vector field of the 2-manifold \mathbb{M} (\times stands for the cross product):

$$n = \frac{\frac{\partial S_i}{\partial u} \times \frac{\partial S_i}{\partial v}}{\left\| \frac{\partial S_i}{\partial u} \times \frac{\partial S_i}{\partial v} \right\|} \quad (4.10)$$

Definition 32 (Normal curvature) Let t be a tangent vector to a 2-manifold \mathbb{M} in one of its elements $m \in \mathbb{M}$ and n its unit normal vector field ($t \cdot n(m) = 0$). The normal curvature of a 2-manifold \mathbb{M} in m in the direction t is given by the following expression:

$$\kappa_n(t) = \frac{G^{-1} Q t \cdot t}{\|t\|^2} \quad (4.11)$$

Definition 33 (Principal curvatures) The principal directions e_1 and e_2 of a 2-manifold in one of its elements $m \in \mathbb{M}$ are the two tangent vectors to \mathbb{M} that globally and respectively maximizes and minimizes the normal curvature $\kappa_n(t)$. Their respective values are noted κ_1 and κ_2 and are called the principal curvatures of \mathbb{M} in m .

Definition 34 (Mean curvature) The mean curvature of a 2-manifold \mathbb{M} in one of its elements $m \in \mathbb{M}$, noted κ_H , is given by the following expression:

$$\kappa_H = \frac{\kappa_1 + \kappa_2}{2} \quad (4.12)$$

Definition 35 (Gaussian curvature) The Gaussian curvature of a 2-manifold \mathbb{M} in one of its elements $m \in \mathbb{M}$, noted κ_G , is given by the following expression:

$$\kappa_G = \kappa_1 \cdot \kappa_2 \quad (4.13)$$

Intuitively, curvature measurements characterizes the amount by which a local portion of a surface deviates from being flat.

The mean curvature is an extrinsic measure of the surface curvature. It is dependent on the embedding of the surface in the Euclidean 3D space.

On the contrary, the Gaussian curvature is an intrinsic measure of the surface curvature. It is not dependent on the embedding of the surface in the Euclidean 3D space. Thus it is also an interesting geometrical surface invariant for shape characterization.

4.1.2 Elements of Morse theory

In differential topology, Morse theory (Mil63, FK97, Ede01) is a powerful framework for direct manifold topology characterization, introduced by Marston Morse in the beginning of the twentieth century (Mor25, Mor34).

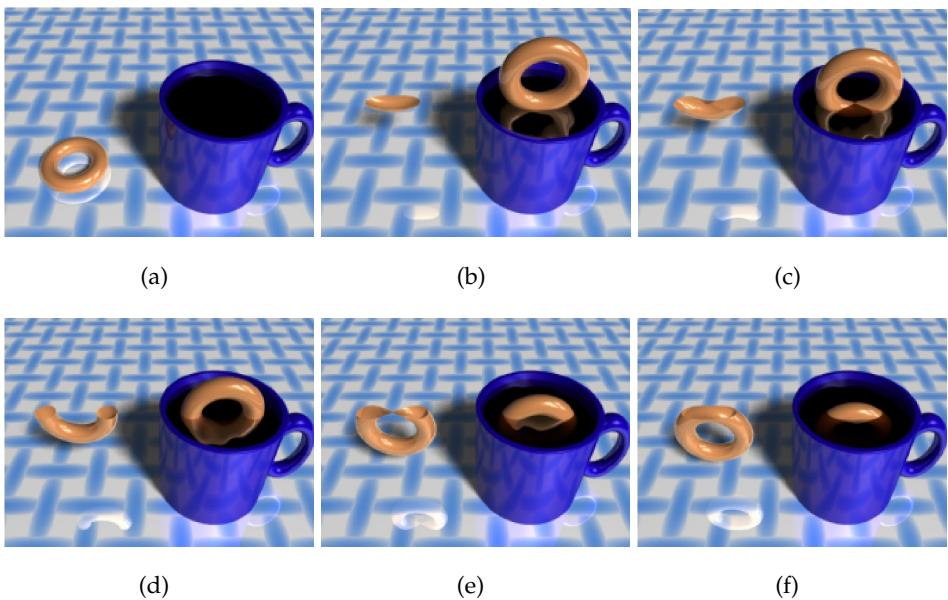


Figure 4.3 – *Coffee-dunked doughnut analogy to Morse theory* (Har98).

In particular, Morse theory enables to infer manifolds' topological invariants (as those introduced above) from the study of a certain class of smooth functions defined on them, called the *Morse functions*. We give here a few elements leading to this result.

Intuitive overview

Morse theory can be observed in everyday life, as suggested by John Hart in his *coffee-dunked doughnut* analogy to Morse theory (Har98), illustrated in figure 4.3.

Considering a doughnut progressively immersed in coffee, one can have a close look at the evolution of the curve resulting from the intersection of the coffee surface and that of the doughnut. This intersection has been emphasized by showing the immersed portion of the doughnut surface on the left of the pictures (figure 4.3).

Along the immersion, one can notice that the *topology* of the intersection curve evolves at very precise configurations of the doughnut. In particular, making the doughnut just slightly touch the coffee surface results in an intersection curve which falls to a point. Then, it progressively shifts to a connected closed curve, as shown in figure 4.3(b). At a very precise configuration of the doughnut surface (figure 4.3(c)), called a *saddle point*, the intersection curve's topology varies: it bifurcates into two disjoint components (figure 4.3(d)). These two *connected components* individually evolve

until they re-connect in another symmetric *saddle point* (figure 4.3(e)). Finally, the intersection curve evolves as a single connected component and fall to a point when the doughnut is nearly completely immersed (figure 4.3(f)).

An interesting observation of this particular experiment is that the intersection curve disconnected as many times as it re-connected: once. This observation is a topological invariant of the doughnut surface, previously introduced as *surface genus*. Indeed, the doughnut surface's genus equals one.

This experiment showed that by focusing on the topology of some function *level lines* and specifically on the configurations where this topology varies, one can infer some topological invariant of the whole surface. In the following, we formalize this result.

Definitions and results

Definition 36 (Critical point) *Let f be a smooth real valued function defined on a smooth compact d -manifold \mathbb{M} $f : \mathbb{M} \rightarrow \mathbb{R}$. An element $m \in \mathbb{M}$ is a critical point of f if f partial derivatives of all orders equal zero in m . $f(m)$ is called a critical value of f . If m is not a critical point, it is a regular point of f and $f(m)$ is called a regular value.*

Definition 37 (Hessian matrix) *The Hessian matrix of a smooth real valued function f defined on a smooth compact d -manifold in an element $m \in \mathbb{M}$ is the matrix of the partial derivatives of f of second order $d^2 f(m)$.*

Definition 38 (Non-degenerate critical point) *A critical point m of a smooth real valued function f defined on a smooth compact d -manifold is said to be non-degenerate if the Hessian matrix of f is non-singular at m .*

Definition 39

(Morse function) *A smooth real valued function f defined on a smooth compact d -manifold \mathbb{M} is called a Morse function if all its critical points are non-degenerate.*

Definition 40 (Isolated critical point) *A critical point m of a smooth real valued function f defined on a smooth compact d -manifold \mathbb{M} is called an isolated critical point if there exists an open neighborhood of m on \mathbb{M} such that m is the unique critical point.*

Property 5 (Morse function) *A Morse function defined on a smooth compact d -manifold \mathbb{M} counts finitely many critical points and all of them are isolated.*

This property is particularly important for topological invariant inference.

Definition 41 (Index of a non-degenerate critical point) *The index of a non-degenerate critical point of a smooth real valued function f defined on a smooth compact d -manifold is the number of negative eigenvalues of the Hessian matrix of f at the critical point.*

Considering the simple example of 2-manifolds embedded in \mathbb{R}^3 with the following embedding functions defined respectively to some local coordinate system (u, v) :

$$\mathbf{S}_1(u, v) = [u, v, u^2 + v^2] \quad (4.14)$$

$$\mathbf{S}_2(u, v) = [u, v, u^2 - v^2] \quad (4.15)$$

$$\mathbf{S}_3(u, v) = [u, v, -u^2 - v^2] \quad (4.16)$$

Considering now the functions f_1, f_2 and f_3 that map a point of these respective surfaces to its z -coordinate. Each of these functions admit a critical point at coordinate $(u, v) = (0, 0)$. The index of these critical points is here given by the number of minus signs in f_1, f_2 and f_3 expressions. In particular, $f_1(\mathbf{S}_1(0, 0))$ is a local minimum of f_1 : its index equals 0. $f_2(\mathbf{S}_2(0, 0))$ is called a *saddle point*: its index equals 1. $f_3(\mathbf{S}_3(0, 0))$ is a local maximum of f_3 : its index equals 2.

Definition 42 (Morse number) *The i^{th} Morse number, noted $\mu_i(f), 0 \leq i \leq d$, of a Morse function f defined on a smooth compact d -manifold \mathbb{M} equals the number of critical points of f of index i .*

Definition 43

(Morse-Euler relation) *The alternated sum of Morse numbers of a Morse function f defined on a smooth compact connected d -manifold \mathbb{M} equals the Euler characteristic $\chi(\mathbb{M})$ of the d -manifold \mathbb{M} :*

$$\chi(\mathbb{M}) = \sum_{i=0}^d (-1)^i \mu_i(f) \quad (4.17)$$

This relation is a major result from Morse theory. By itself, it captures the fact that the critical point setting of a Morse function fully describes the topology of a smooth, compact, connected and orientable manifold.

In the coffee-dunked doughnut analogy (figure 4.3), the considered Morse function f was the *height* function. It admitted a global minimum when the doughnut first touched the surface of the coffee ($\mu_0 = 1$), two



Figure 4.4 – Georges Reeb (right) in Oberwolfach, 1953.

saddle points when the intersection curve disconnected and re-connected ($\mu_1 = 2$) and one global maximum once the doughnut was nearly completely immersed ($\mu_2 = 1$). Consequently, in this example, we have

$$\chi(\mathbb{M}) = 1 - 2 + 1 = 0 = 2 - 2g - b \quad (4.18)$$

where $b = 0$ is the number of boundary component of the doughnut surface and $g = 1$ its genus.

Theorem 1 (Morse theorem) *Any smooth real valued function f defined on a smooth compact d -manifold \mathbb{M} can be transformed into a Morse function by a slight perturbation.*

This property results from the fact that Morse functions are everywhere dense in the space of smooth functions. This result has a very attractive applicative impact: in practice, every smooth function defined on a surface can benefit Morse function properties by a slight transformation which transforms degenerate critical points into non-degenerate ones.

As a conclusion, we introduced some elements of Morse theory and presented the most significant results with regard to 3D shape modeling. In particular, any smooth function can be easily transformed into a Morse function and the critical point setting of a Morse function fully describes the topology of a smooth, compact, connected and orientable manifold.

4.1.3 Reeb graphs

Reeb graphs are abstract representations of a certain subset of Morse functions defined on smooth compact manifolds. They have been introduced

in 1946 by Georges Reeb (Ree46) (figure 4.4). Recently, their use in 3D shape modeling has generated a significant amount of literature.

Beyond the topological descriptive properties they inherit from Morse theory framework, Reeb graphs provide an additional description of the critical points' connectivity setting, through the explicit tracking of level set topology evolution, resulting in a expressive topology descripting representation.

Definition 44 (Level set) *The level set associated to the value $r \in \mathbb{R}$ of a smooth real valued function f defined on a smooth compact d -manifold \mathbb{M} , noted $f^{-1}(r)$, is the set of elements $m \in \mathbb{M}$ such that $f(m) = r$.*

Level sets are also sometimes called *level manifolds*.

Property 6 (Level manifold) *The level sets associated to regular values of a smooth real valued function f defined on a smooth compact d -manifold \mathbb{M} are smooth $(d - 1)$ -manifolds.*

Definition 45 (Connected level set) *The connected level sets associated to the value $r \in \mathbb{R}$ of a smooth real valued function f defined on a smooth compact d -manifold \mathbb{M} are the connected components of the level set $f^{-1}(r)$.*

Definition 46 (Level line) *A level line is a level set of a smooth real valued function f defined on a 2-manifold.*

Definition 47 (Contour) *A contour is a connected component of a level-line.*

As we have seen in the coffee-dunked doughnut analogy, the curve at the intersection between the surface of the doughnut and that of the coffee sees its topology vary at the critical points of the *height* function defined on the doughnut. This observation is explained below.

Definition 48 (Gradient) *The gradient ∇f of a function f is a vector field whose components are the partial derivatives of f .*

Property 7 (Gradient and level line) *The gradient ∇f of a smooth real valued function f defined on a smooth compact d -manifold \mathbb{M} is everywhere locally perpendicular to the level sets of f .*

As a consequence of this property, if a level set of a function defined on a d -manifold fails in some configurations to be a smooth $(d - 1)$ -manifold, then the gradient vector must be the zero vector in these configurations. Reciprocally, level sets are not smooth $(d - 1)$ -manifolds when passing

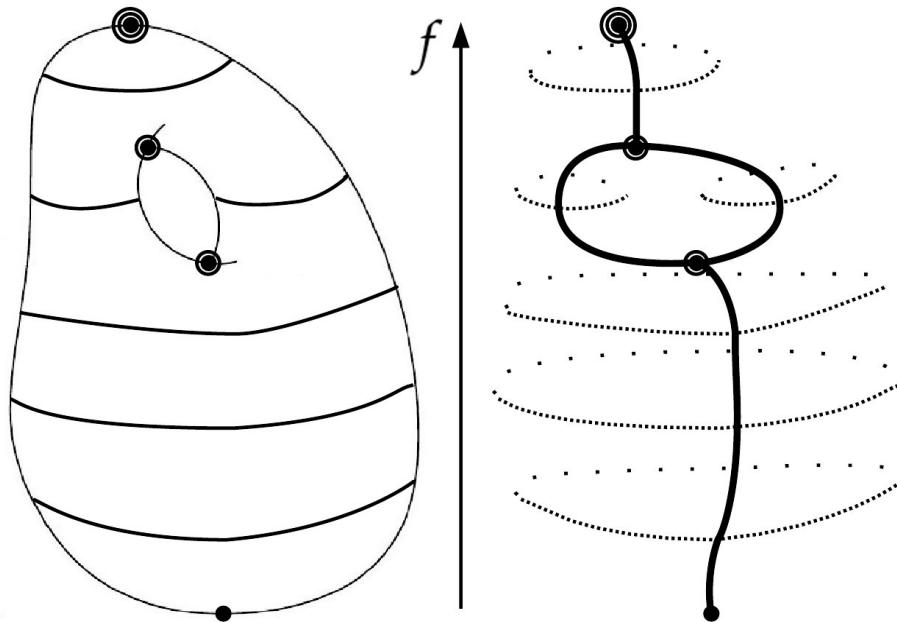


Figure 4.5 – A smooth compact 2-manifold \mathbb{M} and the Reeb graph $R(f)$ of its height function f , both embedded in \mathbb{R}^3 .

through the critical values of f , while they are smooth $(d - 1)$ -manifolds when passing through its regular values (property 6). This observation attests that level set topology transitions only occur when level sets pass through the critical points of f .

While Morse theory mainly focuses on the relation of the critical points of Morse functions and the underlying manifold's topology, Reeb graphs provide an additional information by representing the explicit topology evolution of the level sets of (simple) Morse functions.

Definition 49 (Simple Morse function) A Morse function f defined on a smooth compact d -manifold \mathbb{M} is a simple Morse function if all its critical points have distinct critical values.

Definition 50

(Reeb graph) Let $f : \mathbb{M} \rightarrow \mathbb{R}$ be a simple Morse function defined on a smooth compact d -manifold \mathbb{M} . Let $(f^{-1}(f(m)))_m$ be the connected component of $f^{-1}(f(m))$ containing the element $m \in \mathbb{M}$.

The Reeb graph $R(f)$ is the quotient space on $\mathbb{M} \times \mathbb{R}$ by the equivalence relation $(m_1, f(m_1)) \sim (m_2, f(m_2))$, which holds if :

$$\begin{cases} f(m_1) = f(m_2) \\ m_2 \in (f^{-1}(f(m_1)))_{m_1} \end{cases}$$

Figure 4.5 shows a smooth compact 2-manifold embedded in \mathbb{R}^3 . Some of the level lines of the *height* function f have been drawn on it as well as its critical points (with a number of concentric circles corresponding to the index of the critical point). An embedding of the Reeb graph $R(f)$ appears on the right of the figure, such that each equivalence class is contracted to a point on the bold lines.

As simple Morse functions are everywhere dense in the space of Morse functions, simple Morse functions can be obtained from slight perturbations which modify f in the neighborhood of critical points having the same value. In practice, this property ensures that any smooth function can be used for Reeb graph computation, after sufficient perturbation.

As shown in figure 4.5, the Reeb graph $R(f)$ provides an expressive skeletal representation of the structure of the manifold by capturing the topology evolution of f level sets. Moreover, it benefits from the topology characterization properties of Morse theory framework.

Property 8 (Equivalence class degree), (Ree46) *Let $R(f)$ be the Reeb graph of a simple Morse function f defined on a smooth compact d -manifold \mathbb{M} .*

- *The degree of an equivalence class $c \in R(f)$ of index 0 or d equals 1.*
- *If $d \geq 3$, the degree of an equivalence class $c \in R(f)$ of index 1 or $(d - 1)$ equals 2 or 3.*
- *If $d = 2$, the degree of an equivalence class $c \in R(f)$ of index 1 is 2, 3 or 4.*
- *The degree of all other equivalence classes equals 2.*

In the special case of smooth, connected and orientable compact 2-manifolds, the degree of an equivalence class corresponding to a minimum (index 0) or a maximum (index 2) equals 1 (see figure 4.5). Moreover, the degree of equivalence classes containing saddle points (index 1) always equals 3 (Gra71) (see figure 4.5).

The general properties enumerated by Georges Reeb (Ree46) have also been recently specified to the analysis of the number of loops in Reeb graphs, for the restricted case of 2-manifolds (CMEH*03):

Property 9 (Loops in Reeb graphs of 2-manifolds) *Let $R(f)$ be the Reeb graph of a simple Morse function f defined on a smooth compact connected 2-manifold \mathbb{M} of genus g .*

- *If \mathbb{M} is orientable*
 - *and closed, then $R(f)$ has exactly g loops;*

- and has b boundary components, then $R(f)$ has between g and $2g + b - 1$ loops;
- otherwise, if \mathbb{M} is not orientable
 - and closed, then $R(f)$ has between 0 and $\frac{g}{2}$ loops;
 - and has b boundary components, then $R(f)$ has between 0 and $g + b - 1$ loops.

In the context of 3D shape modeling, considered 2-manifolds are mostly orientable and often closed.

These further results show that, beyond the expressive skeletal representation it provides, the Reeb graph also finely captures the topology of a 2-manifold.

In the following, we translate the concepts introduced so far into the discrete setting, more tangible for computers.

4.2 DISCRETE SETTING

In most cases, it is far more convenient for computers to consider the discrete counter parts of smooth concepts. In this section, we present the notions derived from the previously introduced smooth concepts to the discrete setting and focus on their computation. In particular, we will restrict from now the discussion to 2-manifolds (surfaces).

4.2.1 Manifold simplicial decompositions

First, we introduce through a set of definitions the notion of triangulation.

Definition 51 (Simplex) A d -simplex is the convex hull σ of $d + 1$ affinely independent points of an Euclidean space \mathbb{R}^n , with $0 \leq d \leq n$. d is the dimension of σ .

Definition 52 (Vertex) A vertex v is a 0-simplex of \mathbb{R}^3 .

Definition 53 (Edge) An edge e is a 1-simplex of \mathbb{R}^3 .

Definition 54 (Triangle) A triangle t is a 2-simplex of \mathbb{R}^3 .

Definition 55 (Face) A face τ of a d -simplex σ is the simplex defined by a non-empty subset of the $d + 1$ points of σ , and is noted $\tau \leq \sigma$.

Definition 56 (Simplicial complex) A simplicial complex \mathcal{K} is a finite collection of non-empty simplices $\{\sigma_i\}$, such that every face τ of a simplex σ_i is also in \mathcal{K} , and any two simplices σ_i and σ_j intersect in a common face or not at all.

Definition 57 (Underlying space) *The underlying space of a simplicial complex \mathcal{K} is the union of its simplices $|\mathcal{K}| = \cup_{\sigma \in \mathcal{K}} \sigma$.*

Definition 58

(Triangulation) *The triangulation of a manifold \mathbb{M} is a simplicial complex \mathcal{K} whose underlying space $|\mathcal{K}|$ is homeomorphic to \mathbb{M} .*

To summarize, in the case of a simplicial decomposition of a 2-manifold, the triangulation is a union of 2-simplices (triangles) which is homeomorphic to the surface, such that any couple of its simplices intersects in a common face (edge or vertex) or not at all. If the manifold is smooth, two triangles only intersect in a common edge or not at all.

Definition 59 (Triangulated surface) *A triangulated surface, noted \mathcal{S} , is a triangulation of a 2-manifold \mathbb{M} .*

As a matter of fact, triangulated surfaces turn out to be elegant discrete approximations of 2-manifolds. Indeed, 2-manifolds are topological spaces for which each element has a neighborhood homeomorphic to \mathbb{R}^2 . This property is preserved in the case of surface meshes, where the notion of neighborhood has been translated to the concrete discrete notion of 2-simplex.

Topological invariants

We here precise for triangulated surfaces the topological invariants previously mentioned in the smooth setting.

Definition 60 (Triangulated surface with boundary) *A triangulated surface with boundary is a triangulated surface \mathcal{S} which contains 1-simplices that are a face of only one 2-simplex.*

Definition 61 (Triangulated surface's boundary) *The boundary of a triangulated surface \mathcal{S} is the set noted $\partial\mathcal{S}$ of its 1-simplices that are a face of only one 2-simplex.*

Definition 62 (Connected simplicial complex) *A simplicial complex \mathcal{K} is disconnected if it is the union of disjoint sets of simplices. Otherwise, \mathcal{K} is connected.*

Definition 63 (Connected components of a simplicial complex) *The maximal connected subsets of a simplicial complex \mathcal{K} are called its connected components.*

Definition 64 (Triangulated surface's boundary components) *The connected components of the boundary $\partial\mathcal{S}$ of a triangulated surface \mathcal{S} with boundary are called the boundary components of \mathcal{S} .*

Definition 65 (Genus of a triangulated surface) *The genus g of a triangulated surface \mathcal{S} is the maximal number of finite collections of 1-simplices of \mathcal{S} whose underlying spaces are homeomorphic to non-intersecting connected closed 1-manifolds and which cut \mathcal{S} (creating artificial boundary components) while still preserving the number of connected components of \mathcal{S} .*

Definition 66

(Euler relation II) *Let n_i be the number of i -simplices of a connected and orientable triangulated surface \mathcal{S} , g its genus and b its number of boundary components. The Euler characteristic $\chi(\mathcal{S})$ of \mathcal{S} is a topological invariant given by the following expression:*

$$\chi(\mathcal{S}) = \sum_{i=0}^2 (-1)^i n_i = 2 - 2g - b \quad (4.19)$$

This relation shows that the topological invariants of a 2-manifold can still be captured if a simplicial decomposition of it is considered.

Some geometrical invariants

The notion of geodesics on surfaces (*shortest paths*) comes quite straightforwardly in the discrete setting.

Definition 67 (Geodesic on a triangulated surface) *A geodesic between two 0-simplices v_1 and v_2 of a connected triangulated surface \mathcal{S} is a finite collection of the 1-simplices of \mathcal{S} whose underlying space is homeomorphic to a connected 1-manifold containing v_1 and v_2 such that the sum of the lengths of its 1-simplices, noted $\delta(v_1, v_2)$, is minimal. $\delta(v_1, v_2)$ is called the geodesic distance between v_1 and v_2 .*

In practice, several techniques exist for the computation of geodesics on triangulated surfaces. In particular, accurate geodesics can be computed by forcing the path to cross 2-simplices (SSK*05) (corresponding to the addition of extra simplices on the optimal path). Most of the times, the canonical Dijkstra algorithm (Dij59) (shortest paths in graphs with weighted edges) applied on the graph formed by the sets of 0-simplices and 1-simplices of \mathcal{S} and considering the length of a 1-simplex as its weight provides sufficient precision. Figure 4.6 shows a geodesic between two vertices of a triangulated surface computed with the Dijkstra algorithm, illustrating the fact that geodesics are paths intrinsically defined on the surface.

The definition of other differential geometry tools in the discrete set-

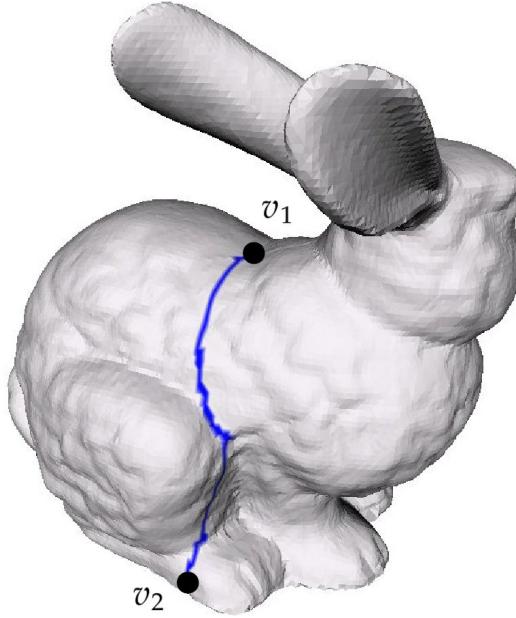


Figure 4.6 – Example of geodesic between two vertices of a triangulated surface of the Stanford bunny 3D shape.

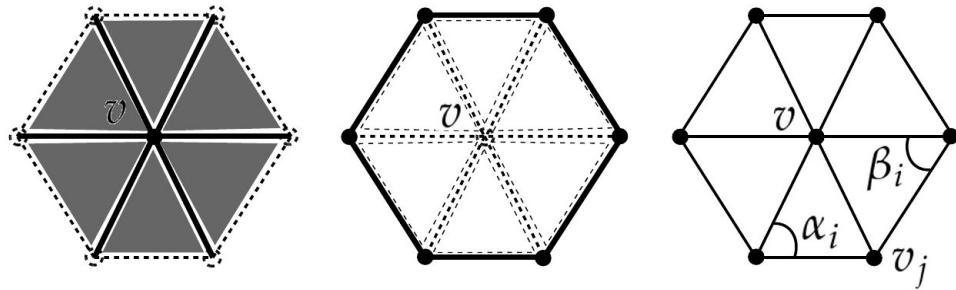


Figure 4.7 – The star $St(v)$ of a vertex v (left), its link $Lk(v)$ (middle) and the opposite angles α_i and β_i of the edge (v, v_i) (right).

ting is less straightforward. In particular, Meyer et al. (MDSBo2) proposed a consistent transposition of curvature concepts to triangulated surfaces.

First, the notion of neighborhood in the discrete setting needs to be introduced.

Definition 68 (Star) *The star of a simplex σ of a simplicial complex \mathcal{K} is the set of simplices of \mathcal{K} that contain σ : $St(\sigma) = \{\tau \in \mathcal{K}, \sigma \leq \tau\}$. We will note $St_d(\sigma)$ the set of d -simplices of $St(\sigma)$.*

Definition 69 (Link) *The link of σ is the set of faces of the simplices of $St(\sigma)$ that do not contain σ : $Lk(\sigma) = \{\tau \in St(\sigma), \tau \cap \sigma = \emptyset\}$. We will note $Lk_d(\sigma)$ the set of d -simplices of $Lk(\sigma)$.*

In other words, the star of a simplex σ is the set of simplices having



Figure 4.8 – The discrete mean curvature (left) and the discrete Gaussian curvature (right) computed on a triangulated surface.

σ as a face (in bold and grey in figure 4.7, left). The notion of link is illustrated in bold in figure 4.7 (middle).

Definition 70 (Discrete mean curvature) *The discrete mean curvature of a triangulated surface \mathcal{S} in one of its vertex v , noted $\kappa_{\mathcal{H}}$, is given by the following expression:*

$$\kappa_{\mathcal{H}}(v) = \frac{1}{4\mathcal{A}} \left| \sum_{v_i \in Lk_0(v)} (\cot \alpha_i + \cot \beta_i) \overrightarrow{vv_i} \right| \quad (4.20)$$

where \mathcal{A} is an area component based on the Voronoi decomposition of $St(v)$ (MDSBo2) and α_i and β_i are the opposite angles of the edge (v, v_i) (figure 4.7, right).

Definition 71 (Discrete Gaussian curvature) *The discrete Gaussian curvature of a triangulated surface \mathcal{S} in one of its vertex v , noted $\kappa_{\mathcal{G}}$, is given by the following expression:*

$$\kappa_{\mathcal{G}}(v) = (2\pi - \sum_{t_i \in St_2(v)} \theta_i) / \mathcal{A} \quad (4.21)$$

where \mathcal{A} is an area component based on the Voronoi decomposition of $St(v)$ (MDSBo2) and θ_i is the angle of t_i at vertex v .

Definition 72 (Discrete principal curvatures) *The discrete principal curvatures κ_1 and κ_2 of a triangulated surface \mathcal{S} in one of its vertex v are given by the following expressions:*

$$\kappa_1(v) = \kappa_{\mathcal{H}}(v) + \sqrt{\kappa_{\mathcal{H}}^2(v) - \kappa_{\mathcal{G}}^2(v)} \quad (4.22)$$

$$\kappa_2(v) = \kappa_{\mathcal{H}}(v) - \sqrt{\kappa_{\mathcal{H}}^2(v) - \kappa_{\mathcal{G}}^2(v)} \quad (4.23)$$

The discrete principal curvatures e_1 and e_2 of a triangulated surface S in one of its vertex v are the two tangent vectors to S that globally and respectively maximizes and minimizes the normal curvature

These concepts are illustrated in figure 4.8, where a color gradient depicts the discrete mean and Gaussian curvatures computed on a triangulated surface, showing that concave and convex regions of the surface can be discriminated through this measure.

4.2.2 Morse theory in the discrete setting

Several authors have investigated the transposition of Morse theory results from the smooth to the discrete setting in the case of triangulated surfaces. In particular, Banchoff (Ban67) proved that the Morse-Euler relation was preserved in the case of piecewise linear general functions sampling Morse functions and Forman (For98) further extended the Morse theory framework.

Definition 73 (Piecewise linear function) *A piecewise linear function (or PL function) defined on a simplicial complex $f : \mathcal{K} \rightarrow \mathbb{R}$ is a function valued on the 0-simplices of \mathcal{K} and whose values on the other simplices of \mathcal{K} are defined by linear interpolation.*

A piecewise linear function on a triangulated surface can be seen as a *sampling* of a function defined on a smooth 2-manifold.

It is clear that piecewise linear functions are not smooth and thus definitely not Morse functions. However, to extend Morse theory results to triangulated surfaces, it is possible to define piecewise linear functions, called PL Morse functions, whose combinatorial properties resemble those of Morse functions in the smooth setting.

First, the notion of critical point for piecewise linear functions needs to be introduced. In the smooth setting, critical points are the points of the manifold where the gradient of the function vanishes. In the discrete setting, most authors (TV98, LV99, EHZ01) focus on the *critical vertices* (or the *piecewise linear critical points*), which are the vertices whose position best approximates that of smooth critical points.

Definition 74 (General piecewise linear function) *A piecewise linear function f defined on a simplicial complex \mathcal{K} is called general if for every couple (v_1, v_2) of its 0-simplices being faces of a common 1-simplex, $f(v_1) \neq f(v_2)$.*

Definition 75 (Lower link, Upper link) *The lower link $Lk^-(v)$ of a vertex v relatively to a*

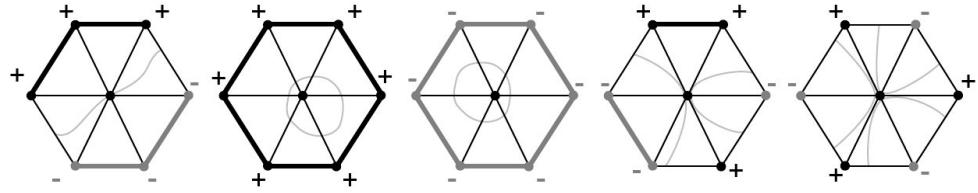


Figure 4.9 – A regular vertex and some critical vertices. From left to right: regular vertex ($n_{Lk^-}(v) = n_{Lk^+}(v) = 1$), a minimum vertex ($n_{Lk^-}(v) = 0, n_{Lk^+}(v) = 1$), a maximum vertex ($n_{Lk^+}(v) = 0, n_{Lk^-}(v) = 1$), a 1-saddle ($n_{Lk^-}(v) = n_{Lk^+}(v) = 2$) and a 2-saddle ($n_{Lk^-}(v) = n_{Lk^+}(v) = 3$).

general piecewise linear function f defined on a simplicial complex \mathcal{K} is the subset of $Lk(v)$ such that:

$$Lk^-(v) = \{\sigma \in Lk(v) \mid f(v_i) < f(v), v_i \leq \sigma\} \quad (4.24)$$

Reciprocally, the upper link $Lk^+(v)$ of v is the subset of $Lk(v)$ such that:

$$Lk^+(v) = \{\sigma \in Lk(v) \mid f(v_i) > f(v), v_i \leq \sigma\} \quad (4.25)$$

The number of connected components of $Lk^-(v)$ and $Lk^+(v)$ are respectively noted $n_{Lk^-}(v)$ and $n_{Lk^+}(v)$.

Definition 76 (Regular vertex) A regular vertex v of a triangulated surface \mathcal{S} relatively to a general piecewise linear function f is a vertex such that $n_{Lk^-}(v) = n_{Lk^+}(v) = 1$.

Definition 77 (Critical vertex) A critical vertex v of a triangulated surface \mathcal{S} relatively to a general piecewise linear function f is a vertex which is not a regular vertex.

- A critical vertex v for which $n_{Lk^-}(v) \neq n_{Lk^+}(v) = 1$ is called a minimum vertex.
- A critical vertex v for which $n_{Lk^+}(v) \neq n_{Lk^-}(v) = 1$ is called a maximum vertex.
- A critical vertex v for which $n_{Lk^-}(v) = n_{Lk^+}(v) = (m + 1)$ with $m \neq 0$ is called a saddle vertex of multiplicity m or m -saddle. A 1-saddle is also called a simple saddle.

Figure 4.9 shows a regular vertex and some critical vertices, with the connected components of $Lk^-(v)$ and $Lk^+(v)$ respectively in bold grey and bold black. In the case of critical vertices, we can notice that the topology of f level lines (bright grey) varies inside $St(v)$, falling to a point or splitting in several connected components (two and three for the 1-saddle and the 2-saddle).

Definition 78 (Degenerate critical vertex) A degenerate critical vertex of a general piecewise linear function f defined on a triangulated surface \mathcal{S} is a m -saddle with $m > 1$. All other critical vertices are said to be non-degenerate.

Definition 79

(Piecewise linear Morse function) A general piecewise linear function f defined on a triangulated surface \mathcal{S} is called a piecewise linear Morse function (or PL Morse function) if all its critical vertices are non-degenerate.

PL Morse functions benefit from most of the properties of their smooth counterpart. In particular, if n_{\min} , n_{\max} and n_{saddle} stand respectively for the number of minimum, maximum and 1-saddle vertices of f , Banchoff (Ban67) proved that the Morse-Euler relation was preserved for connected triangulated surfaces:

$$\chi(\mathcal{S}) = n_{\min} - n_{\text{saddle}} + n_{\max} \quad (4.26)$$

Moreover, a piecewise linear function can be constrained to be general by local perturbation, slightly modifying the f values of vertices sharing an edge and having an equal former f value.

Furthermore, PL functions can be easily transformed into PL Morse functions through a perturbation strategy that transform degenerate vertices into non-degenerate ones. In the case of triangulated surfaces, Edelsbrunner et al. (EHZ01) proposed a technique called the *m-saddle unfolding* that splits m -saddles corresponding to degenerate critical vertices (with $m > 1$) into m 1-saddles of slightly distinct f value, by adding extra simplices in the star of the degenerate critical vertices, as shown in figure 4.10.

As conclusion, perturbing any piecewise linear function with the above strategies can transform it into a PL Morse function, which benefits from the topological descriptive properties of the smooth Morse functions.

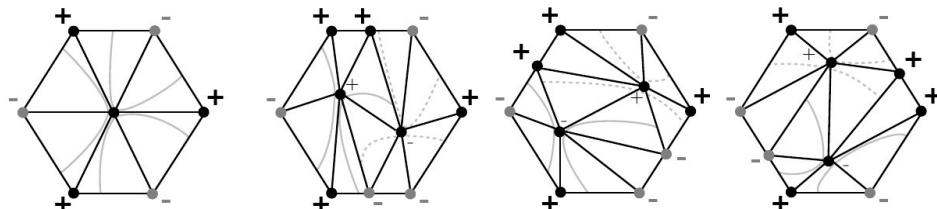


Figure 4.10 – The three possible unfoldings (right) of a 2-saddle (left) into two 1-saddles, with the level lines of the new saddles.

4.2.3 Reeb graphs in the discrete setting

Similarly to the smooth setting, the computation of Reeb graphs on triangulated surfaces requires the considered function to be simple. The concept of *simple piecewise linear Morse function* is easily extended from that of PL Morse function, ensuring the considered function has distinct values at its critical vertices.

However, most Reeb graph computation algorithms require the input function to be distinctly valued on each vertex of the triangulated surface. This is a stronger constraint that still can be satisfied with local perturbation.

The first algorithm for Reeb graph computation in the discrete setting was proposed by Shinagawa et al. (SKK91), requiring $O(n^2)$ steps, with n the number of edges of the triangulated surface. Tarasov and Vyalyi (TV98) proposed an algorithm for the special case of genus zero simplicial complexes that runs in $O(n \log(n))$ steps, with n the number of simplices. Carr et al. (CSAA00) simplified and extended this approach to simplicial complexes of arbitrary dimension. The first Reeb graph computation algorithm for triangulated surfaces of arbitrary genus running in $O(n \log(n))$ steps (with n the number of edges) was proposed by Cole-McLaughlin et al. (CMEH^{*}03). Recently, Pascucci et al. (PSBM07) proposed a sophisticated technique for the computation of Reeb graphs for functions defined on streaming simplicial complexes of arbitrary dimension and genus. Some authors also propose algorithms for topological noise smoothing on Reeb graph, by a topology consistent pruning of the output graph (CSdPo4, PSBM07). Notice also that in the context of shape similarity estimation, some authors (HSKK01, TS05, BMSF06) propose to compute Reeb graph approximations by hierarchies of dichotomies of the function base domain, called *multi-resolution Reeb graphs*. Such representations have shown useful in the context of shape comparison but they might not capture the topological descriptive properties of PL Morse functions since they are not based on critical vertex analysis.

Despite this amount of literature, it is hard to find a formal definition of the transposition of the Reeb graph concept to the discrete setting, like that of smooth Morse functions to PL Morse functions.

Most authors describe informally Reeb graphs in the discrete setting, noted $\mathcal{R}(f)$, as graphs whose nodes are equivalence classes and whose edges are adjacency relations between them. By taking advantage of the combinatorial properties of PL Morse functions (which resemble those

of Morse functions of the smooth setting), the degree of the equivalence classes in the discrete setting is guaranteed to equal that of the smooth setting (property 8, page 78).

Moreover, by forcing f to have distinct values on each vertex, the output graph counts consequently finitely many equivalence classes (as much as vertices in the triangulation) and is indeed a discrete representation of the smooth Reeb graph.

Thus, the computation of a Reeb graph in the discrete setting is seen by certain authors as equivalent to a sorting operation (leading to a $O(n \log(n))$ time complexity). The vertices of the triangulated surface are sorted by increasing f value and level set are virtually updated by sweeping the sorted set of vertices. In particular, branching in the graph occurs only when critical vertices are swept (CMEH*03).

Standard algorithms for Reeb graph computation in the discrete setting, like that described in (CMEH*03), consequently produce a discrete representation of the smooth Reeb graph, whose sampling is determined by the sampling of the smooth surface through its triangulation.

However, a major drawback of that kind of approach is that the equivalence classes of the output Reeb graphs do not explicitly encode contours, they just contract them to points in the graph. Consequently, it is possible to consider a map $\psi : S \rightarrow \mathcal{R}(f)$, mapping a vertex to its equivalence class, but not its *inverse* ψ^{-1} since contours are not explicitly stored in data-structures. Contour reconstruction from discrete Reeb graphs is often seen as an application (vKvOB*97), while it should be part of the computation process as suggested by the definition of Reeb graphs in the smooth setting (definition 50, page 77).

4.3 CHAPTER CONCLUSION

In this chapter, we concentrated all the theoretical preliminaries to the contributions presented in this manuscript. We progressively introduced the concept of surface in the smooth setting and some of its invariants, gave some elements of Morse theory and defined Reeb graphs.

Following a recurrent methodology in Computer Science, we then transposed all these concepts to the discrete setting, more tangible for computers, and gave some discussion elements about the techniques proposed for their computation.

In the following chapters, we present the contributions of this thesis. In particular, we explore the use of Reeb graphs for intrinsic topological rep-

resentation of triangulated surfaces and propose a new shape abstraction based on an enhanced version of Reeb graphs in the discrete setting.

Then, we show its utility in three research problems related to Computer Graphics and Computer Vision.

4.4 FRENCH CHAPTER ABSTRACT

Nous avons concentré dans ce chapitre tous les préliminaires théoriques des contributions présentées dans ce manuscrit. Nous avons progressivement introduit le concept de surface dans le continu ainsi que certains de ses invariants. Nous avons également donné quelques éléments de la théorie de Morse et avons défini les graphes de Reeb.

En suivant une méthodologie récurrente en Informatique, nous avons ensuite transposé tous ces concepts vers le discret, plus tangible pour les ordinateurs, et avons donné des éléments de discussion concernant les techniques existantes pour leur calcul.

Dans les chapitres suivants, nous présentons les contributions de cette thèse. En particulier, nous explorons l'utilisation des graphes de Reeb pour la représentation intrinsèque de surfaces triangulées et proposons une nouvelle abstraction de forme basée sur une version améliorée des graphes de Reeb dans le discret.

Ensuite, nous montrerons l'utilité de cette abstraction intrinsèque de forme dans trois problèmes de recherche liés à l'Informatique Graphique et à la Vision par Ordinateur.

5

ENHANCED TOPOLOGICAL SKELETONS

CONTENTS

5.1	SCIENTIFIC ISSUES	93
5.1.1	Intrinsic and expressive PL Morse function	94
5.1.2	Geometry aware Reeb graph computation algorithm . . .	95
5.2	INTRINSIC AND EXPRESSIVE PL MORSE FUNCTION	96
5.2.1	Feature vertex extraction	96
5.2.2	PL Morse function definition	102
5.3	ENHANCED TOPOLOGICAL SKELETON COMPUTATION	109
5.3.1	Concept of Reeb graph in the discrete setting	109
5.3.2	Reeb graph computation algorithm	113
5.3.3	Data-structures and shape modeling	115
5.3.4	Time complexity	117
5.3.5	Experiments and results	120
5.4	CHAPTER CONCLUSION	125
5.5	FRENCH CHAPTER ABSTRACT	126

THIS chapter presents the first contribution of this thesis. We present a new algorithm for the computation of Reeb graphs which explicitly encodes contour approximations, enabling to study not only their topological but also their geometrical evolution. The resulting representation is called *enhanced topological skeleton*.

We also describe the computation of a piecewise linear function integrating affine and isometric invariance properties, based on feature vertex extraction. The Reeb graph of this function is thus invariant to affine and

isometric transformations and also shows to be an expressive surface representation.

5.1 SCIENTIFIC ISSUES

In the previous chapters of this manuscript, we progressively introduced the problem addressed by 3D shape modeling research. In chapter 2, we introduced the notion of 3D shapes in Computer Science. We showed they were mainly described through boundary representations. In particular, the surface mesh, and specifically the triangulated surface, is a popular way to represent 3D shapes because of its versatility in the contexts of interactive rendering, shape acquisition or computer-aided design.

However, surface mesh is an extrinsic shape representation and it suffers from important variability under different sampling strategies and under canonical shape-non-altering surface transformations, such as affine or isometric transformations.

This observation motivates 3D shape modeling research for the definition and computation of intrinsic surface mesh representations. We reviewed the related work in chapter 3 and showed that Reeb graphs proposed an intrinsic and expressive shape representation. First, Reeb graph invariance properties can be explicitly integrated in the definition of the PL Morse function used for its computation. Secondly, the Reeb graph forms a high-level skeletal representation of the surface (often referred to as *topological skeleton*), which provides an expressive representation of the global structure of the surface and which benefits shape understanding applications. Finally, thanks to its theoretical roots, the Reeb graph efficiently captures the topological invariants of the surface and enables a precise topology control of the surface.

For these reasons, we focus in this thesis on Reeb graphs and further investigate their use in shape modeling. To this end, we presented the theoretical preliminaries related to surfaces and Reeb graphs in chapter 4.

In particular, we identify two main problems to be addressed for the use of Reeb graphs in 3D shape modeling:

- The definition of a PL Morse function f enabling the computation of an *intrinsic* and *expressive* Reeb graph;
- The definition of a Reeb graph computation algorithm that can also capture the surface geometry.

In this chapter, we propose solutions to these two problems.

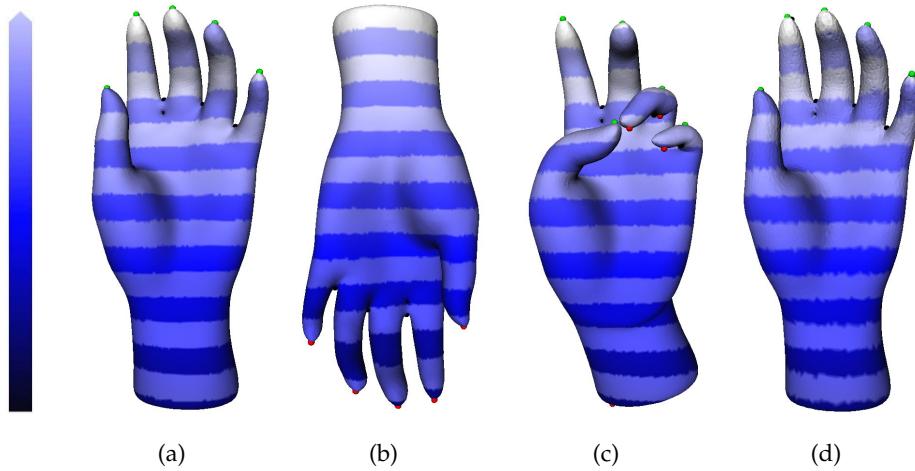


Figure 5.1 – *Shape preserving transformations (affine 5.1(b), near-isometric 5.1(c), under-sampling 5.1(d)) of a triangulated surface and the resulting height functions and their critical vertices.*

5.1.1 Intrinsic and expressive PL Morse function

One of the interest of Reeb graphs is that their invariance property can be explicitly formulated through the definition of the PL Morse function f . Indeed, the geometrical invariance of the Reeb graph is determined by the function f .

Figure 5.1 illustrates the example of chapter 2, which considers shape preserving surface transformations. The color gradient depicts the *height* PL Morse function $f : \mathcal{S} \rightarrow \mathbb{R}$ such that $f(v) = y$ for each vertex of \mathcal{S} . Critical vertices are drawn respectively in red, black and green for minimum, saddle and maximum vertices.

Although these functions are all PL Morse functions (and consequently capture the topological characteristics of \mathcal{S}), their critical vertex configuration varies. For example, after near-isometric transformation (figure 5.1(c)), f counts additional critical vertices on bent fingers which will result in additional branching in the Reeb graph. Consequently, the Reeb graph of the height function is not intrinsic to the surface since it will vary under isometric and affine transformations (notice f is nearly identical after sampling transformation, figure 5.1(d)).

Based on this observation, Lazarus and Verroust (LV99) proposed to consider the PL Morse function which maps a vertex v to its geodesic distance $\delta(v, v_s)$ to a user-selected source vertex v_s . As geodesic distances are intrinsic geometrical surface characteristics (cf. definitions 30 and 67, page 70 and 81), they are invariant under affine and isometric transfor-

mations. To get rid of the source vertex selection problem, Hilaga et al. (HSKK01) propose to integrate this function all over the mesh, at the cost of an increased computational effort. However, this function can generate an important number of critical vertices (and thus a Reeb graph with many *insignificant* branches), whose repartition makes the definition of a simplification strategy difficult.

To overcome this issue, Ni et al. (NGHo4) propose a method to compute *fair* PL Morse functions with a minimum number of critical vertices. However, the critical vertex number reduction process is user-controlled which prevents its application in automatic Reeb graph extraction contexts (such as shape comparison).

In this chapter, we present an automatic technique for the computation of a PL Morse function which is intrinsic to the surface, and whose critical vertex set depicts the most important geometrical features of the surface.

5.1.2 Geometry aware Reeb graph computation algorithm

Standard Reeb graph computation algorithms, like that described in (CMEH^{*}03), just contract connected components of level sets to points. Consequently, given an equivalence class in the Reeb graph, one cannot recover directly the corresponding contour on the surface. Thus, the Reeb graph provided by such techniques cannot capture the geometry of the contours. Generally speaking, they do not provide an output data-structure which exactly reflects the definition of Reeb graph in the smooth setting since the equivalence classes do not explicitly correspond to contour approximations.

To overcome this issue, in the context of shape matching, Hilaga et al. (HSKK01) introduced the concept of *multi-resolution Reeb graph* (MRG), which is a hierarchy of dichotomies of f base domain. Each node of an MRG thus corresponds to a connected set of a pre-image by f of a given interval of \mathbb{R} . Consequently, one can study the geometry of these connected sets. However, multi-resolution Reeb graph, because they do not focus on the critical vertices of f , may not benefit from the combinatorial properties of PL Morse functions and may not preserve the properties of the Reeb graph in the smooth case (missing some loops for example if the interval is too large).

In this chapter, we present a new Reeb graph computation algorithm (which can be seen as an extension of that by Cole-McLaughlin (CMEH^{*}03)). It focuses on the critical vertices of the input PL Morse func-

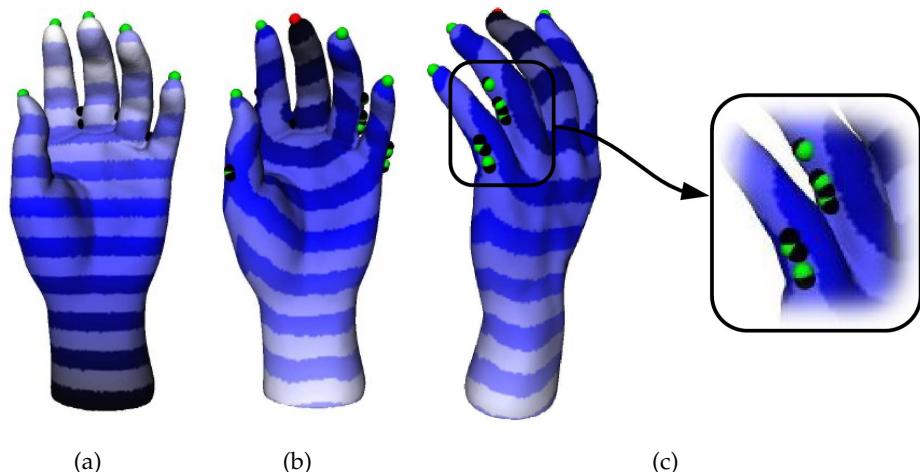


Figure 5.2 – Some PL functions and their critical vertices.

tion and thus captures the surface topological invariants. Moreover, it explicitly encode the equivalence relations at the base of the definition of Reeb graphs in the smooth setting, enabling the geometrical study of contours.

5.2 INTRINSIC AND EXPRESSIVE PL MORSE FUNCTION

In this section, we detail our strategy for the computation of an intrinsic and expressive PL Morse function. To ensure invariance towards affine and isometric transformations, we propose to use geodesic distances as the main surface measurement. Then we propose to focus on *feature vertices* as references for such measurements.

5.2.1 Feature vertex extraction

Feature points is a concept introduced by several authors (MPo2, KLTo5, LZo7) for which it is hard however to find a formal definition. It refers to the points of a surface located at the extremity of its prominent components. Feature points can also be described as the set of points that are the furthest away (in the geodesic sense) from all the other points of the surface.

Algorithm overview

Feature points are characteristic of the shape and correspond to points of geometrical and perceptual interests. Consequently, we propose to use them as reference points for further geodesic measurements.

To prevent the computation of the integral geodesic distance (geodesic distance between each pair of vertices of the triangulated surface), we propose an approximation of the feature point computation based on critical vertex analysis.

Figure 5.2 shows some piecewise linear functions and their set of critical vertices, respectively drawn in red, black and green for minimum, saddle and maximum vertices. Figure 5.2(a) shows that the maximum vertices of the height function ($f(v) = y$) appear on the extremity of the fingers and can be consequently referred to as feature vertices. However, as mentioned above, the height function is extrinsic to the surface. Thus, the set of its critical vertices can vary after isometric transformations. On the contrary, the PL function that maps to a vertex its geodesic distance to a source vertex is intrinsic and does not vary under such transformations. This function $f : \mathcal{S} \rightarrow \mathbb{R}$ such that $f(v) = \delta(v, v_s) \forall v \in \mathcal{S}$ (where $\delta(v, v_s)$ stands for the geodesic distance to the source vertex v_s) has been illustrated in figure 5.2(b). In particular, v_s is the only minimum of f and has been chosen to be on the extremity of the middle finger (in red).

An interesting property of the geodesic distance PL function is that it tends to maximum vertices at the extremity of prominent components (like the fingers, in figure 5.2(b)). Moreover, it is intrinsic to the surface. Consequently, it is a good candidate for feature vertex computation.

However, this function exhibits additional maximum vertices which do not correspond to feature points in specific configurations, as highlighted in figure 5.2(c). The reasons of this observation are the following.

PL approximations of non-differentiable functions

In the smooth setting, it is well known that the minimum of several functions may not be differentiable, depending on the properties of the definition domain. Considering two points p_1 and p_2 of a connected 2-manifold \mathbb{M} , let \mathbb{C} be the space of all possible curves embedded on \mathbb{M} linking p_1 and p_2 and let $f_1, \dots, f_n : \mathbb{M} \rightarrow \mathbb{R}$ be the respective functions mapping p_2 to the length of the curves $C_1, \dots, C_n \in \mathbb{C}$, $n \rightarrow +\infty$. Then, the function that maps a point p_2 to its geodesic distance to a source point p_1 can be viewed as the minimum of several functions: $f = \min_{f_i} f_i(p_2)$.

Consequently, the geodesic distance may not be differentiable in the smooth setting. In particular, it will not be differentiable in the configurations where distinct curves of identical (minimal) length exist. As a result, in these configurations, the gradient of the geodesic distance is not the

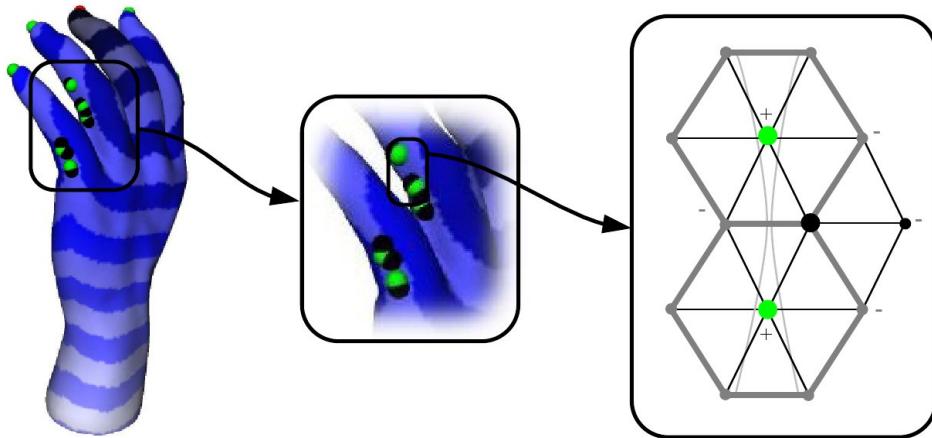


Figure 5.3 – Critical vertices due to the sampling of a non-differentiable function.

zero vector in the points of non-differentiability (it is undefined) and will be directed towards opposite directions in their neighborhood.

In the discrete setting, this phenomenon leads to a distinct observation, illustrated in figure 5.3. In the configurations where several geodesic paths exist (see the level line of the smooth geodesic function in light grey in figure 5.3 right), due to the sampling of the triangulated surface, maximum vertices can appear (in green). As a result, 1-saddle vertices will also emerge in their neighborhood (black vertex with bigger radius in figure 5.3 right). Indeed, the upper link of this vertex has two connected components (composed of the two adjacent maximum vertices) and its lower link has two connected components. Thus, the piecewise linear approximation of non-differentiable functions leads to the appearance of critical vertices referred to as *extra* critical vertices in the configurations of *non-differentiability*.

Moreover, the more the triangulated surface will be dense, the more *extra* critical vertices will appear in these configurations.

Notice however that the PL approximation of a non-differentiable function (and thus non Morse function) can be a PL Morse function as long as its critical vertices are non-degenerate. If we consider the triangulated surface s composed of the neighborhood depicted in figure 5.3 (right), it has one boundary component and genus zero. Thus, its Euler characteristic is given by: $\chi(s) = 2 - 2g - b = 1$. Moreover, the alternated sum of its critical vertex set is: $n_{\min} - n_{\text{saddle}} + n_{\max} = 0 - 1 + 2 = 1 = \chi(s)$ (Morse-Euler relation for PL Morse functions).

The fact that the PL approximation of a non-differentiable function can be a PL Morse function (and thus benefits from Morse function topological

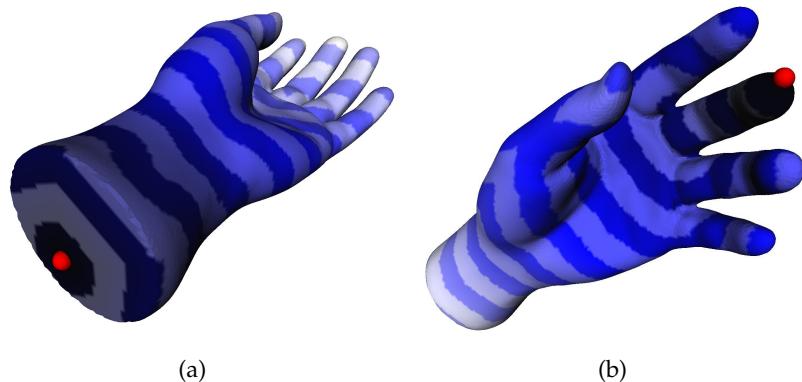


Figure 5.4 – Source vertices v_1 (5.4(a)) and v_2 (5.4(b)) of a triangulated surface (most geodesic-distant vertices).

properties) is an important observation which will be used later on this dissertation.

Feature vertex computation

As shown in figure 5.2, the geodesic distance function tends to maximum vertices at the extremity of prominent components. To distinguish these critical vertices from those that occur at *non-differentiable configurations*, we propose the following strategy. As the geodesic distance tends to maximum vertices in feature point areas even if the position of the source vertex varies, we propose to run a crossed analysis of the maximum vertex set of two geodesic distance functions, relative to two distinct source vertices.

To guarantee that these functions are not too correlated, we propose to select as source vertices the two most geodesic-distant vertices of the mesh. These vertices can be obtained by computing the diameter (Die05) of the graph formed by the set of vertices and edges of the triangulated surface.

Let v_0 be the vertex of an input closed connected and orientable triangulated surface \mathcal{S} such that its euclidean distance to the barycenter of \mathcal{S} is minimal and let $f_0 : \mathcal{S} \rightarrow \mathbb{R}$ map a vertex to its geodesic distance¹ to v_0 :

$$f_0(v) = \delta(v, v_0), \forall v \in \mathcal{S} \quad (5.1)$$

Let v_1 be the vertex that maximizes f_0 on \mathcal{S} and let $f_1 : \mathcal{S} \rightarrow \mathbb{R}$ be the following function:

$$f_1(v) = \delta(v, v_1), \forall v \in \mathcal{S} \quad (5.2)$$

¹ In the rest of this manuscript, geodesic distances are approximated by the Dijkstra algorithm (Dij59).

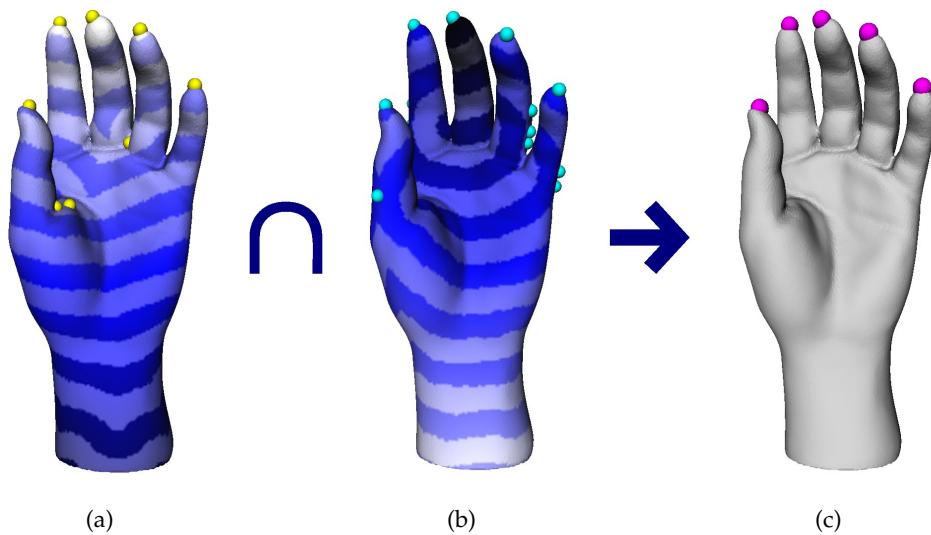


Figure 5.5 – \mathcal{F}_1 , \mathcal{F}_2 and \mathcal{F} sets on a triangulated surface.

Let v_2 be the vertex that maximizes f_1 on \mathcal{S} and let $f_2 : \mathcal{S} \rightarrow \mathbb{R}$ be the following function:

$$f_2(v) = \delta(v, v_2), \forall v \in \mathcal{S} \quad (5.3)$$

Figure 5.4 shows the two source vertex v_1 (figure 5.4(a)) and v_2 (figure 5.4(b)) on the hand triangulated surface. f_1 and f_2 are depicted by the color gradient. $\delta(v_1, v_2)$ is considered as a good approximation of the diameter of the triangulation graph (LV99).

Let \mathcal{F}_1 and \mathcal{F}_2 be respectively the sets of extremum vertices (minima plus maxima) of f_1 and f_2 on \mathcal{S} . We base the *feature vertex* extraction process on the hypothesis that the extremum vertices of f_1 and f_2 corresponding to *feature vertices* occur in the same neighborhoods and that the *extra critical vertices* (due to the *non-differentiability* of the geodesic function in the smooth setting) do not.

This is a faithful hypothesis in the sense that *extra critical vertices* occur when several geodesics exist and that geodesics are intrinsic to the source vertices. Consequently, *extra critical vertices* (which do not correspond to *feature vertices*) are intrinsic to the source vertex choice. Here, we took the precaution to take the most geodesic-distant vertices of the triangulated surface as source vertices to guarantee that the *extra critical vertices* occur in distinct configurations. In figure 5.5, \mathcal{F}_1 vertex set has been displayed in yellow (figure 5.5(a)) and \mathcal{F}_2 vertex set has been displayed in light blue (figure 5.5(b)). f_1 and f_2 tend to maximum vertices in

the configurations of feature points and their *extra* critical vertices occur in distinct configurations.

Then the notion of feature vertices can be defined.

Definition 80 (Feature vertices) *Let v_1 and v_2 be the couple of vertices of a closed connected and orientable triangulated surface \mathcal{S} which maximizes the geodesic distance $\delta(v_1, v_2)$. Let $f_1, f_2 : \mathcal{S} \rightarrow \mathbb{R}$ map respectively each vertex $v \in \mathcal{S}$ to $\delta(v, v_1)$ and $\delta(v, v_2)$. The set of feature vertices \mathcal{F} of \mathcal{S} is the intersection of the sets of extremum vertices of f_1 and f_2 , noted \mathcal{F}_1 and \mathcal{F}_2 :*

$$\mathcal{F} = \mathcal{F}_1 \cap \mathcal{F}_2 \quad (5.4)$$

Due to the sampling of the underlying 2-manifold, the critical vertices of f_1 and f_2 located on the extremity of the prominent components might not occur exactly on the same vertices. Consequently, we define the intersection operator \cap with the following set of constraints, where δ_n stands for the normalized geodesic distance function (to impose scale invariance):

$$v \in \mathcal{F} = \mathcal{F}_1 \cap \mathcal{F}_2 \iff \begin{cases} \exists v_{\mathcal{F}_1} \in \mathcal{F}_1 \quad / \quad \delta_n(v, v_{\mathcal{F}_1}) < \epsilon \\ \exists v_{\mathcal{F}_2} \in \mathcal{F}_2 \quad / \quad \delta_n(v, v_{\mathcal{F}_2}) < \epsilon \\ \delta_n(v, v_i) > \epsilon \quad \forall v_i \in \mathcal{F} \\ \epsilon, \delta_n \in [0, 1] \end{cases} \quad (5.5)$$

Figure 5.5(c) shows in pink five of the six feature vertices of a triangulated surface (the sixth feature point, v_1 , is located under the wrist). This figure attests that feature vertices are located in configurations of geometrical and perceptual interest. In all of the experiments reported in this manuscript, ϵ has been set to 0.05 (see experimental discussion, page 120).

Thanks to the use of geodesic distances, the feature vertex computation is intrinsic to the triangulated surface. Moreover, as it is not based on curvature estimation (which is particularly noisy in the discrete setting), it is more robust than the approach described in (MPo2). Furthermore, on the contrary to (MPo2), it can resolve feature vertex extraction on constant curvature areas. As it is based on a minimal number of geodesic distance computations, it requires a lower computational effort than the approach described in (KLT05) (see page 117 for time-complexity estimation). Finally, due to their present definition, at least two anti-podal feature vertices (v_1 and v_2) are always guaranteed to exist whatever the geometry of the input triangulated surface is.

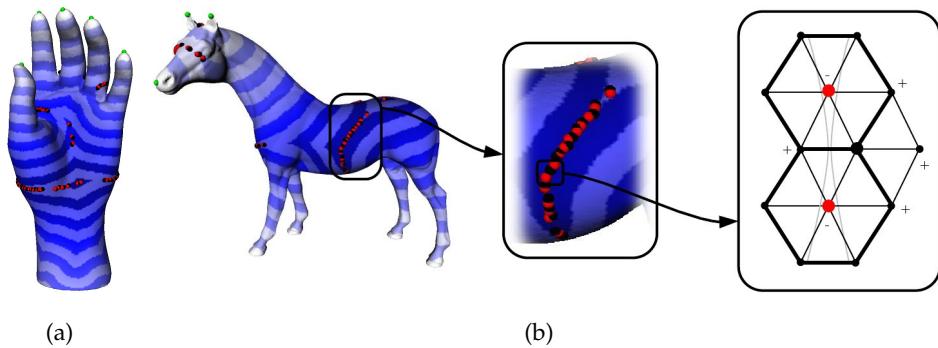


Figure 5.6 – f_I PL function on two triangulated surfaces and its extra critical vertices.

5.2.2 PL Morse function definition

Feature vertices are intrinsic to the triangulated surface and they correspond to important geometrical shape features. Consequently, their are good candidate for reference vertices for the definition of an intrinsic and expressive PL Morse function.

Initial function

To guarantee invariance towards affine and isometric transformations, we base the function definition on geodesic distance measurements. In particular, to impose scaling invariance, we use normalized geodesic distances (normalized with regard to the triangulated surface global minimum and maximum), noted δ_n .

Then, to force the Reeb graphs branches to follow the protrusions of the shape, we define the following initial PL function $f_I : \mathcal{S} \rightarrow \mathbb{R}$:

$$f_I(v) = 1 - \delta_n(v, v_c), \quad \forall v \in \mathcal{S} \quad (5.6)$$

where v_c is the geodesic-closest feature vertex from v :

$$\delta_n(v, v_c) = \min_{v_i \in \mathcal{F}} \delta_n(v, v_i) \quad (5.7)$$

As f_I is defined, it tends to minima in the *center* of the objects and to maxima at the extremity of the shape protrusions, as shown in figure 5.6. Moreover, the fact that f_I level lines follow the protrusions of the shapes (respectively six and seven feature vertices in figures 5.6(a) and 5.6(b)) can also be observed.

f_I is a linear combination of a minimum of several functions. Consequently, a function defined in the smooth setting with the same expression may not be differentiable. In the discrete setting, this property results in



Figure 5.7 – General perturbation strategy of an arbitrary PL function f to a simple Morse function.

the presence of *extra* critical vertices in the areas of the triangulated surface corresponding to the non-differentiable areas of the function in the smooth setting, as shown in figure 5.6(b) (right) and discussed in paragraph *PL approximations of non-differentiable functions* page 97. In particular, f_I exhibits an important number of minimum and saddle vertices in these configurations. Moreover, the more the sampling of the triangulated surface will be dense, the more minimum and saddle vertices will appear. As a consequence, the Reeb graph of f_I will also exhibit an important number of branches in these configurations. To ensure an expressive representation of the shape, we focus in the next paragraph on a strategy that guarantees that the branches of the resulting Reeb graph only correspond to the protrusions of the shape identified by the feature vertices. To this end, we introduce a f_I perturbation strategy that cancels the sets of minimum and saddle vertices appearing in areas of *non-differentiability*.

Function perturbation

To benefit from its topological properties enumerated in the smooth setting (properties 8 and 9 page 78), a Reeb graph in the discrete setting needs to be computed relative to a PL Morse function which is also simple. A canonical solution to the transformation of an arbitrary piecewise linear function into a simple PL Morse function is *perturbation*, as illustrated in figure 5.7.

In particular, the transition from PL functions to PL Morse functions can be achieved by the m -saddle unfolding technique (EHZ01), transforming degenerate critical vertices into sets of non-degenerate ones. A solution to the transformation of PL Morse functions to simple PL Morse functions is the introduction of a perturbation term, guaranteeing that the function's critical vertices have distinct values.

If a PL function is distinctly valued in all of the vertices of \mathcal{S} , then it is obvious it is both a general and simple PL function. Consequently, we propose a perturbation strategy transforming the previously defined f_I PL function into a general and simple PL function by defining perturbations such that f_I takes distinct values in every vertex of the input surface \mathcal{S} .

An example of perturbation strategy is the following. Let $V_{s_0} = \{v_0, v_1, \dots, v_i, \dots, v_{N_V}\}$ be the ordered set of vertices of \mathcal{S} sorted by increasing f values. A possible perturbation transforming f_I into a general and simple function can be obtained by considering the following function $f_{p_0} : \mathcal{S} \rightarrow \mathbb{R}$:

$$f_{p_0}(v) = \frac{i}{N_V} \quad (5.8)$$

where i is the index of v in V_{s_0} and N_V is the number of vertices of \mathcal{S} .

$f_{p_0}(v)$ is an injection and thus general and simple. As it preserves order between vertices relative to f_I , the lower and upper links of \mathcal{S} vertices remain unchanged. Thus f_{p_0} has exactly the same set of critical vertices of f_I and therefore inherits from its combinatorial properties. Notice this perturbation is implicitly used in the Reeb graph computation algorithm described in (CMEH*03), since the triangulated surface is swept by increasing values of the input PL function (then branching in the graph is tracked when sweeping critical vertices).

To get rid of the *extra* critical vertices due to the fact that the expression of f_I in the smooth setting describes a non-differentiable function, we propose to include minimum and saddle vertex cancellation in a perturbation strategy similar to the example presented above.

In particular, we include an additional constraint, referred to as *minimum cancellation constraint* in the sorting of the set of vertices of \mathcal{S} such that the first element of the ordered set becomes the only minimum vertex of the perturbed function. With such a constraint, all other minimum and related saddle vertices (corresponding to the previously mentioned *extra* critical vertices) will then be cancelled.

To implement the minimum cancellation constraint, we start from the following observation: the geodesic distance function to a source vertex admits only one minimum vertex (the source vertex). The geodesic distance function is computed most of the time with the Dijkstra algorithm (Dij59), which can be seen as a sorting algorithm, driven by the connectivity of the graph.

Thus, we propose to use a variant of the Dijkstra algorithm on the graph formed by the vertices and edges of \mathcal{S} but consider as vertex weight its f_I value instead of its approximated geodesic distance, as described in algorithm 1.

Let $V_s = \{v_0, v_1, \dots, v_i, \dots, v_{N_V}\}$ be the ordered sequence of vertices of \mathcal{S} processed by this algorithm (vertex v_i is processed at iteration i and v_0 is f_I global minimum vertex). Then, V_s is sorted with regard to f_I and with

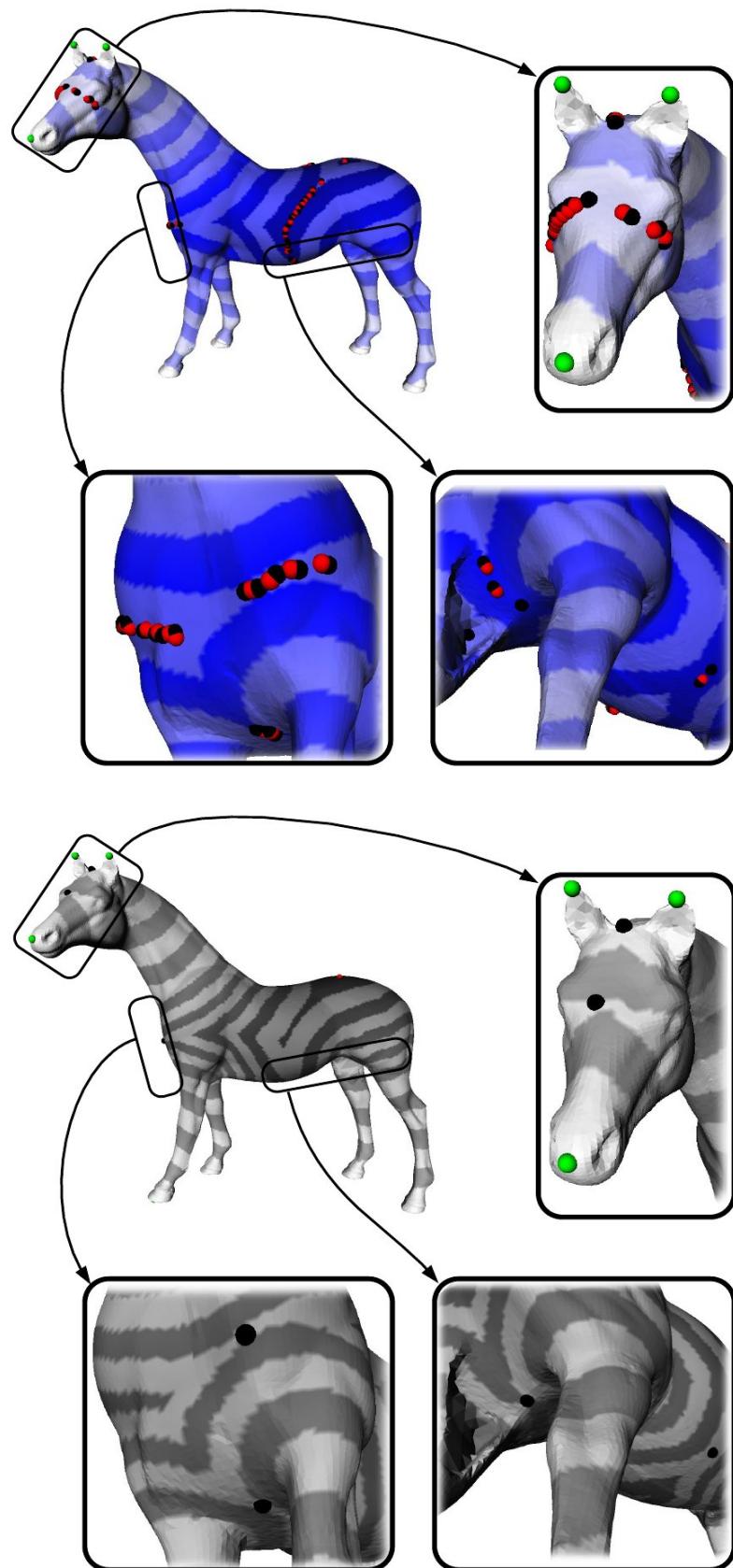


Figure 5.8 – f_1 (top) and f (bottom) critical vertex repartition on the Cyberware horse triangulated surface.

1: f_I value driven Dijkstra based function perturbation.

Data: a triangulated surface \mathcal{S} ;

Data: the function f_I valued on each vertex of \mathcal{S} ;

Result: a perturbed version of f_I , noted f , containing only one minimum vertex.

begin

$i \leftarrow 0$

$visited \leftarrow \emptyset$

$candidates \leftarrow argmin_{v \in \mathcal{S}} f_I(v)$

do

$v_t \leftarrow argmin_{v \in candidates} f_I(v)$

$f(v_t) \leftarrow \frac{i}{N_V}$

$candidates \leftarrow candidates - \{v_t\}$

$visited \leftarrow visited \cup \{v_t\}$

$candidates \leftarrow candidates \cup \{v_n \in Lk_0(v_t) \mid v_n \notin visited\}$

$i \leftarrow i + 1$

while $candidates \neq \emptyset$;

end

regard to the connectivity of the triangulated surface. Then we introduce the perturbation induced by the following function $f : \mathcal{S} \rightarrow \mathbb{R}$:

$$f(v) = \frac{i}{N_V} \quad (5.9)$$

where i is the index of v in V_s . Notice f varies between 0 and 1 to guarantee sampling invariance.

By definition, the only minimum vertex of f is v_0 which is also f_I global minimum vertex. f preserves order between vertices relative to f_I except in the configurations of f_I local minimum vertices, which correspond to *extra* critical vertices.

Figure 5.8 illustrates the result of this perturbation strategy by showing a comparison of the critical vertex set of f_I (blue color gradient) and f (grey color gradient). As the f_I -value based Dijkstra algorithm visits vertices in order of f_I values and of connectivity, *extra* critical vertices (cf. zoomed figures) are processed iteratively and have consequently distinct increasing values. Thus, minimum vertices and their corresponding saddle vertices are cancelled through this perturbation strategy and only the saddle vertices corresponding to f_I level line topology variation remain unchanged (black spheres in bottom figures). Notice moreover that f_I and

f level lines are equivalent (since f is based on f_I) except in the configurations of f_I extra critical vertices.

Finally, to guarantee that f is a PL Morse function, m -saddle unfolding ($m > 1$) is achieved to transform degenerate critical vertices into non-degenerate ones. The overall algorithm for f computation is then given in algorithm 2.

2: f computation algorithm.

Data: a triangulated surface \mathcal{S} ;

Result: an intrinsic and expressive simple PL Morse function

$$f : \mathcal{S} \rightarrow \mathbb{R}.$$

begin

- 1. Feature vertex computation;
- 2. f_I PL function computation;
- 3. f_I -value driven Dijkstra based perturbation;
- 4. m -saddle unfolding ($m > 1$) (EHZo1).

end

f PL Morse function properties

Thanks to our perturbation strategy, f is a bijection. Thus it is both general and simple. Thanks to m -saddle unfolding, it is a PL Morse function. Thus f is a simple (and general) PL Morse function. Consequently, the Reeb graph of f will benefit from all the properties of the Reeb graph defined in the smooth setting (properties 8 and 9 pages 78 and 78).

Moreover, f critical vertex set captures the important geometrical and perceptual features of the shape of the input triangulated surface. Its maximum vertices correspond to the feature vertices and its saddle vertices correspond to valley configurations between protrusions. See also figure 5.9 (bottom row) where f maximum vertices are located at the extremity of the fingers and where its saddle vertices are located in the valleys between fingers. Finally, f admits only one minimum vertex located near the *center* of the object.

For comparison, in figure 5.8, f_I counts respectively 64, 69 and 7 minimum, saddle and maximum vertices. f counts respectively 1, 6 and 7 minimum, saddle and maximum vertices. Notice the Morse-Euler relation

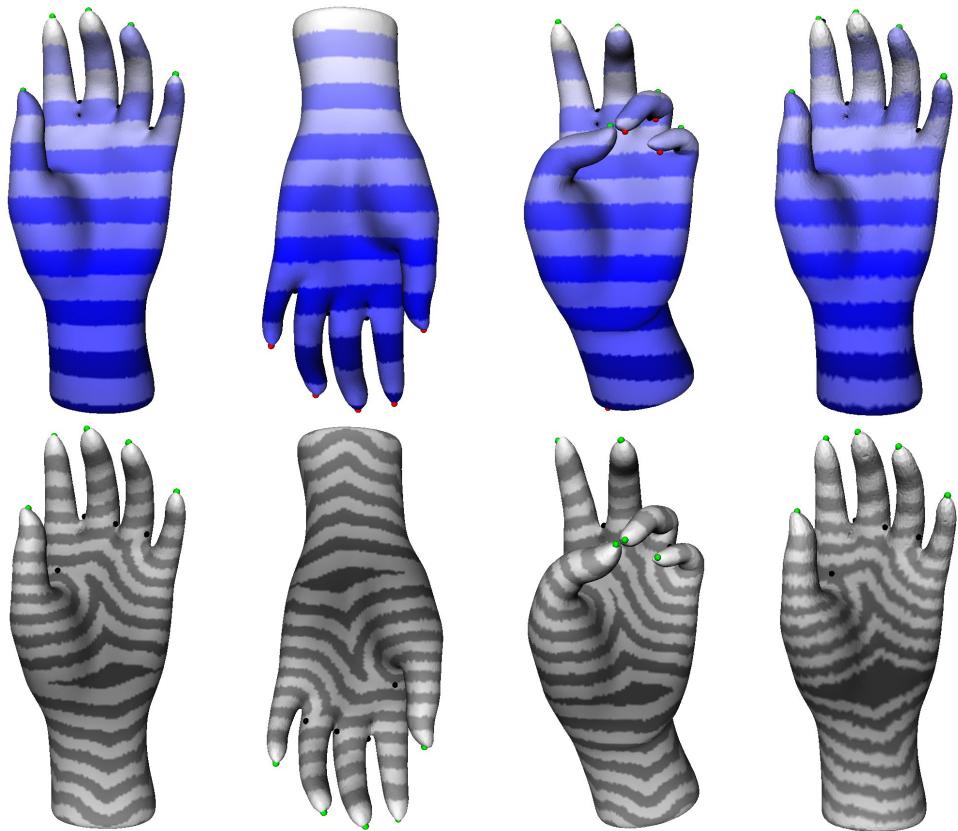


Figure 5.9 – Visual comparison between the height PL Morse function (top) and the f PL Morse function (bottom) after shape preserving transformations (affine, near-isometric and under-sampling).

holds in both cases (the Cyberware horse triangulated surface has genus zero and no boundary component).

Figure 5.9 shows a visual comparison between the invariance properties of f and the *height* function towards shape-preserving transformations (affine, near-isometric and under-sampling). As f is computed from surface intrinsic measurements (geodesic distances), f level lines are equivalent despite these transformations and f critical vertex set remains stable. Notice moreover that feature vertex extraction is also robust despite these transformations, showing that feature points are intrinsic to the shape of the surface.

As a conclusion, we thus presented in this section a technique for the computation of an intrinsic and expressive simple PL Morse function.

5.3 ENHANCED TOPOLOGICAL SKELETON COMPUTATION

As the PL function defined in the previous section, noted f , is a simple PL Morse function, any algorithm based on critical vertex analysis (like that presented in (CMEH^{*}03)) can be used for Reeb graph computation. However, that kind of algorithms just contracts connected components of level lines to points, without any explicit relation from the graph back to the contours on the surface. Consequently, these algorithms provide a symbolic representation of the triangulated surface which does not explicitly capture its geometry.

In this section, we introduce a definition of the Reeb graph in the discrete setting and propose a new algorithm for its computation, providing a data-structure which is more compliant with the definition of Reeb graphs in the smooth setting.

5.3.1 Concept of Reeb graph in the discrete setting

In the smooth setting, the definition of Reeb graphs is based on that of an equivalence relation between contours of a simple Morse function f .

In particular, due to the properties of Morse functions, the critical points of f are the particular points of the surface where the topology of f level lines varies. To present a discrete counterpart of Reeb graphs (like we introduced PL Morse functions relative to smooth Morse functions), we consequently propose to use an analog development.

First, we introduce the notion of *discrete contour*. Then we introduce an equivalence relation at the base of Reeb graph definition.

In the smooth setting, the notion of *sub-level set* is particularly useful in Morse theory.

Definition 81 (Sub-level set) *The sub-level set associated to the value $r \in \mathbb{R}$ of a smooth real valued function f defined on a smooth compact d -manifold $f : \mathbb{M} \rightarrow \mathbb{R}$, noted $f^{-1}(-\infty, r)$, is the set of elements $m \in \mathbb{M}$ such that $f(m) < r$.*

The topology of a sub-level set is closely related to that of the corresponding level set. Indeed, the boundary $\partial f^{-1}(-\infty, r)$ of $f^{-1}(-\infty, r)$ contains the level set $f^{-1}(r)$. If \mathbb{M} has no boundary components, then the connected components of $f^{-1}(r)$ are the only boundary components of $f^{-1}(-\infty, r)$. As a consequence, Morse theory can also be interpreted in term of topology analysis of sub-level sets.

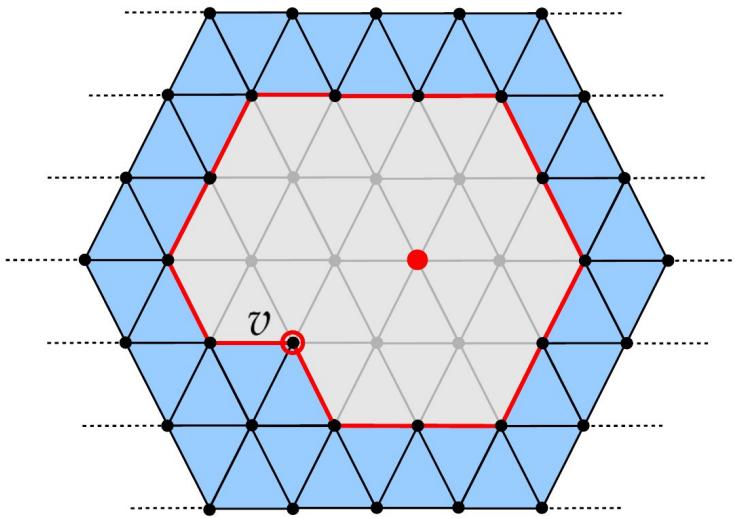


Figure 5.10 – Discrete level line $f^{-1}(f(v))$ around a minimum vertex of f (red).

The level set $f^{-1}(r)$ can be seen as the *upper boundary* of the sub-level set $f^{-1}(-\infty, r)$ (its boundary of highest f value).

Based on this definition of level set, we introduce the concept of discrete level line on a triangulated surface with the following notions:

Definition 82 (Discrete sub-level set) Let f be a PL function defined on a triangulated surface $f : \mathcal{S} \rightarrow \mathbb{R}$ such that it takes distinct values on each 0-simplex of \mathcal{S} . The discrete sub-level set associated to the 0-simplex $v \in \mathcal{S}$, noted $f^{-1}(-\infty, f(v))$, is the set of 2-simplices t_i of \mathcal{S} such that:

$$f^{-1}(-\infty, f(v)) = \{t_i \in \mathcal{S} \mid \exists v_j \leq t_i \mid f(v_j) < f(v)\} \quad (5.10)$$

A discrete level line can then be seen as the upper boundary of a discrete sub-level set:

Definition 83 (Discrete level line) Let f be a PL function defined on a triangulated surface $f : \mathcal{S} \rightarrow \mathbb{R}$ such that it takes distinct values on each 0-simplex of \mathcal{S} . The discrete level line associated to the 0-simplex $v \in \mathcal{S}$, noted $f^{-1}(f(v))$, is the union of the 0-simplices v_i and of the 1-simplices e_j being faces of 2-simplices $t_k \in f^{-1}(-\infty, f(v))$ such that:

$$\begin{aligned} f^{-1}(f(v)) = v \cup \{v_i \leq t_k \mid t_k \in f^{-1}(-\infty, f(v)) \mid f(v_i) > f(v)\} \\ \cup \{e_j \leq t_k \mid t_k \in f^{-1}(-\infty, f(v)) \mid f(v_l) \geq f(v), \forall v_l \leq e_j\} \end{aligned} \quad (5.11)$$

Figures 5.10, 5.11 and 5.12 present examples of discrete sub-level sets (in light grey) and of discrete level lines (in red). In figures 5.10 and 5.12, the discrete level line $f^{-1}(f(v))$ is composed of the vertex v (red circle),



Figure 5.11 – Discrete level lines (in red) on a triangulated surface.

of the red edges and of the vertices of the red edges. Moreover, in figure 5.10, all the vertices in grey have a lower f value than v and all the vertices in black (except v) have an upper f value than v .

Definition 84 (Discrete contour) Let f be a PL function defined on a triangulated surface $f : \mathcal{S} \rightarrow \mathbb{R}$ such that it takes distinct values on each 0-simplex of \mathcal{S} . The discrete contour associated to the 0-simplex $v \in \mathcal{S}$, noted $(f^{-1}(f(v)))_v$, is the connected component of $f^{-1}(f(v))$ containing v .

Then, Reeb graphs in the discrete setting can be introduced.

Definition 85 (Reeb graph of a triangulated surface) Let f be a simple PL Morse function defined on a triangulated surface $f : \mathcal{S} \rightarrow \mathbb{R}$ such that it takes distinct values on each 0-simplex of \mathcal{S} .

The discrete Reeb graph $\mathcal{R}(f)$ is the quotient space on $\mathcal{S} \times \mathbb{R}$ by the equivalence relation $(v_1, f(v)) \sim (v_2, f(v))$, which holds if:

$$\begin{cases} v_1, v_2 \in f^{-1}(f(v)) \\ v_1, v_2 \in (f^{-1}(f(v)))_v \end{cases}$$

Property 10 (Equivalence classes) Let f be a simple PL Morse function defined on a triangulated surface $f : \mathcal{S} \rightarrow \mathbb{R}$ such that it takes distinct values on each 0-simplex of \mathcal{S} and N_V be the number of vertices in \mathcal{S} . Then the discrete Reeb graph $\mathcal{R}(f)$ will be composed of finitely many equivalence classes (N_V equivalence classes).

Figure 5.12 zooms in the star of vertices for which the discrete contour $(f^{-1}(f(v)))_v$ has been drawn:

- In the case of the regular vertex (left), $(f^{-1}(f(v)))_v$ restricted to the star $St(v)$ is composed of two edges and three vertices and is homeomorphic to a 1-manifold.
- In the case of the minimum vertex (middle left), $(f^{-1}(f(v)))_v$ falls to the vertex v itself and thus is homeomorphic to a point.

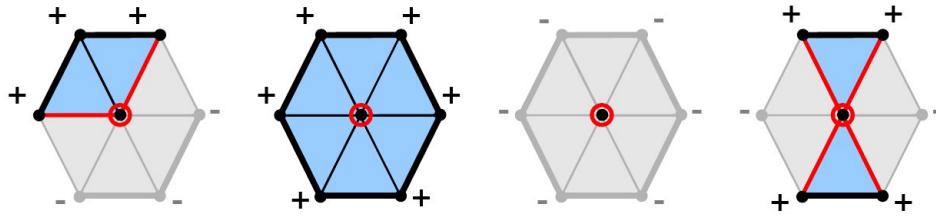


Figure 5.12 – Discrete sub-level sets (light grey) and discrete contours (red) of a regular vertex (left) and of non-degenerate critical vertices (from left to right: minimum, maximum and 1-saddle) on their star.

- The same observation goes in the case of the maximum vertex (middle right).
- In the case of the 1-saddle vertex (right), $(f^{-1}(f(v)))_v$ restricted to the start $St(v)$ is composed of four edges and of five vertices and has the topology of a self-intersecting curve.

These examples show that the topology of the discrete contours remains that of a 1-manifold except at critical vertices where it can vary.

Consequently, branching in the discrete Reeb graph $\mathcal{R}(f)$ will only occur at critical vertices (which are all non-degenerate since f is required to be a simple PL Morse function). Thus, as we introduced it, the discrete Reeb graph is guaranteed to benefit from all the topological properties of its smooth counterpart.

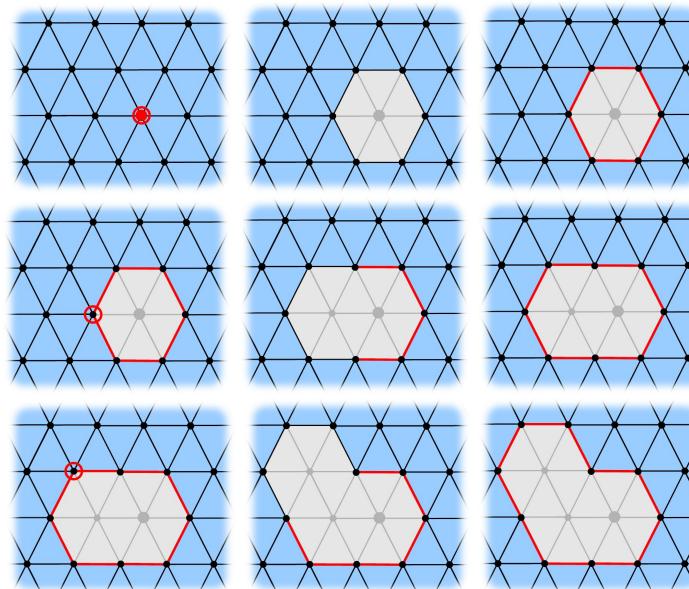


Figure 5.13 – The first three iterations of algorithm 3 on an arbitrary distinctly valued PL Morse function (one iteration per row).

5.3.2 Reeb graph computation algorithm

To compute a data-structure respecting the definition of the discrete Reeb graph (definition 85 page 111), each equivalence class of the Reeb graph has to be explicitly computed. To do so, we propose to use the sweeping approach described in algorithm 3.

3: Discrete Reeb graph computation algorithm.

Data: a connected and orientable triangulated surface \mathcal{S} ;

Data: a distinctly valued PL Morse function $f : \mathcal{S} \rightarrow \mathbb{R}$.

Result: the discrete Reeb graph $\mathcal{R}(f)$.

begin

$min \leftarrow \{v \in \mathcal{S} \mid f(v) < f(v_n), \forall v_n \in Lk_0(v)\}$

$min \leftarrow min - \{\operatorname{argmin}_{v \in \mathcal{S}} f(v)\}$

$sub_level_set \leftarrow \emptyset$

$level_set \leftarrow \{\operatorname{argmin}_{v \in \mathcal{S}} f(v)\}$

do

$v_t \leftarrow \operatorname{argmin}_{v \in level_set} f(v)$

$v_{min} \leftarrow \operatorname{argmin}_{v \in min} f(v)$

if $f(v_{min}) < f(v_t)$ **then**

$v_t \leftarrow v_{min}$

$min \leftarrow min - \{v_{min}\}$

$level_set \leftarrow level_set \cup \{v_t\}$

end

$(f^{-1}(f(v_t)))_{v_t} \leftarrow$ Connected set of $level_set$, containing v_t

$\mathcal{R}(f)$ update

$level_set \leftarrow level_set - \{v_t\}$

$sub_level_set \leftarrow sub_level_set \cup \{v_t\}$

$level_set \leftarrow level_set \cup \{v_n \in Lk_0^+(v_t)\}$

while $level_set \neq \emptyset$;

end

Discrete contour computation

Figure 5.13 shows the first three iterations of algorithm 3 on an arbitrary distinctly valued PL Morse function. At each iteration, the current discrete sub-level set is drawn in light grey and the discrete contour relatively to the vertex v is drawn with red edges (v has a red circle).

At the beginning of each iteration t , sub_level_set contains the set of vertices $\{v \in \mathcal{S} \mid f(v) < f(v_t)\}$. Moreover, at the end of an iteration,

level_set is updated with the vertices of the upper link of v_t . Thus *level_set* contains the discrete level line of the next vertex to process. In particular, it follows from definition 83 that v minimizes f on $f^{-1}(f(v))$. Consequently, each updated level set is associated with its vertex which minimizes f ($v_t \leftarrow \operatorname{argmin}_{v \in \text{level_set}} f(v)$).

Finally, $(f^{-1}(f(v_t)))_{v_t}$ is computed by visiting in a recursive fashion *level_set*, starting in v_t . Consequently, at the end of algorithm 3, each vertex v of \mathcal{S} is equipped with its discrete contour $(f^{-1}(f(v)))_v$ (its equivalence class).

Equivalence class connection

Equivalence classes (discrete contours) are connected to each other to form the discrete Reeb graph of f by the $\mathcal{R}(f)$ update procedure of algorithm 3. This procedure keeps track of the creation, termination, splitting and merging of discrete contours. In particular, as f is required to be a simple PL Morse function, only four cases have to be considered:

1. v_t is a minimum vertex of f :

A new branch \mathcal{B} in the Reeb graph $\mathcal{R}(f)$ is created and $(f^{-1}(f(v_t)))_{v_t}$ is appended to it;

2. v_t is a regular vertex of f :

$(f^{-1}(f(v_t)))_{v_t}$ is appended to the branch \mathcal{B} of $\mathcal{R}(f)$ containing the equivalence classes of the vertices of $Lk_0^-(v_t)$;

3. v_t is a maximum vertex of f :

$(f^{-1}(f(v_t)))_{v_t}$ is appended to the branch \mathcal{B} of $\mathcal{R}(f)$ containing the equivalence classes of the vertices of $Lk_0^-(v_t)$;

4. v_t is a 1-saddle vertex of f . Then it is either a *splitting saddle* or a *merging saddle*:

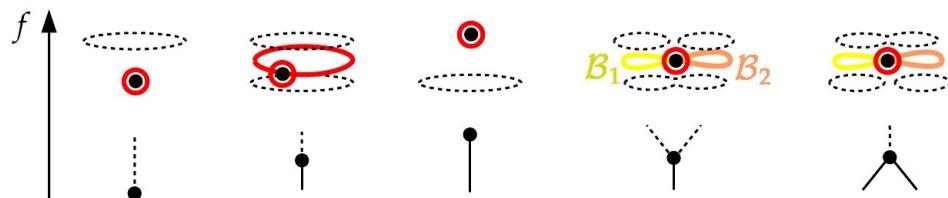


Figure 5.14 – Connecting the equivalence classes in the Reeb graph: discrete contour configuration (top row) and corresponding connection in the Reeb graph (bottom row). From left to right: minimum vertex (case 1), regular vertex (case 2), maximum vertex (case 3), splitting 1-saddle (case 4(a)) and merging 1-saddle (case 4(b)).

- (a) if the equivalence classes of all the vertices of $Lk_0^-(v_t)$ belong to the same branch \mathcal{B} of $\mathcal{R}(f)$, then v_t is a *splitting saddle*.
 - $(f^{-1}(f(v_t)))_{v_t}$ is appended to \mathcal{B} , two new branches \mathcal{B}_1 and \mathcal{B}_2 are created and $(f^{-1}(f(v_t)))_{v_t}$ is also appended to them. Moreover, each vertex $v \in (f^{-1}(f(v_t)))_{v_t}$ is *noticed* to be appended either to \mathcal{B}_1 or \mathcal{B}_2 when it will be processed (instead of \mathcal{B}), as illustrated in figure 5.14.
- (b) if the equivalence classes of the vertices of $Lk_0^-(v_t)$ belong to two distinct branches \mathcal{B}_1 and \mathcal{B}_2 of $\mathcal{R}(f)$, then v_t is a *merging saddle*.
 - $(f^{-1}(f(v_t)))_{v_t}$ is appended to \mathcal{B}_1 and \mathcal{B}_2 . A new branch \mathcal{B} is created and $(f^{-1}(f(v_t)))_{v_t}$ is appended to it. Moreover, each vertex $v \in (f^{-1}(f(v_t)))_{v_t}$ is *noticed* to be appended to \mathcal{B} when it will be processed (instead of \mathcal{B}_1 or \mathcal{B}_2).

Vertex classification (regular, minimum, maximum, splitting or merging saddle) can be achieved by analyzing the link of the vertices (cf figure 4.9 page 85) or equivalently by keeping track of the variation of the number of connected subsets of *level_set* in algorithm 3.

Notice the $\mathcal{R}(f)$ *update* procedure connects equivalence classes in a way compliant to the properties of the Reeb graph in the smooth setting (property 8 page 78): equivalence classes' degree respectively equals 2 for regular vertices, 1 for minimum and maximum vertices and 3 for saddle vertices.

The output data-structure of algorithm 3 is indeed a discrete Reeb graph, a graph whose nodes are equivalence classes associated to connected components of level lines and whose structure captures the topology of the input triangulated surface. On the contrary to the algorithm presented by Cole-McLaughlin et al. (CMEH*03), our algorithm explicitly associates each equivalence class to its corresponding contour $(f^{-1}(f(v)))_v$ instead of just contracting it to a node in the graph. Consequently, the output data-structure also captures the triangulated surface geometry and enables further geometrical study of f level sets.

5.3.3 Data-structures and shape modeling

The output data-structure of algorithm 3 can be described as follows.

Equivalence classes

First, each equivalence class $(f^{-1}(f(v)))_v$ records the following pieces of information:

- The associated vertex v ;
- The associated f value ($f(v)$);
- The list of vertices $v_i \in (f^{-1}(f(v)))_v$;
- The list of edges $e_j \in (f^{-1}(f(v)))_v$.

Reeb graph

The notion of *branch* of the Reeb graph is modeled by a list of contiguous equivalence classes. The notion of *edge* can be preferred to make a full analogy to graph theory.

Definition 86 (Reeb graph edge) *Let $\mathcal{R}(f)$ be the Reeb graph of a simple PL Morse function f defined on a triangulated surface $f : \mathcal{S} \rightarrow \mathbb{R}$. The set $\mathcal{E} = \{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_i, \dots, \mathcal{E}_n\}$ of maximally connected unions of equivalence classes of regular vertices of f is called the set of edges of $\mathcal{R}(f)$. Each set \mathcal{E}_i is an edge of $\mathcal{R}(f)$.*

It follows from this definition that the equivalence classes corresponding to critical vertices do not belong to edges. As a matter of fact, since they have a degree distinct from 2 in the graph, they play a specific role and can be seen as nodes.

Definition 87 (Reeb graph node) *Let $\mathcal{R}(f)$ be the Reeb graph of a simple PL Morse function f defined on a triangulated surface $f : \mathcal{S} \rightarrow \mathbb{R}$. The set of equivalence classes corresponding to critical vertices of f is the set of nodes $\mathcal{N} = \{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_i, \dots, \mathcal{N}_n\}$ of $\mathcal{R}(f)$.*

As a consequence, the output data-structure of algorithm 3 is a graph, whose nodes are the equivalence classes of the critical vertices of f and whose edges are contiguous connected sets of the equivalence classes of the regular vertices of f .

Consequently, each edge is equipped with a collection of contiguous equivalence classes (see the data-structure defined previously). As contours are processed by increasing f value, these collections are also sorted by f value.

Shape modeling properties

On the contrary to the algorithm presented by Cole-McLaughlin et al. (CMEH^{*}03), our algorithm explicitly stores in the graph the corresponding contour of each equivalence class. If the input triangulated surface counts N_V vertices, the output Reeb graph will then counts N_V equivalence classes (and contours) and thus provides the finest possible contour-based representation of the surface.

Consequently, the proposed data-structure captures the surface geometry in the sense it is a **complete** representation of the input triangulated surface. Indeed, the union of all the equivalence classes of the Reeb graph provides the input triangulation.

This enhanced data-structure is then very attractive for shape modeling applications since it efficiently captures the surface topology and its geometry. In particular, it enables to study not only the topological evolution of the level sets of f but also their geometrical evolution by computing geometrical measurements on each contour of an edge of the Reeb graph. For these reasons, we refer to the output data-structure of algorithm 3 as an *enhanced topological skeleton*.

Finally, we showed the PL Morse function f defined in section 5.2 is intrinsic to the triangulated surface. Its critical vertices correspond to important geometrical and perceptual features (such as feature vertices and valley configurations between protrusions). As f is based on intrinsic surface measurements, it is invariant towards affine and isometric transformations and shows to be robust towards triangulation under-sampling. The enhanced topological skeleton of this function inherits from these intrinsic and expressive properties and thus is a good candidate for shape modeling problems.

5.3.4 Time complexity

In this section we study the time complexity of the presented algorithms with regard to the number of vertices N_V of the input triangulated surface.

In particular, the time complexity of these algorithms is determined by the way ordered lists are implemented. Most of them are implemented with a balanced search tree (CLR94). If the balanced search tree contains n elements, then these standard operations have the following time complexity:

- Removal of the first element of the ordered set: $O(\log(n))$;

- Insertion of an element in the ordered set: $O(\log(n))$;

Feature vertex computation

The extraction of the feature vertices of the input triangulated surface \mathcal{S} requires two basic operations.

1. **Vertex classification** (regular, minimum, saddle and maximum):

- At each vertex v , the number of steps required is determined by the number $|Lk_0(v)|$ of vertices in $Lk_0(v)$. $|Lk_0(v)|$ is generally negligible with regard to N_V and can be considered as a constant for all the vertices. Thus, the vertex classification operation requires $O(N_V)$ steps.

2. **Geodesic distance computation:**

- Geodesic distances are computed using the Dijkstra algorithm (Dij59). Its complexity is determined by the way ordered sets are implemented. Here, they are implemented by balanced search tree. Consequently, computing the geodesic distance of the N_V vertices of \mathcal{S} to a source vertex requires $O(N_V \log(N_V))$ steps in the worst case.

As a consequence, the overall complexity of the extraction of the feature vertices of \mathcal{S} is bounded by that of geodesic distance computation and thus is $O(N_V \log(N_V))$. Notice moreover, that geodesic distances are computed only three times (f_0 , f_1 and f_2).

PL Morse function computation

The computation of the function f defined in section 5.2 page 96 (and used for enhanced topological skeleton computation) requires two main operations:

1. **Geodesic distance computation:**

- By using a balanced search tree in the Dijkstra algorithm, the computation of the geodesic distance of the N_V vertices of \mathcal{S} to a source vertex requires $O(N_V \log(N_V))$ steps in the worst case.

2. **Function perturbation:**

- The perturbation approach presented in algorithm 1 (page 106) visits each vertex of \mathcal{S} only once. In particular, at each iteration

the removal of the first element of the ordered set *candidates* and the insertion of a new element both take $O(\log(N_V))$ steps (balanced search tree implementation). Consequently, the overall perturbation algorithm takes $O(N_V \log(N_V))$. As for m -saddle unfolding ($m > 1$), this operation is linear in the number of degenerate saddle vertices.

The overall complexity for the computation of the function f defined in section 5.2 is thus bounded by $O(N_V \log(N_V))$. Notice that the number of geodesic distance computations equals the number of feature vertices $|\mathcal{F}|$, which depends on the surface geometrical complexity and which is negligible with regard to N_V .

Enhanced topological skeleton computation

The enhanced topological skeleton computation described in algorithm 3 visits one vertex per iteration. Indeed, vertices are added only once to the ordered set *level_set* and put definitely in the set *sub_level_set* when they are removed from *level_set*.

At each iteration t , the vertex v_t is removed from the ordered set *level_set* and the vertices $\{v_n \in Lk_0^+(v_t)\}$ are also inserted in *level_set*. The ordered set *level_set* is implemented with a binary search tree. Consequently, removal and insertion are performed in $O(\log(N_V))$ steps in the worst case.

The copy of the contour $(f^{-1}(f(v_t)))_{v_t}$ in the equivalence class data-structure runs in linear time and takes $O(N_V)$ steps in the worst case ($O(\sqrt{N_V})$ is a good approximation in the average case). Consequently, the time complexity of an iteration of the algorithm 5.2 is bounded by that of the contour copy and thus is $O(N_V)$ in the worst case.

As the algorithm runs as many iterations as vertices in \mathcal{S} , the enhanced topological computation thus requires $O(N_V^2)$ iterations in the worst case ($O(N_V \sqrt{N_V})$ is a good approximation in the average case). Notice the computational effort overhead with regard to the algorithm presented by Cole-McLaughlin et al. (CMEH^{*}03) is essentially due to the storage operation of the discrete contours in the equivalence class data-structure (which is the main advantage of the algorithm from a shape modeling point of view).

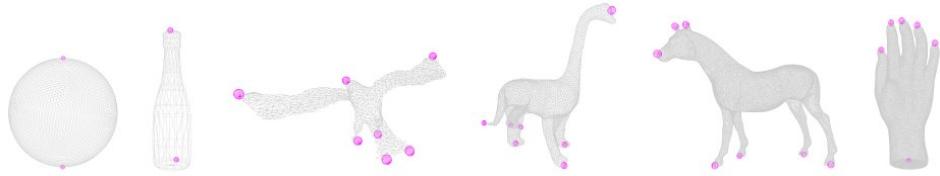


Figure 5.15 – Feature vertices of several triangulated surfaces.

5.3.5 Experiments and results

This section presents experimental results of enhanced topological skeleton computations, including computation times and examples of computations on various triangulated surfaces under a variety of transformations.

Experimental setup

All the algorithms presented in this chapter (and in the rest of this manuscript) have been entirely implemented in C language, on top of two free software libraries: the GNU Triangulated Surface library (GTS, version 0.7.6) (gts06) and the GNU Scientific Library (GSL, version 1.11) (gslo8). Programs have been compiled with the GNU Compiler Collection (GCC, version 4.1.2) (gcc08) and have executed on a GNU/Linux operating system (kernel version 2.6.24) running on a desktop PC with a 3GHz P4-CPU and 2 Gb of RAM.

The triangulated surfaces presented in this section are extracted from the Gamma (gamo3) and the AIM@SHAPE (aimo6) 3D shape repositories.

Parameter setting

The only parameter that intervenes in the approach is the ϵ parameter, which denotes the normalized geodesic radius of the intersection neighborhood in feature vertex extraction (cf. page 99). This value has been set to $\epsilon = 0.05$ for all the experiments reported in the manuscript. Figure 5.15 shows the feature vertices (in pink) of several triangulated surfaces with this setting.

Even if the shape of the input triangulated surface does not exhibit any particular protrusion, the feature vertex extraction algorithm guarantees that at least two antipodal feature vertices are extracted, so that PL Morse function defined in section 5.2 (page 96) can always be computed. This property can be observed on the sphere (figure 5.15, left) where two poles are extracted as feature vertices.

Triangulated surface	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.2$
Sphere	2	2	2	2	2
Bottle	2	2	2	2	2
Bird	9	9	7	7	6
Dinosaur	6	8	7	7	7
Horse	6	6	7	7	7
Hand	2	5	6	6	8

Table 5.1 – Number of feature vertices with different ϵ values (default value: $\epsilon = 0.05$).

Triangulated surface	Triangles	f computation (s.)	ETS (s.)	Total (s.)
Fan (5.18(a))	456	0.052	0.008	0.060
Bottle (5.18(b))	512	0.033	0.014	0.047
Bird (5.18(c))	1996	0.295	0.130	0.425
Bi-torus (5.18(d))	6004	0.549	0.644	1.193
Sphere (5.18(e))	8000	0.664	1.394	2.058
Dinosaur (5.18(f))	9989	1.813	1.323	3.136
Boy (5.18(g))	16878	5.184	2.726	7.910
Horse (5.18(h))	39996	12.041	14.499	26.54
Hand (5.18(i))	52000	12.908	51.439	64.347

Table 5.2 – Computation times for f computation and perturbation (including feature vertex extraction), enhanced topological skeleton computation (ETS) and for the overall process.

Table 5.3.5 shows that the number of extracted feature vertices is quite stable when ϵ varies. In table 5.3.5, the column corresponding to the default setting for ϵ (used in all the experiments of this manuscript) is bold ($\epsilon = 0.05$).

Computation times

Table 5.2 reports the computation times for the f PL Morse function computation and perturbation (including feature vertex extraction), for the enhanced topological skeleton extraction and for the overall process for the triangulated surfaces presented in figure 5.18.

Figure 5.16 shows the computation time of f computation with regard to the number of triangles of the input triangulated surface in comparison to the $O(n \log(n))$ function. Figure 5.17 shows the computation time of the enhanced topological skeleton extraction with regard to the number

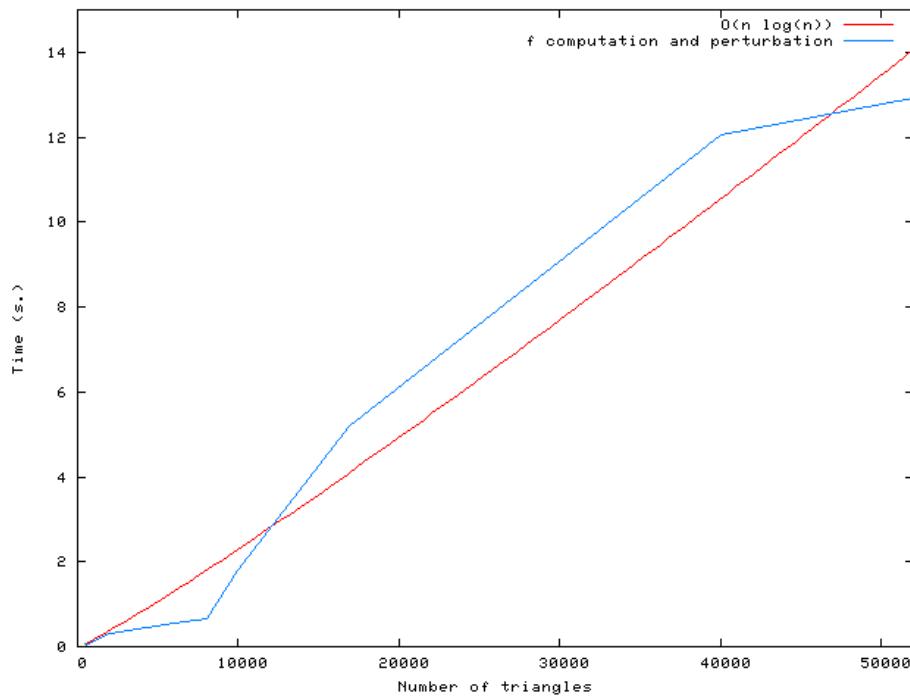


Figure 5.16 – f PL Morse function computation time (including feature vertex extraction and perturbation) with regard to the number of triangles of the input triangulated surface.

of triangles of the input triangulated surface in comparison to the $O(n^2)$ function. These figures show that the experimental measurements of computation times validate the time complexity estimation given in section 5.3.4, page 117.

Shape modeling properties

The Reeb graph is an abstract representation of the surface. Consequently, to give a visual representation of it, an embedding strategy in \mathbb{R}^3 has to be specified.

In particular, to underline the way enhanced topological skeletons efficiently capture the surface topology, we propose the following symbolic embedding, based on a dual representation where the edges of the Reeb graph are represented by nodes and the adjacency relations between them by edges. In particular, we embed a node of a dual Reeb graph at the barycenter of all the contours of its equivalence classes. The resulting embedding is shown in figure 5.18 for several triangulated surfaces.

In this figure, we can observe that the enhanced topological skeleton indeed inherits from the topological properties of the Reeb graph in the smooth setting. In particular, notice that the maximum degree of a node is three, which correspond to the degree of an equivalence class of a saddle

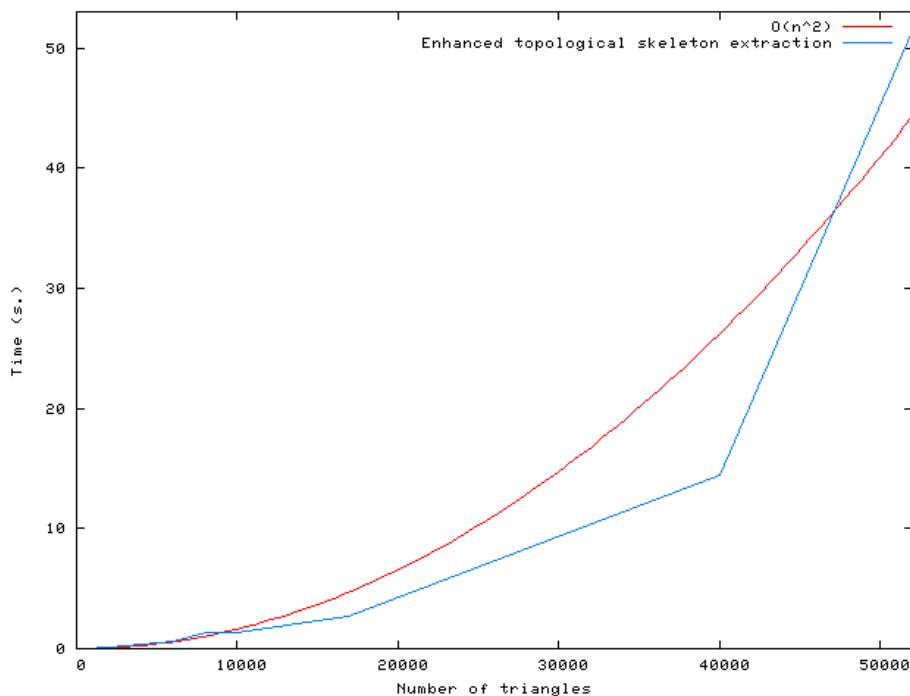


Figure 5.17 – Enhanced topological computation time with regard to the number of triangles of the input triangulated surface.

vertex. Thus, the enhanced topological skeleton efficiently captures the topology of the surface. In figure 5.18(d), the bi-torus has genus two and the enhanced topological skeleton thus has two loops.

If the shape of the triangulated surface does not exhibit any particular protrusion, the feature vertex extraction algorithm guarantees that at least two antipodal feature vertices are extracted. This can be observed for the surfaces of figures 5.18(b) and 5.18(e). In particular, the enhanced topological skeleton of these two surfaces are equivalent since their shape has no particular protrusion and their surfaces are homeomorphic.

Moreover, the figure 5.18 shows that no noisy geometrical details are encoded in the enhanced topological skeleton. Thanks to the feature vertex extraction algorithm, only the important geometrical features generate branches in the graph, providing an expressive shape representation which is beneficial for shape understanding applications.

The figure 5.19 shows the invariance properties of the f PL Morse function defined in section 5.2 (page 125) and its robustness towards surface under-sampling and surface noise. In particular, in figure 5.19(e), the position of each vertex of the triangulated surface has been randomly moved with an amplitude of $\pm 1\%$ of the lengths of the bounding box of the whole

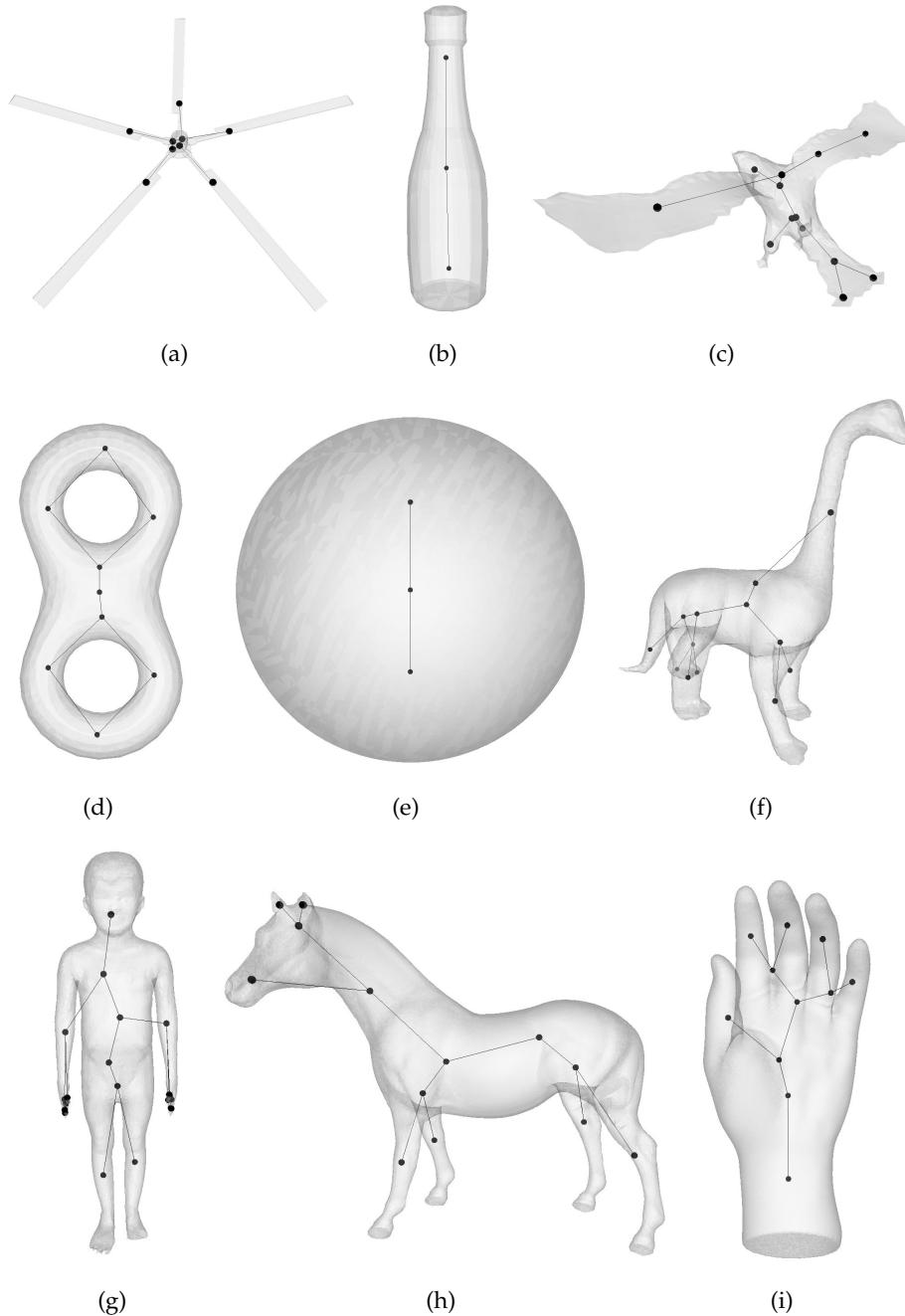


Figure 5.18 – Symbolic embedding of the enhanced topological skeleton of f on several triangulated surfaces.

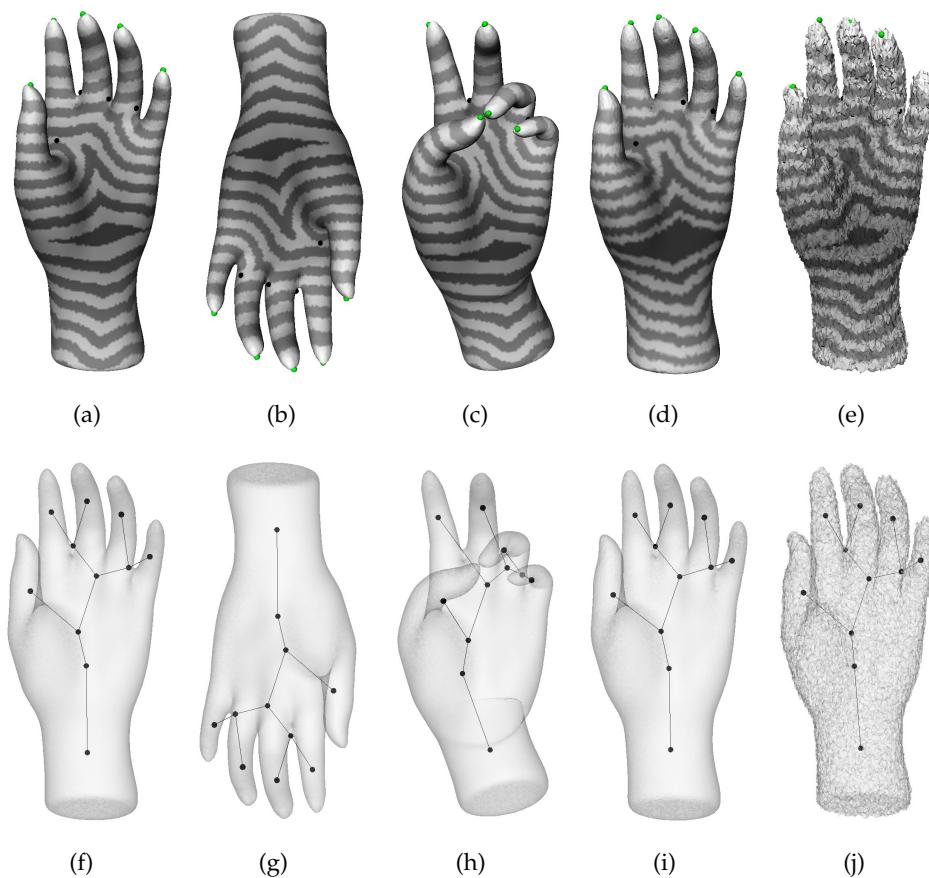


Figure 5.19 – f PL Morse function invariance and robustness towards shape-preserving transformations (affine and near-isometric), under-sampling and random noise and the symbolic embedding of the resulting enhanced topological skeleton.

triangulated surface. This noise introduces a similar (but amplified) perturbation as that introduced by scanning noise.

Despite these surface transformations, f only slightly varies. Consequently, its contours vary in a compatible way and the structure of the resulting enhanced topological skeleton remains unchanged. Consequently, the enhanced topological skeleton of f is an intrinsic and stable representation of the input triangulated surface.

5.4 CHAPTER CONCLUSION

In this chapter, we investigated the use of differential topology for intrinsic and expressive 3D shape modeling. In particular, we proposed to represent the input triangulated surface by an enhanced version of the discrete Reeb graph of a PL Morse function f based on geodesic distances to feature vertices. We introduced two main contributions.

First, we presented an algorithm for the extraction of the feature vertices of the triangulated surface (vertices located at the extremity of the protrusions of the shape). This algorithm is fast and stable and computes an intrinsic set of feature vertices.

Second, we developed a formal definition of Reeb graphs in the discrete setting and proposed an algorithm for their computation which provides output data-structures more compliant with the definition of Reeb graphs in the smooth setting than previous approaches. This enhanced version of discrete Reeb graphs has been referred to as *enhanced topological skeleton*.

Experiments showed that the enhanced topological skeleton of the f function is an intrinsic triangulated surface representation which is both stable and expressive. Consequently it is a good candidate for shape understanding related applications.

In the following chapters, we show the efficiency of enhanced topological skeleton based 3D shape modeling in three research problems related to Computer Graphics and Computer Vision. In particular, we underline how both of the two contributions presented in this chapter improve the state-of-the-art resolutions of these problems.

5.5 FRENCH CHAPTER ABSTRACT

Dans ce chapitre, nous avons exploré l'utilisation de la topologie différentielle pour la modélisation intrinsèque et expressive de formes 3D. En particulier, nous avons proposé de représenter les surfaces triangulées en entrée par une version améliorée du graphe de Reeb discret d'une fonction de Morse simple linéaire par morceaux f basée sur les distances géodésiques aux sommets caractéristiques. Nous avons introduit deux contributions principales.

Tout d'abord, nous avons présenté un algorithme pour l'extraction des sommets caractéristiques d'une surface triangulée (sommets situés aux extrémités des protubérances de la forme). Cet algorithme est rapide et stable et calcul un ensemble intrinsèque de sommets caractéristiques.

Ensuite, nous avons développé une définition formelle des graphes de Reeb dans le discret et nous avons proposé un algorithme pour leur calcul qui fournit en sortie des structures de données plus compatibles avec la définition des graphes de Reeb dans le continu que les approches précédentes. Cette version avancée des graphes de Reeb discrets est désigné par le terme de *squelette topologique avancé*.

Des expériences ont montré que le squelette topologique avancé de la fonction f est une représentation intrinsèque de la surface triangulée qui est à la fois stable et expressive. Par conséquent, il s'agit d'un outil intéressant pour les applications liées à la compréhension de forme.

Dans les chapitres suivants, nous montrons l'efficacité de la modélisation de forme 3D par squelette topologique avancé dans trois problèmes de recherche liés à l'Informatique Graphique et à la Vision par Ordinateur. En particulier, nous mettons en évidence comment chacun des apports scientifiques présentés dans ce chapitre contribue à améliorer les solutions de l'état de l'art à ces problèmes.

6

APPLICATION TO CONTROL SKELETON EXTRACTION

CONTENTS

6.1	CONTROL SKELETON PARADIGM	131
6.1.1	Motivations	131
6.1.2	Related work	132
6.2	AUTOMATIC CONTROL SKELETON EXTRACTION BY CONTOUR CONTRACTION AND CONTOUR CONSTRICTION COMPUTATION	133
6.2.1	Rigid segment definition by contour contraction	133
6.2.2	Joint definition by contour constriction computation	135
6.3	EXPERIMENTS AND RESULTS	138
6.3.1	Parameter setting	140
6.3.2	Computation times	141
6.3.3	Robustness	142
6.3.4	Discussion	143
6.4	APPLICATION TO SHAPE POSE EDITING	144
6.5	CHAPTER CONCLUSION	145
6.6	FRENCH CHAPTER ABSTRACT	145

THIS chapter introduces the most evident application of enhanced topological skeletons, which is control skeleton automatic extraction for 3D shape pose edition.

First, control skeleton segments are extracted by contour contraction. Then we introduce geometrical calculus on the Reeb graph (thanks to its enhanced version introduced in the previous chapter) for the computation of contour constrictions for plausible articulation detection.

Experiments show that the proposed algorithm is fast in practice and robust to several surface degradations. Finally, an application to 3D shape pose edition is proposed to underline the validity and the applicative interest of the approach.



Figure 6.1 – *Control skeleton driven shape pose editing of a 3D character* (Sébastien Domine, NVIDIA).

6.1 CONTROL SKELETON PARADIGM

6.1.1 Motivations

In several Computer Graphics applications, an intuitive way for shape handling is the *control skeleton*. Such a skeletal representation of the shape benefits to collision detection, shape editing or shape pose editing for animation.

In particular, the control skeleton driven shape pose edition is a paradigm widely used in 3D modeling systems, mostly for 3D animation in motion pictures or video games.

In this context, given a 3D shape represented by a surface mesh, the designer first constructs a hierarchy of rigid segments linked together with joints (the control skeleton) and position it inside the region of the Euclidean 3D space delimited by the surface mesh. Then, the designer notices the system of the association between the segments of the control skeleton and the vertices of the surface mesh. Finally, the pose of the 3D shape can be controlled by articulating the control skeleton, as illustrated in figure 6.1.

However, control skeleton creation is a time-consuming task, even for experienced users. Indeed, as mentioned by Aujay et al. (AHLD07), professional artists may create an initial skeleton relatively quickly but then need many adjustments after animation testing to make the joints exactly match the articulations of the object. Consequently, full or even partial automatic control skeleton reconstruction from surface meshes is a feature that greatly helps designers in their task. For this reason, a significant research effort has been provided by the 3D shape modeling community towards automatic control skeleton extraction from surface meshes.

6.1.2 Related work

According to Wu et al. (WML^{*06), a control skeleton should match certain expected properties.}

- **Simplicity:** the output control skeleton should be concise.
- **Stability:** the control skeleton should only slightly vary if the input shape slightly varies. Moreover, the control skeleton should be identical whatever the pose of the input shape is.
- **Meaningfulness:** the control skeleton should only depict the most important features of the shape and should not encode noisy geometrical details.
- **Neutrality:** the control skeleton should be located inside the shape so the surface can be articulated in any direction.
- **Hierarchy:** the control skeleton should be a hierarchy of connected elements (segments linked by joints).

Curve skeleton extraction techniques (MWLo2, MWOO3, CSYB05, DJo6, WML^{*06, CSM07, RT07, SLS07, YBS07, ATC^{*08, WLo8) are interesting candidates for automatic control skeleton extraction and their relative performance can be evaluated with regard to the above qualitative criteria as well as with regard to their computational effort requirement. We defer the reader to section 3.3.1 of chapter 3 (page 45) for a more detailed survey about curve skeleton extraction techniques.}}

However, most of these methods only consider joint locations in the configurations of branching of the skeleton (bifurcations and junctions). Consequently, these approaches fail at identifying joints in configurations of constant topology, like the elbow of the arm of a humanoid character since the upper arm and the forearm form a unique tubular component.

A common property of deformable 3D characters is that the joints that are not located in the configurations of branching of the skeleton are located in the narrowest portions of the surface, due to the modeling of muscles. For example, in figure 6.1, the knees of the character are narrow regions of the legs and the elbows and the wrists are also narrow regions of the arms.

Based on this hypothesis, Hétroy and Attali suggest the possibility of articulation detection through *constriction* computation (HA03a). The constrictions of a surface are closed curves embedded on it which minimize

their length. Hétroy also showed experimentally (H05) that constriction location depends on the surface curvature and that consequently constriction detection could be approximated through curvature analysis.

In this chapter, we present the most evident application of enhanced topological skeletons, which is control skeleton extraction. In particular, we benefit from the contour-based representation of the surface provided by enhanced topological skeleton to define geometrical calculus on the Reeb graph. First, the segments of the skeleton are defined by contour contraction. Second, by sweeping the ordered set of contours recorded in the edges of the enhanced topological skeleton, we isolate the contours that minimize a geometrical measure to identify contour constrictions, and thus plausible articulations. To our knowledge, this is the first approach which combines the analysis of the topological **and** geometrical evolution of the level sets of a piecewise linear function for shape characterization. Such an approach is possible thanks to the enhanced version of Reeb graphs provided by the enhanced topological skeleton computation algorithm.

6.2 AUTOMATIC CONTROL SKELETON EXTRACTION BY CONTOUR CONTRACTION AND CONTOUR CONSTRICTION COMPUTATION

In this section we present a technique for automatic control skeleton extraction from triangulated surfaces. First, the enhanced topological skeleton with regard to the PL Morse function to feature vertices f (defined in section 5.2, page 96) is computed.

The rigid segments of the control skeleton are then defined and positioned inside the 3D shape by contour contraction. At this stage, the joints between the segments only correspond to the configurations where the critical vertices of f are located.

Then, plausible additional joints, located on areas of constant topology, are identified by discriminating constrictions among the collection of contours stored in the enhanced topological skeleton.

6.2.1 Rigid segment definition by contour contraction

To constrain the position of the control skeleton segments inside the 3D shape while still efficiently keeping track of the geometry of the triangulated surface, we use the following strategy.

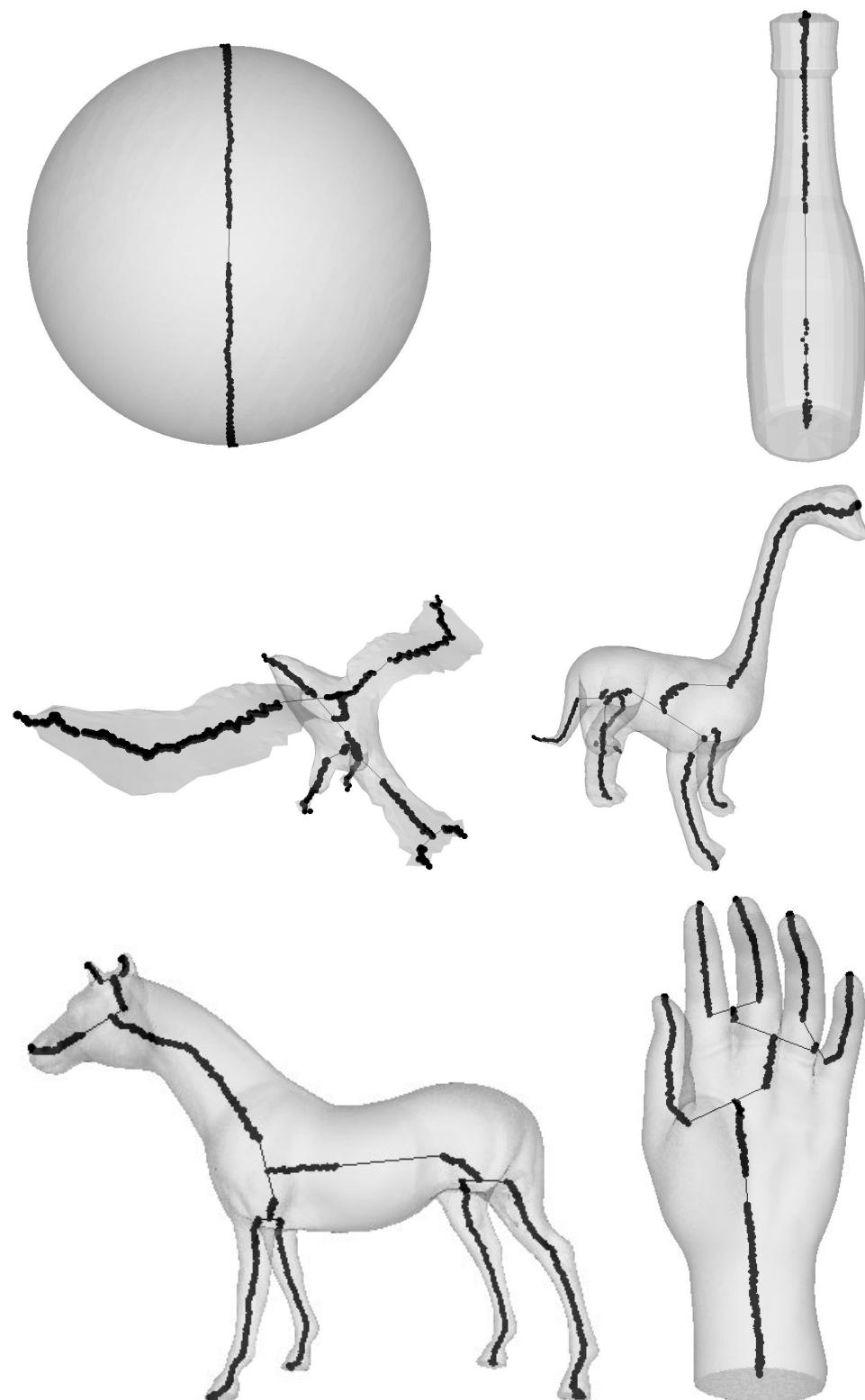


Figure 6.2 – Control skeleton segment positioning by contour contraction.

We propose to define each point of the segments relative to the equivalence classes of the Reeb graph $\mathcal{R}(f)$ of the input triangulated surface \mathcal{S} . In particular, each equivalence class of the Reeb graph is contracted to a point and embedded at the barycenter of its corresponding discrete contour $(f^{-1}(f(v)))_v$, as shown in figure 6.2.

In this figure, we can observe that the segments of the skeleton are indeed inside the object and moreover they follow the global geometry of the shape. Furthermore, notice that the degree of equivalence classes corresponding to regular vertices equals two, that of classes corresponding to minimum and maximum vertices equals one and that of classes corresponding to 1-saddle vertices equals three, correspondingly to the properties of the Reeb graph in the smooth setting (property 8 page 78). In particular, equivalence classes of 1-saddle vertices correspond to bifurcations and junctions in the control skeleton and can be used as joints between protrusions.

Notice that some authors (LV99, PSBM07) propose a similar embedding strategy by partitioning f domain and contracting the connected components of the pre-images of the resulting intervals. Consequently, the fidelity of the embedded skeleton to the geometry of the surface depends on the definition of the intervals of f base domain. If these intervals are too large, the resulting skeleton might miss some geometrical features of the shape and might get out the 3D shape. Here, we overcome this problem thanks to the enhanced version of Reeb graphs provided by enhanced topological skeleton. Indeed, it provides the finest possible contour-based representation, since the N_V equivalence classes of the graph are each equipped with their corresponding contour (N_V is the number of vertices in \mathcal{S}).

6.2.2 Joint definition by contour constriction computation

To identify joints located on plausible articulations, we exploit the hypothesis that due to the modeling of the muscles of the input characters, the constrictions of the surface should match some of their articulations, as suggested in (H05). In particular, the constrictions of a surface are closely related to its curvature (H05). Consequently, we propose to isolate among the contour collection of the enhanced topological skeleton those which minimize a geometrical measurement based on curvature analysis.



Figure 6.3 – Several curvature measurements on the hand triangulated surface: discrete mean curvature κ_H (left), discrete Gaussian curvature κ_G (middle) and curvature index κ_I (KvD92) (right).

Curvature measurement

On the contrary to the mean curvature, the Gaussian curvature is intrinsic to the surface and does not depend on its embedding in the Euclidean 3D space. Consequently, it is an interesting measure for a robust extraction of the control skeleton joints (which does not depend on the input character embedding). However, Gaussian curvature is known to be noisy in the discrete setting. For this reason, Koenderink et al. (KvD92) introduced the curvature index, whose expression is the following (we defer the reader to chapter 4 page 83 for the expressions of $\kappa_H(v)$ and $\kappa_G(v)$):

$$\kappa_I(v) = \begin{cases} \frac{2}{\pi} \arctan \frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2} & \kappa_1 \neq \kappa_2 \\ 0 & \kappa_1 = \kappa_2 \end{cases} \quad (6.1)$$

with:

$$\kappa_1(v) = \kappa_H(v) + \sqrt{\kappa_H^2(v) - \kappa_G(v)} \quad (6.2)$$

$$\kappa_2(v) = \kappa_H(v) - \sqrt{\kappa_H^2(v) - \kappa_G(v)} \quad (6.3)$$

Figure 6.3 shows with a color gradient the computation of the discrete mean curvature (left), of the discrete Gaussian curvature (middle) and of the curvature index (right). This figure shows that the curvature index is a more smooth measurement than the Gaussian curvature. Moreover, it gives better contrast between the concave (dark color) and convex (light color) parts of the triangulated surface. For this reason, the curvature index will be used for contour constriction computation.

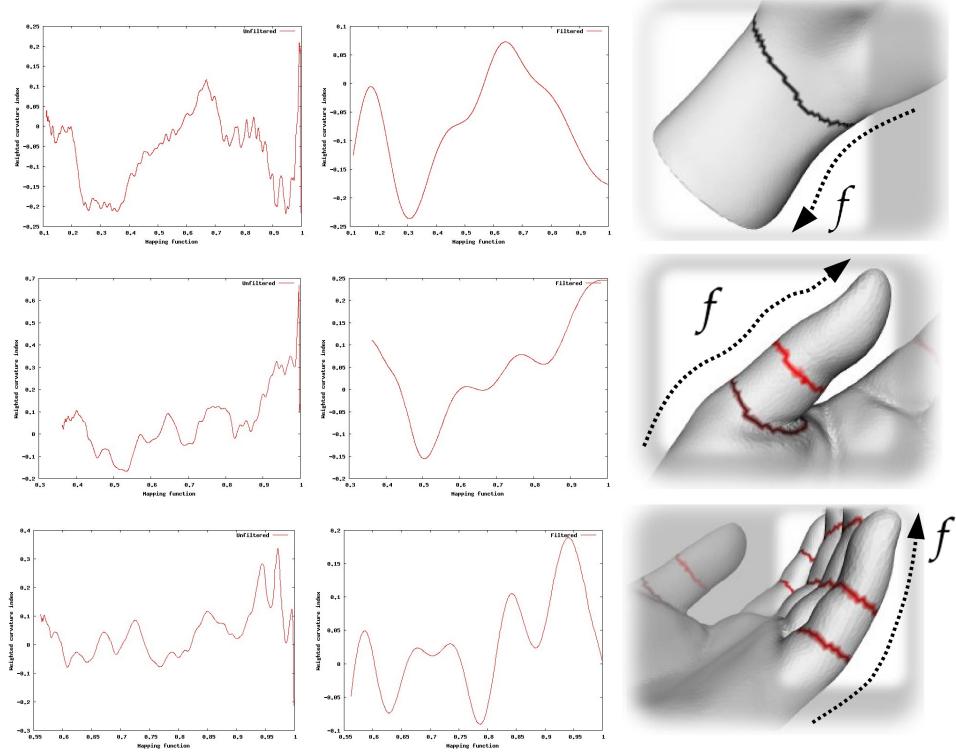


Figure 6.4 – g (left) and \hat{g} (middle) with regard to f (X axis) for several edges of $\mathcal{R}(f)$ and the corresponding surface regions (right).

Contour geometry measurement

As suggested by Hétry (Hé5), constrictions appear on the most concave regions of a surface. Consequently we propose to detect contour constrictions by computing on the equivalence classes $(f^{-1}(f(v)))_v$ of the Reeb graph a geometrical measurement function based on the curvature index, such that $g : \mathcal{R}(f) \rightarrow \mathbb{R}$:

$$g((f^{-1}(f(v)))_v) = \frac{\sum_{v_i \in (f^{-1}(f(v)))_v} \kappa_I(v_i) \cdot (\mathcal{L}_{e_1}(v_i) + \mathcal{L}_{e_2}(v_i))}{2 \cdot \mathcal{P}((f^{-1}(f(v)))_v)} \quad (6.4)$$

where $\mathcal{L}_{e_1}(v_i)$ and $\mathcal{L}_{e_2}(v_i)$ are the lengths of the two adjacent edges of v_i on the contour $(f^{-1}(f(v)))_v$ and where $\mathcal{P}((f^{-1}(f(v)))_v)$ is the perimeter of the contour $(f^{-1}(f(v)))_v$. Such a weighted average of the curvature index enables a curvature estimation around the contour which is independent of the sampling of the contours (determined by the sampling of \mathcal{S}).

Figure 6.4 (left) shows the evolution of g with regard to f on three edges of $\mathcal{R}(f)$ (see the definition of edges page 116). Each edge stores a collection of contours of equivalence classes of regular vertices which is sorted by increasing f values. Thus, it is straightforward to compute g relative to f .

Despite the use of the curvature index, the function g is still noisy on $\mathcal{R}(f)$ (see figure 6.4, left column). To remove high frequency noise from g , we apply an ideal low-pass filter of cutoff frequency f_τ , defined by the following transfer function:

$$\mathcal{H}(f) = \begin{cases} 1 & f \leq f_\tau \\ 0 & f > f_\tau \end{cases} \quad (6.5)$$

Then, we considered the filtered version of g , noted $\hat{g} : \mathcal{R}(f) \rightarrow \mathbb{R}$:

$$\hat{g}(\mathcal{R}(f)) := FT^{-1}(\mathcal{H}(f) \cdot FT(g(\mathcal{R}(f)))) \quad (6.6)$$

where FT stands for the Fourier Transform. Notice that the definition of geometrical calculus on $\mathcal{R}(f)$ is possible thanks to the enhanced contour-based version of Reeb graphs introduced in the previous chapter.

Figure 6.4 (middle) represents the evolution of \hat{g} with regard to f on three edges of $\mathcal{R}(f)$ and shows low-pass filtering smooths g and enables a clear distinction of g extremum contours.

Contour constriction computation

If \hat{g} is positive, then the surface around the corresponding contour is globally convex, otherwise it is concave. As constrictions are located on the most concave regions of the surface (Hó5), we focus on the negative minimum contours of \hat{g} to identify contour constrictions.

In figure 6.4, \hat{g} has one negative minimum contour on the wrist of the hand triangulated surface (top row, middle), two on the thumb (second row, middle) and two on the middle finger (bottom row, middle). Corresponding minimum contours of \hat{g} have been displayed in red on the triangulated surface (figure 6.4, right). Moreover, to avoid the identification of several minimum contours in the configuration of a unique surface constriction, we add the constraint that two consecutive contour minimizing \hat{g} must have a difference of f value greater than a threshold Δ .

Finally, the contours that minimize \hat{g} are identified as *contour constrictions* and are reported as special joints in the control skeleton as shown in figure 6.5 (right) where they have been reported with red points.

6.3 EXPERIMENTS AND RESULTS

Figure 6.5 shows some examples of control skeletons after contour contraction and contour constriction computation (right column). The rigid

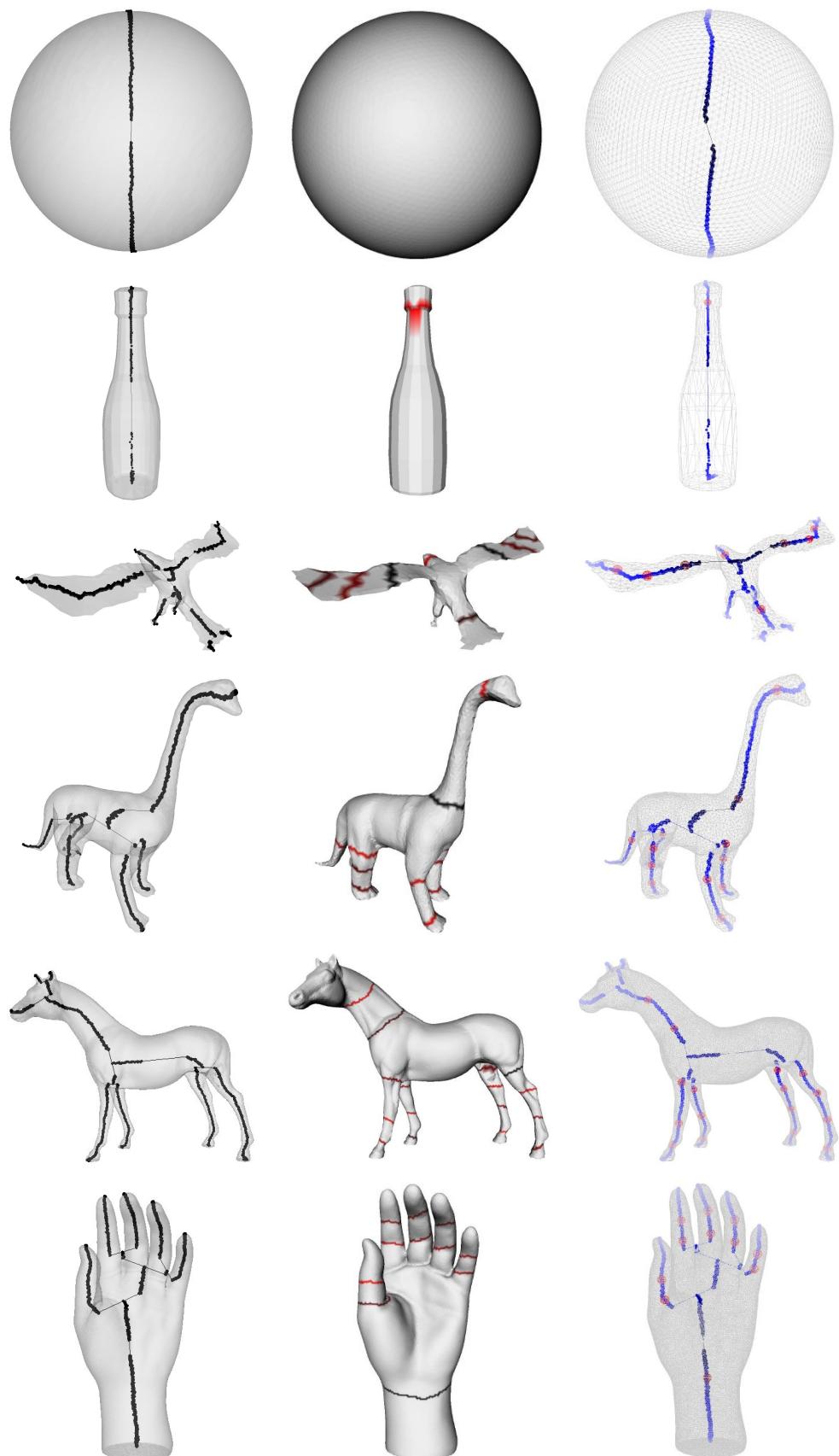


Figure 6.5 – Control skeleton after contour contraction (left), contour constrictions (middle) and control skeleton after contour constriction computation (right).

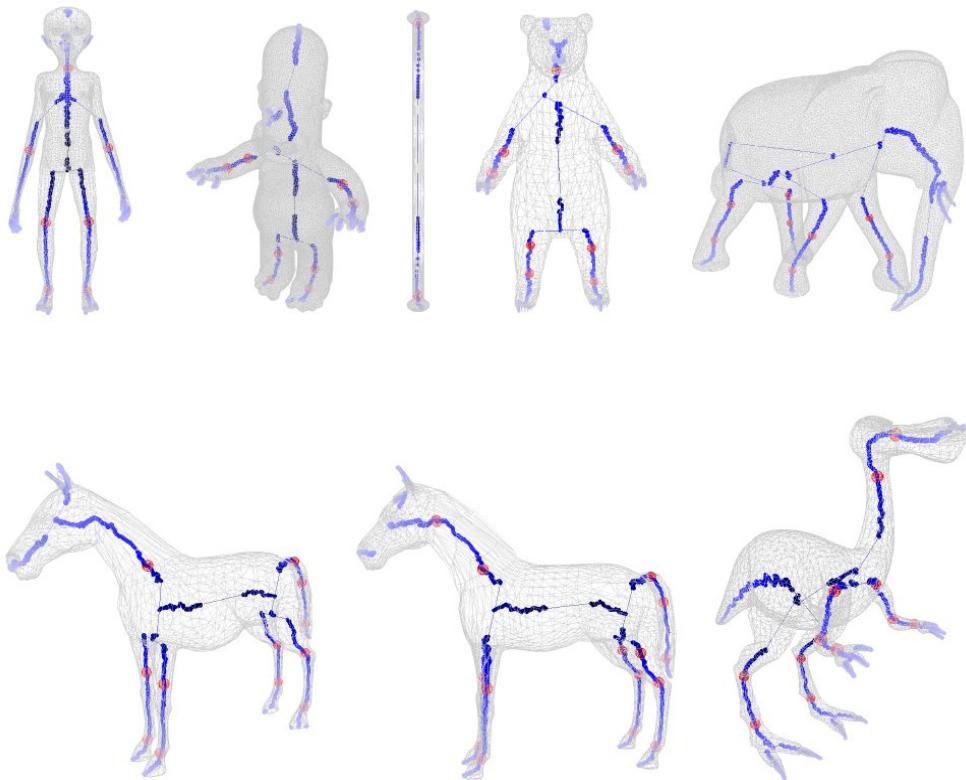


Figure 6.6 – *Control skeleton of other triangulated surfaces.*

segments of the skeleton are reported in blue and special joints corresponding to contour constrictions in red. Figure 6.6 shows the resulting control skeletons of other triangulated surfaces.

In this section, we present further experimental results, including computation times and discussion elements.

6.3.1 Parameter setting

Two parameters intervene in the computation of control skeleton from the enhanced topological skeleton.

First, the parameter f_τ determines the cut-off frequency for the low-pass filtering of the geometrical measurement used for contour constriction computation. It prevents the analysis of a noisy geometrical measurement and a sensible computation of contour constriction. Table 6.1 shows the effect of f_τ variation on the number of computed contour constrictions. The default value for all experiments is $f_\tau = 8$ (bold column).

Second, the parameter Δ constraints consecutive contour constrictions to have a minimal difference of f values. Table 6.2 shows the effect of Δ variation on the number of computed contour constrictions. Notice this

Triangulated surface	$f_\tau = 1$	$f_\tau = 5$	$f_\tau = 8$	$f_\tau = 10$	$f_\tau = 20$
Sphere	0	0	0	0	0
Bottle	0	1	1	2	2
Bird	0	8	8	8	8
Dinosaur	0	12	13	15	13
Horse	0	12	16	14	13
Hand	0	10	11	13	13

Table 6.1 – Number of contour constrictions of $\mathcal{R}(f)$ with several f_τ cut-off frequency values (default value: $f_\tau = 8$).

Triangulated surface	$\Delta = 0$	$\Delta = 0.05$	$\Delta = 0.1$	$\Delta = 0.2$
Sphere	0	0	0	0
Bottle	1	1	1	1
Bird	8	8	8	8
Dinosaur	17	17	13	10
Horse	18	18	16	10
Hand	11	11	11	6

Table 6.2 – Number of contour constrictions of $\mathcal{R}(f)$ with several Δ parameter values (default value: $\Delta = 0.1$).

parameter has a low influence on the result but prevents constriction over-detection on certain triangulated surfaces. The default setting for all the experiments is $\Delta = 0.1$ (bold column).

6.3.2 Computation times

Once the enhanced topological skeleton of the input triangulated surface is computed with regard to f , contour contraction depends on the number of equivalence classes in $\mathcal{R}(f)$ and on the number of vertices in the discrete contours. Consequently, this operation requires $O(N_V^2)$ steps in the worst case, with N_V the number of vertices in \mathcal{S} .

The computational effort related to the geometrical measurement on the contours of $\mathcal{R}(f)$ also depends on the number of vertices in the contours. As g is evaluated on each contour, its computation requires $O(N_V^2)$ steps. Finally, low-pass smoothing on g is implemented with the fast fourier transform algorithm (CT65) and requires $O(N_V \log(N_V))$ steps.

Consequently, the overall time complexity of control skeleton computation using enhanced topological skeleton is bounded by $O(N_V^2)$ in the

Triangulated surface	Triangles	Computation time (s.)
Bottle	512	0.045
Bird	1996	0.425
Sphere	8000	6.342
Dinosaur	9989	3.490
Horse	39996	33.793
Hand	52000	109.194

Table 6.3 – Overall computation times for control skeleton extraction.

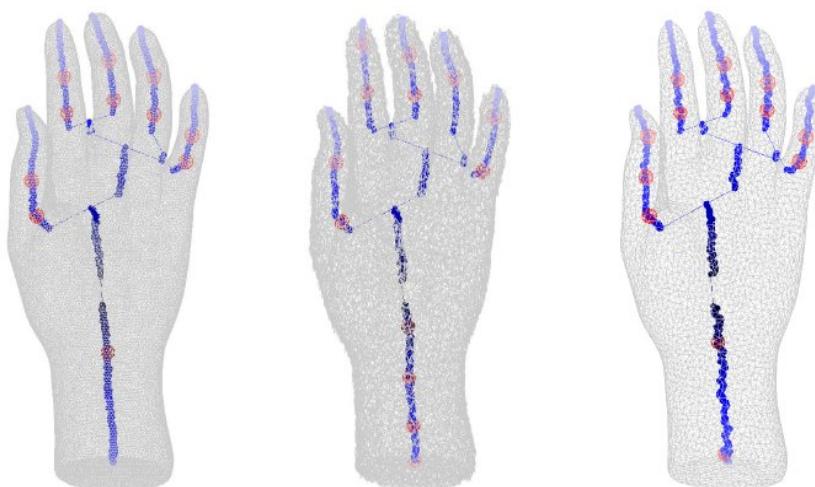


Figure 6.7 – Control skeleton robustness towards triangulated surface degradations: random noise (middle) and under-sampling (right).

worst case. Table 6.3 provides examples of overall computation times with the experimental setup described in section 5.3.5 page 120.

6.3.3 Robustness

Control skeletons extracted from the enhanced topological skeletons computed with regard to the f PL Morse function to feature vertices (defined in section 5.2, page 96) benefit from the invariance and robustness properties of f .

Thus, control skeletons are invariant towards affine and isometric transformations (see experiments reported in figure 5.19 page 125). However, curvature index computation is quite noise sensitive. Figure 6.7 shows the computation of control skeletons after random noise (where vertices have been moved randomly with an amplitude of $\pm 1\%$ of the lengths of the bounding box of the shape) and under-sampling (where the number of vertices in the triangulated surface has been divided by five).

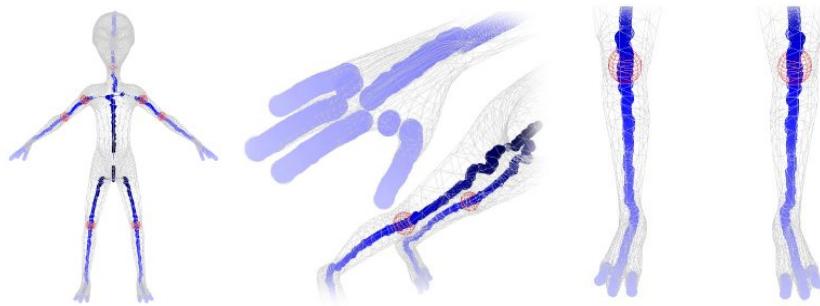


Figure 6.8 – Zooming in the alien triangulated surface: feature vertex extraction implies the creation of segments in the control skeleton for each finger and toe.

This figure shows that the computation of contour constriction slightly varies after these degradations (see the related joints in red).

6.3.4 Discussion

Figures 6.5 and 6.6 show that the control skeletons computed from the enhanced topological skeletons respect the qualitative expectations enumerated by Wu et al. (WML*06):

- **Simplicity:** Output control skeletons are sets of connected points (more concise than medial surfaces for example).
- **Stability:** Output control skeletons benefit from the geometrical invariance properties of the PL Morse function f and benefit from the topological invariance properties of $\mathcal{R}(f)$. Moreover, they are only slightly sensible to surface degradations like random noise or under-sampling.
- **Meaningfulness:** Thanks to the feature vertex extraction, only the most important geometrical and perceptual shape features are depicted in the output control skeletons, as shown in figure 6.8.
- **Neutrality:** thanks to the contour-based modeling provided by enhanced topological skeletons, segments are guaranteed to be positioned inside the 3D shape.
- **Hierarchy:** Output control skeletons are connected and the segments can be ordered with regard to f .

In contrast with standard curve skeleton extraction techniques, our method has the advantage to also extract additional plausible joints located on surface constrictions, which is an interesting feature for character

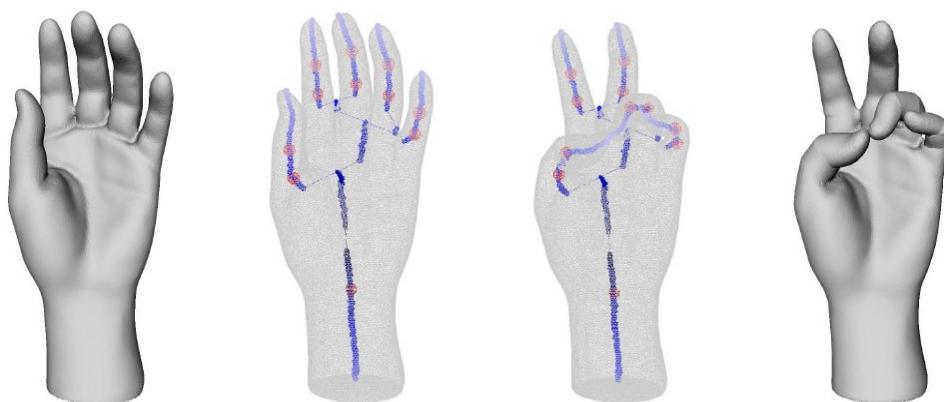


Figure 6.9 – Control skeleton of the hand triangulated surface (middle left), articulated skeleton (middle right) and resulting pose edition (right).

articulation. As shown in figures 6.5 and 6.6 contour constrictions indeed correspond to plausible articulations, like knees, elbows or phalanx joints.

Finally, control skeletons are automatically extracted with satisfactory computation times with regard to state-of-the-arts techniques (for comparison, in practice, our algorithm runs about twenty times faster than that presented in (WML*06) for equivalently sampled triangulated surfaces).

6.4 APPLICATION TO SHAPE POSE EDITING

Figure 6.9 demonstrates the applicative interest of automatic control skeleton extraction for character animation.

First, the enhanced topological skeleton with regard to f is computed for the input triangulated surface and the control skeleton is extracted (figure 6.9, middle left).

Then the user simply grabs the segments of the control skeleton and apply rotations to them, relatively to the joints (in red in figure 6.9, middle right).

The enhanced topological records a full correspondence between the points of the segments (contracted equivalence classes) and the vertices of the related contours. Consequently, the segment rotations can be applied to the implied vertices directly to articulate the shape (figure 6.9, right). Notice moreover that the same procedure can be applied if the user wants to apply a given pose to several other triangulated surfaces (automatic shape pose transfer). Furthermore, shape pose modifications can be concisely encoded through a set of parametric rotations, enabling automatic animation.

As a conclusion, the control skeleton serves as a powerful shape handling abstraction, which enables 3D shape pose editing in just a few user interactions.

6.5 CHAPTER CONCLUSION

In this chapter, we introduced some geometrical calculus tools on the discrete Reeb graph $\mathcal{R}(f)$ for control skeleton driven 3D shape pose edition. Thanks to the enhanced contour-based version of Reeb graphs introduced in the previous chapter, fine shape geometry characterization is possible. In particular, we proposed to model control skeleton segments by contour contraction, so the output skeleton naturally captures the geometry of the input shape. Moreover, we introduced geometrical calculus on the Reeb graph to isolate contours of minimum curvature index, corresponding to plausible shape articulations.

The output control skeletons benefit from the geometrical invariance properties of the enhanced topological skeleton computed relatively to the PL Morse function to feature vertices (affine and isometric invariance).

Experiments showed that the control skeletons computed with this technique are quite insensible to triangulated surface degradations like random noise or under-sampling. Moreover, skeleton extraction is achieved in satisfactory computation times, which are lower than representative state-of-the-art techniques.

Finally, we demonstrated the applicative interest and the expressive properties of extracted skeletons in a 3D shape pose editing application, directly usable with most of actual 3D modeling software systems.

6.6 FRENCH CHAPTER ABSTRACT

Dans ce chapitre, nous avons introduit la notion de calcul géométrique sur le graphe de Reeb discret $\mathcal{R}(f)$ pour l'édition de pose d'objets 3D déformables dirigée par squelette de contrôle. Grâce à la version enrichies de contours des graphes de Reeb introduite dans le chapitre précédent, une caractérisation fine de la géométrie de la surface est rendue possible. En particulier, nous avons proposé de modéliser les segments du squelette de contrôle par contraction de contour, pour que le squelette de sortie capture naturellement la géométrie de la forme. De plus, nous avons introduit la notion de calcul géométrique sur le graphe de Reeb pour isoler les con-

tours d'index de courbure minimum, correspondant à des articulations plausibles de la forme.

Les squelettes de contrôle calculés bénéficient de l'invariance géométrique des squelettes topologiques avancés calculé relativement à la fonction de Morse aux sommets caractéristiques (invariance affine et isométrique).

Des expériences ont montré que les squelettes de contrôle calculés avec cette technique sont relativement insensible aux dégradations de surface comme le bruit aléatoire ou le sous échantillonnage. Par ailleurs, l'extraction de squelette s'effectue dans des temps d'exécution satisfaisants, inférieurs à ceux de méthodes représentatives de l'état de l'art.

Enfin, nous avons montré l'intérêt applicatif et l'expressivité des squelettes extraits dans une application d'édition de pose d'objets 3D déformables, directement utilisable avec la plupart des logiciels de modélisation 3D actuels.

7

APPLICATION TO PARTIAL SHAPE RETRIEVAL

CONTENTS

7.1	SCIENTIFIC CONTEXT	149
7.1.1	Motivations	149
7.1.2	Related work	150
7.1.3	Problem statement and method overview	151
7.2	REEB CHART UNFOLDING BASED GEOMETRY SIGNATURE	154
7.2.1	Reeb chart topology	154
7.2.2	Reeb chart unfolding	157
7.3	REEB PATTERNS IN REEB GRAPHS	161
7.3.1	Reeb patterns and structural distortion	161
7.3.2	Reeb pattern topology	163
7.4	PARTIAL SIMILARITY COMPUTATION BASED ON REEB PATTERN UNFOLDING	167
7.4.1	Reeb pattern combination enumeration	168
7.4.2	Expansion of common sub-graphs	168
7.4.3	Intra-Reeb pattern node-to-node matching	170
7.4.4	Partial similarity estimation	170
7.5	EXPERIMENTAL STUDY	171
7.5.1	SHREC 2007 partial shape retrieval benchmark	172
7.5.2	Experimental setup	174
7.5.3	Performance evaluation	176
7.5.4	Robustness evaluation	178
7.5.5	Discussion	179
7.6	APPLICATION TO MODELING BY EXAMPLE	180

7.7 CHAPTER CONCLUSION	180
7.8 FRENCH CHAPTER ABSTRACT	182

THIS chapter introduces an application of enhanced topological skeleton based 3D shape modeling to the design of a partial 3D shape retrieval system. This challenging application enables to provide a quantitative measurement of the efficiency of the shape modeling approach.

In this context, given a query 3D shape, the system is expected to retrieve, in interactive processing time, the entries of a 3D shape collection which share visually similar sub-parts with the input query, even if they visually differ globally.

In the proposed approach, each triangulated surface is represented by a concise version of the enhanced topological skeleton of the PL Morse function to feature vertices. Resulting graphs are enriched with geometry signatures based on parameterization techniques. Then partial similarity between shapes is evaluated by computing a variant of their maximum common sub-graphs.

By investigating Reeb graph theory, we both improve the 3D shape description and comparison steps. In particular, we introduce the new notions of *Reeb charts* and *Reeb patterns* on Reeb graphs and provide theoretical results on their topological properties. This topology control enables the definition of intrinsic geometry signatures based on surface parameterization techniques and it also enables the definition of a sub-graph matching technique which is robust to Reeb graph specific distortion.

Extensive experiments show the superiority of the approach on the SHREC 2007 partial shape retrieval benchmark. To further asses the efficiency of our method, a full implementation is freely available on the Web¹.

¹ <http://www.lifl.fr/~tierny/sinamis.html>

7.1 SCIENTIFIC CONTEXT

7.1.1 Motivations

Visual similarity based 3D shape retrieval systems aim at helping human users browsing large collections of 3D shapes in an interactive and intuitive way. In this framework, the user specifies a 3D shape as an example query and the system is expected to sort automatically the entries of the collection by decreasing visual similarity, providing as top results the most similar entries.

An important amount of literature has been provided for global similarity based shape retrieval, presenting methods enabling the retrieval of similar objects despite affine transformations (CTSOo3, FMK^{*}03, RCB05, RLBo8) or even isometric transformations (such as shape bending or character articulation) (HSKK01, TS05, GSCO07, JZ07). We defer the reader to survey articles (TV04, BKS^{*}05, IJL^{*}05) for a broad overview of global similarity based retrieval methods.

However, certain applicative contexts, such as modeling-by-example (FKS^{*}04) (where new shapes are created by cutting and pasting existing shape sub-parts) or registration (HKDH04), might require to retrieve objects with regard to *partial* similarity. In such a paradigm, systems are expected to retrieve objects that have *similar sub-parts* even if they visually differ globally.

Unlike global similarity based shape retrieval, only few papers have addressed the partial 3D shape retrieval problem, while it is seen as the next challenging open issue by the shape retrieval community (CTSOo3, FMK^{*}03). Moreover, it is a more general problem than global shape similarity estimation (two globally similar shapes will also be similar partially) and thus it is a research topic of larger impact.

The design of a partial similarity based 3D shape retrieval system has to bring solutions to two sub-problems, both addressed in this chapter:

1. Shape signature computation:

- Each entry of the 3D shape collection must be represented by a concise *shape signature* (a compact shape representation) so as to speed up the similarity estimation process to guarantee interactive querying.
- Shape signatures must capture local shape features so as the shape comparison process can match shape sub-parts and evaluate partial similarity.

- Shape signatures must also be invariant to shape-preserving transformations and robust to surface degradations, so as to retrieve partially similar objects modulo these transformations.

2. Shape signature comparison:

- An algorithm for shape signature comparison must also be proposed. It must integrate the matching of shape sub-parts and must provide a similarity score for each partial match.
- The comparison algorithm must run with satisfactory computational effort to guarantee interactive querying.

7.1.2 Related work

Partial shape retrieval state-of-the-art techniques can be roughly classified into two categories.

On the first hand, *local descriptors* based techniques aim at characterizing the local properties of a large number of small features extracted on the shape. Then, partial shape similarity is estimated by feature point-to-point matching and matching similarity estimation. Liu et al. (LZQ06) propose to use a Monte-Carlo sampling on the surface mesh and to capture the local aspect of the shape with spin-image signatures (JH99). Funkhouser and Shilane (FS06) present a more sophisticated sampling strategy and then describe local geometry with several descriptors based on Spherical Harmonics (FMK^{*}03) but this method is only experimented for global shape retrieval. In order to deal with the combinatorial explosion due to individual local feature comparison, such approaches use complex data-structures to drive the matching process, such as priority queues (FS06) or feature clusters (LZQ06) inspired by text document analysis. Gal et al. (GCO06) present an interesting geometrical hashing mechanism associated with a local surface description based on curvature analysis. However, as underlined by Biasotti et al. (BMSF06), most of local descriptor methods base their partial similarity estimation on point-to-point matching only. This is particularly detrimental in term of re-usability in applicative contexts such as modeling-by-example (FKS^{*}04) where the similar sub-parts have to be explicitly identified and extracted.

On the other hand, structural based approaches present the advantage to explicitly identify the surface patches that have been matched. Moreover, the combinatorial explosion due to feature comparison can be easily reduced by reasoning on the structure of the shape instead of using com-

plex hashing mechanisms. These methods first segment the shape and represent it by a graph depicting the structural relations between the segments. Then, partial shape similarity is estimated using graph matching techniques. Cornea et al. (CDS^{*}05) propose to extract the curve-skeleton of the shape and then employ the Earth Mover's distance (KSDD03) to evaluate the partial similarity of the skeletons. However, sub-part signature is based on the Euclidean distance between the surface and its skeleton which makes the method quite sensible to isometric transformations. Biasotti et al. (BMSF06) present an efficient method based on a derived version of multi-resolution Reeb graphs and Spherical Harmonics (FMK^{*}03). However, even if the Reeb graph computation is robust to isometric transformations, sub-parts signatures (spherical harmonics) are not, which is slightly detrimental to the overall robustness of the approach.

In this chapter, we present a fast and efficient structural oriented approach for partial 3D shape retrieval, based on the enhanced topological skeleton computed relative to the PL Morse function to feature vertices (defined in section 5.2, page 96). In particular, we take advantage of the intrinsic properties of Reeb graph theory to improve both the shape description and comparison process.

First, thanks to the feature vertex extraction, this shape representation is able to capture local shape features. For example, it captures the fingers of the hand of a humanoid 3D shape, permitting a partial comparison with a standalone hand 3D shape.

Second, thanks to the contour-based representation of the shape provided by enhanced topological skeletons and thanks to their topological properties, we segment the input triangulated surface into patches of controlled topology and propose an efficient sub-part geometry signature based on parameterization techniques (GV04, WML^{*}06, BCG08).

Finally, we also take advantage of the topological properties of the Reeb graph to propose a fast and efficient approximation of the maximally similar sub-graph computation problem.

7.1.3 Problem statement and method overview

Shape signature

First, the enhanced topological skeleton of the input triangulated surface \mathcal{S} (which is required here to be orientable, connected and closed) is computed relative to the PL Morse function f to feature vertices (defined in section 5.2, page 96). Then, \mathcal{S} is segmented at the equivalence classes of

the critical vertices of f into a set of *charts* \mathcal{C}_i of controlled topology called *Reeb charts*. We show that thanks to the Reeb graph topological properties, Reeb charts have either disk or annulus topology. This topology control enables efficient intrinsic sub-part geometry signature based on parameterization techniques (GV04, WML*06, BCG08). In particular, for each chart \mathcal{C}_i , we compute its *unfolding signature* λ_{ϕ_i} by measuring the distortion of its mapping ϕ_i to its canonical planar domain D_i (a surface with the same topology and having constant Gaussian curvature). Finally, the signature of the input triangulated surface \mathcal{S} consists of a concise version of its dual Reeb graph computed relatively to f , noted \mathcal{R} , associated with the unfolding signatures of the Reeb charts.

Shape signature partial similarity

Given two signatures \mathcal{R}_1 and \mathcal{R}_2 to compare, the goal of partial similarity estimation is to find an optimal injective mapping m^* between a sub-graph $\mathcal{G}_1^* \subset \mathcal{R}_1$ and a sub-graph $\mathcal{G}_2^* \subset \mathcal{R}_2$ which maximizes a similarity function $S(m^*)$, computed relatively to the unfolding signatures of the nodes of \mathcal{R}_1 and \mathcal{R}_2 matched by m^* .

A brute-force approach to this problem consists in computing the set \mathcal{M} of all the possible injective mappings $m : \mathcal{R}_1 \rightarrow \mathcal{R}_2$, then identifying through an expansion process the common sub-graphs $\mathcal{G}_1^m \subset \mathcal{R}_1$ and $\mathcal{G}_2^m \subset \mathcal{R}_2$ (such that $m : \mathcal{G}_1^m \rightarrow \mathcal{G}_2^m$ is an isomorphism) and finally evaluating their geometrical similarity $S(m)$. The solution of this problem would be the mapping $m^* \in \mathcal{M}$ that would maximize S . Let n and k be respectively the number of nodes of the largest and the smallest of the two dual Reeb graphs to compare. With the brute-force approach, $|\mathcal{M}| = \frac{n!}{(n-k)!}$, which results in an exponential time complexity evaluation process. Moreover, in practice, as underlined by Messmer and Bunke (MB98), graphs representing real world objects may be affected by noise or distortion, motivating the use of error-tolerant matching algorithm, whose complexity is generally even greater than exact algorithms (MB98).

To reduce the number of combinations to consider and to guarantee interactive querying, we introduce the notion of *Reeb pattern* on a Reeb graph along with its structural signature. We show the structural signature discards Reeb graph structural distortion and avoids the use of error-tolerant graph matching algorithms.

First, the sets $\mathcal{T}(\mathcal{P})_1$ and $\mathcal{T}(\mathcal{P})_2$ of topological equivalent *Reeb patterns* belonging respectively to \mathcal{R}_1 and \mathcal{R}_2 are detected. Then, the set \mathcal{M} of all

possible mappings m between topology equivalent *Reeb patterns* of $\mathcal{T}(\mathcal{P})_1$ and $\mathcal{T}(\mathcal{P})_2$ ($m : \mathcal{T}(\mathcal{P})_1 \rightarrow \mathcal{T}(\mathcal{P})_2$) is computed. In practice, the average cardinality of \mathcal{M} is twenty. Then, for each mapping m , an expansion algorithm simultaneously expands the common sub-graphs \mathcal{G}_1^m and \mathcal{G}_2^m in \mathcal{R}_1 and \mathcal{R}_2 .

Each mapping m is scored relatively to the geometrical similarity $S(m)$ between the two common sub-graphs \mathcal{G}_1^m and \mathcal{G}_2^m . In particular, $S(m)$ is computed by comparing the unfolding signatures λ_{ϕ_i} and λ_{ϕ_j} of the pairs of matched Reeb charts \mathcal{C}_i and \mathcal{C}_j .

Let \widehat{m}^* maximize on \mathcal{M} the similarity S between the related sub-graphs $\widehat{\mathcal{G}}_1^*$ and $\widehat{\mathcal{G}}_2^*$. The similar sub-parts of the two input triangulated surfaces \mathcal{S}_1 and \mathcal{S}_2 are identified by $\widehat{\mathcal{G}}_1^*$ and $\widehat{\mathcal{G}}_2^*$. Moreover, the partial similarity between \mathcal{S}_1 and \mathcal{S}_2 is given by $S(\widehat{m}^*)$. Algorithm 4 summarizes the overall process.

4: Partial 3D shape retrieval algorithm overview.

Data: a query triangulated surface \mathcal{S}_q ;

Data: a collection of 3D shape signatures C .

Result: the 3D shape signature collection C sorted by decreasing partial similarity.

begin

 Compute the Reeb graph $\mathcal{R}(f)$ of \mathcal{S}_q .

forall $\mathcal{C}_i \in \mathcal{R}(f)$ **do**

 | Compute \mathcal{C}_i unfolding signature λ_{ϕ_i} .

end

 /* The dual Reeb graph accompanied with the unfolding */

 /* signatures, noted \mathcal{R}_q is the shape signature of \mathcal{S}_q . */

 Compute \mathcal{R}_q structural signature.

forall $\mathcal{R}_c \in C$ **do**

 | Compute the set \mathcal{M} of mappings m between topology

 | equivalent Reeb patterns of \mathcal{R}_q and \mathcal{R}_c .

forall $m \in \mathcal{M}$ **do**

 | Expand recursively m to patterns' adjacent neighbor
 | charts.

 | Evaluate $S(m)$.

end

 | Return $S(\widehat{m}^*)$ as the partial similarity between \mathcal{R}_q and \mathcal{R}_c .

end

 Sort C by decreasing value of $S(\widehat{m}^*)$.

end

7.2 REEB CHART UNFOLDING BASED GEOMETRY SIGNATURE

First, the enhanced topological skeleton of the input orientable, connected and closed triangulated surface is computed with regard to the PL Morse function f to feature vertices (defined in section 5.2 page 96). In the framework of partial 3D shape similarity estimation, this function has been preferred to the integral geodesic distance (HSKK01) or the harmonic function (NGH04) because it gives better emphasis on local shape features (like fingers, or toes) thanks to the feature vertices, which is crucial for partial matching. Figure 7.1 shows that even the ears of the camel or the fingers of the humanoid 3D shapes are captured in the enhanced topological skeletons. Consequently, for example, the standalone hand triangulated surface will be able to be matched with the hands of these humanoids.

The dual Reeb graph of the input surface, whose nodes correspond to the edges of the Reeb graph and whose edges correspond to adjacency relations between them, constitute the core of the input surface's shape signature. To accompany this signature with intrinsic geometry characterization, we introduce the notion of *Reeb chart*.

7.2.1 Reeb chart topology

For the sake of intuition, in this chapter, we first introduce new notions in the smooth setting.

Definition 88 (Reeb chart) *Let $R(f)$ be the Reeb graph of a simple Morse function f defined on an orientable, compact, smooth, connected and closed 2-manifold \mathbb{M} . Let $\Psi : \mathbb{M} \rightarrow R(f)$ map each element $m \in \mathbb{M}$ to its equivalence class in $R(f)$. Let $E = \{E_1, E_2, \dots, E_i, \dots, E_n\}$ be the edges (maximally connected unions of equivalence classes containing only regular points of f) of the Reeb graph $R(f)$. $C_i = \Psi^{-1}(E_i)$ is defined as a Reeb chart.*

An analog definition of Reeb charts in the discrete setting, noted \mathcal{C}_i , can easily be extended from the above definition. Figure 7.2(b) shows the symbolic embedding (defined page 5.3.3) of the enhanced topological skeleton of the hand triangulated surface relatively to f . In particular, each edge \mathcal{E}_i has been collapsed to a colored node. Figures 7.2(c) and 7.2(d) show the segmentation of the hand triangulated surface into its Reeb charts. Basically, Reeb charts are the surface patches that correspond to the nodes of the dual Reeb graph.

Property 11 (Reeb chart topology) *Reeb charts of an orientable, compact, smooth, connected*

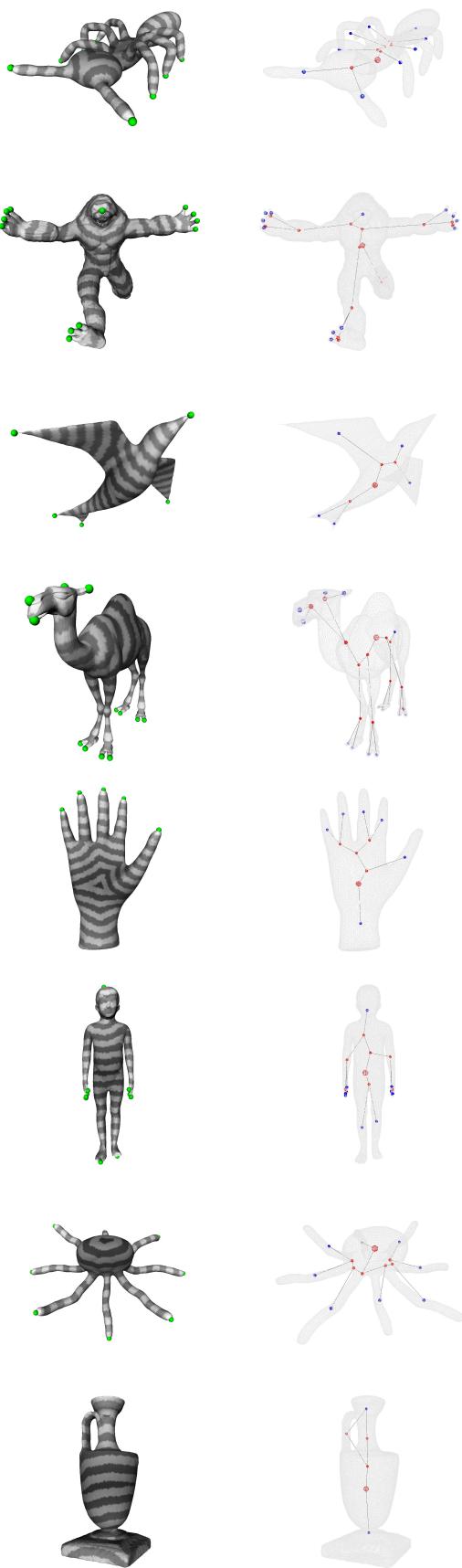


Figure 7.1 – Feature vertices and f functions (left) and symbolic embeddings of the related enhanced topological skeletons (right) of several triangulated surfaces.

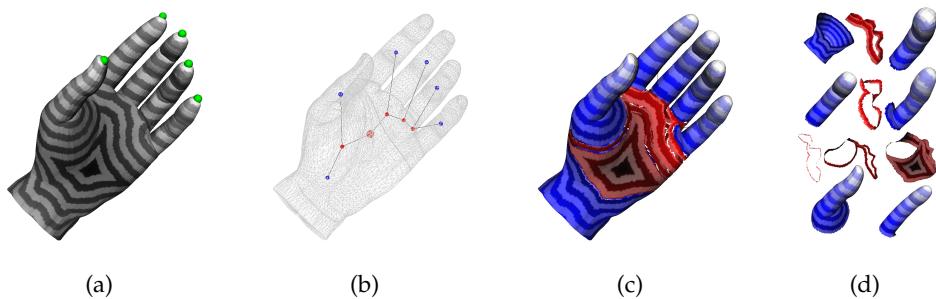


Figure 7.2 – Segmentation of a hand triangulated surface model into its Reeb charts.

and closed 2-manifold \mathbb{M} have either disk or annulus topology, whatever the genus of the manifold is.

This property can be briefly argued as follows.

By definition, an edge E_i has two extremities, whose pre-images by Ψ are circles. In particular, these circles form the boundary components of the chart C_i . By definition, critical points and particularly saddle points equivalence classes are not included in Reeb charts.

Moreover, C_i has genus zero since the Reeb graph of a simple Morse function f defined on an orientable and closed 2-manifold having genus g has g loops (CMEH*03).

Hence, Reeb charts have genus zero, two boundary components and thus annulus topology. However, for charts adjacent to a local extremum of f , the related boundary component collapses to a point (the extremum). Thus, that kind of Reeb chart is given disk topology.

Thanks to the combinatorial properties of PL Morse functions (which are similar to those of smooth Morse functions), this result holds both in the smooth and discrete settings.

In particular, we defined the edges \mathcal{E}_i of the Reeb graph $\mathcal{R}(f)$ in the discrete setting as the maximally connected unions of equivalence classes of regular vertices of f (see the related definition page 116). Consequently, Reeb charts in the discrete setting, noted C_i , have the same number of boundary components as in the smooth setting. Finally, thanks to the combinatorial properties of PL Morse functions, the discrete Reeb graph $\mathcal{R}(f)$ of a simple PL Morse function f defined on a closed orientable triangulated surface S having genus g has also g loops. Thus, the Reeb charts in the discrete setting have also genus zero. Since they also have the same number of boundary components as in the smooth setting, Reeb charts C_i in the discrete setting indeed have the same topology of their smooth counterparts (either disk or annulus topology).

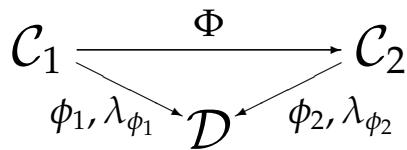


Figure 7.3 – *Reeb chart geometry similarity estimation process: each chart \mathcal{C}_i is mapped under ϕ_i to its canonical planar domain \mathcal{D} and its stretching signature λ_{ϕ_i} is computed with regard to the area distortion introduced by ϕ_i .*

In figure 7.2(d), disk-like Reeb charts have been colored in blue and annulus-like ones in red. Notice this decomposition of the hand triangulated surface brings a certain *visual semantic* since each of the fingers forms a distinct chart.

7.2.2 Reeb chart unfolding

To provide a concise and intrinsic characterization of Reeb charts' geometry, we propose to focus on parameterization techniques (GV04, WML*06, BCG08) described in the geometry shape modeling dedicated section of the survey chapter of this manuscript (see section 3.2.2 page 37).

These techniques aim at describing the intrinsic 2D geometrical information carried by the surface independently from its 3D spatial embedding. In particular, Gu and Vemuri (GV04) proposed a complete and intrinsic surface geometry representation based on mean curvature and conformal factor (or *stretching factor*) computation. Ben-Chen and Gotsman (BCG08) noticed recently that curvature computation is quite noisy in the discrete setting and only consider the *stretching factor* λ for shape comparison (like we already proposed in a former paper (TVD07a)).

Given two Reeb charts \mathcal{C}_1 and \mathcal{C}_2 of equivalent topology to compare, we propose to characterize a map $\Phi : \mathcal{C}_1 \rightarrow \mathcal{C}_2$. However, in the framework of partial shape retrieval, we are interested in the definition of Reeb chart geometry signature. Hence, we characterize Φ using the transition mappings ϕ_1 and ϕ_2 to the Reeb charts' common canonical domain \mathcal{D} : a surface having the same topology and constant Gaussian curvature (in the case of Reeb charts: either the unit planar disk or the unit planar annulus), as shown in figure 7.3: $\Phi := \phi_2^{-1} \circ \phi_1$.

Then, the geometrical similarity between \mathcal{C}_1 and \mathcal{C}_2 is evaluated by comparing the mappings ϕ_1 and ϕ_2 . In particular, we characterize ϕ_1 and ϕ_2 by their *stretching signature* (λ_{ϕ_1} and λ_{ϕ_2}), a function of the area distortion they introduce.

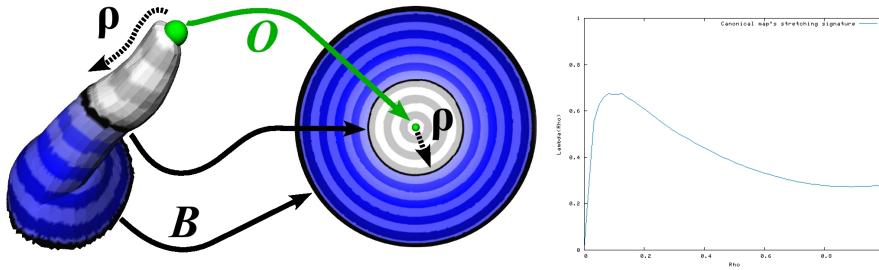


Figure 7.4 – Disk-like Reeb chart unfolding signature computation.

To guarantee affine and isometric invariance, we propose to consider a Reeb chart parameterization based on the PL Morse function f to feature vertices. Thanks to the Reeb graph properties, only two cases have to be considered: disk-like Reeb charts and annulus-like Reeb charts (cf. property 11, page 154).

Disk-like Reeb chart unfolding

Given a disk-like Reeb chart \mathcal{C}_i , let O be the extremum vertex of f it contains and B its (unique) boundary component. We let ϕ_i map O to the center of the unit planar disk \mathcal{D}_D (the canonical domain of disk-like surfaces), B to its boundary component and f level lines to concentric circles, as shown in figure 7.4 where the thumb of the hand triangulated surface of figure 7.2 has been mapped to its planar domain \mathcal{D}_D .

Let $\rho(v) \in (0, 1]$ be the normalized absolute difference of f values between O and a vertex $v \in \mathcal{C}_i$, as shown in figure 7.4. Consequently to the Reeb chart definition, the sub-level sets of ρ have also disk topology, as illustrated by the grey sub-level set in figure 7.4. In particular, as ρ increases, the geometry of the sub-level sets varies. Thus, it induces an evolution in the distortion introduced by their mapping to \mathcal{D}_D .

Consequently, to capture the evolution of Reeb charts' geometry variation, we propose to introduce the *stretching signature* λ_{ϕ_i} of ϕ_i as follows:

$$\lambda_{\phi_i}(\rho) = \frac{\mathcal{A}_{\mathcal{C}_i}(\rho)}{\mathcal{A}_{\mathcal{D}_D}(\rho)} = \frac{\mathcal{A}_{\mathcal{C}_i}(\rho)}{\pi\rho^2} \quad (7.1)$$

where $\mathcal{A}_{\mathcal{C}_i}(\rho)$ stands for the area of the sub-level set for parameter ρ on \mathcal{C}_i and $\mathcal{A}_{\mathcal{D}_D}(\rho)$ stands for the area of the sub-level set on \mathcal{D}_D . $\mathcal{A}_{\mathcal{C}_i}(\rho)$ is computed by summing the areas of the related triangles of \mathcal{S} (after having normalized edge length by f , similarly to ρ). Figure 7.4 (right) shows λ_{ϕ_i} (Y-axis) with regard to ρ (X-axis). Roughly, $\lambda_{\phi_i}(\rho)$ depicts the stretch one has to apply on the Reeb chart to map it to a disk as ρ increases.

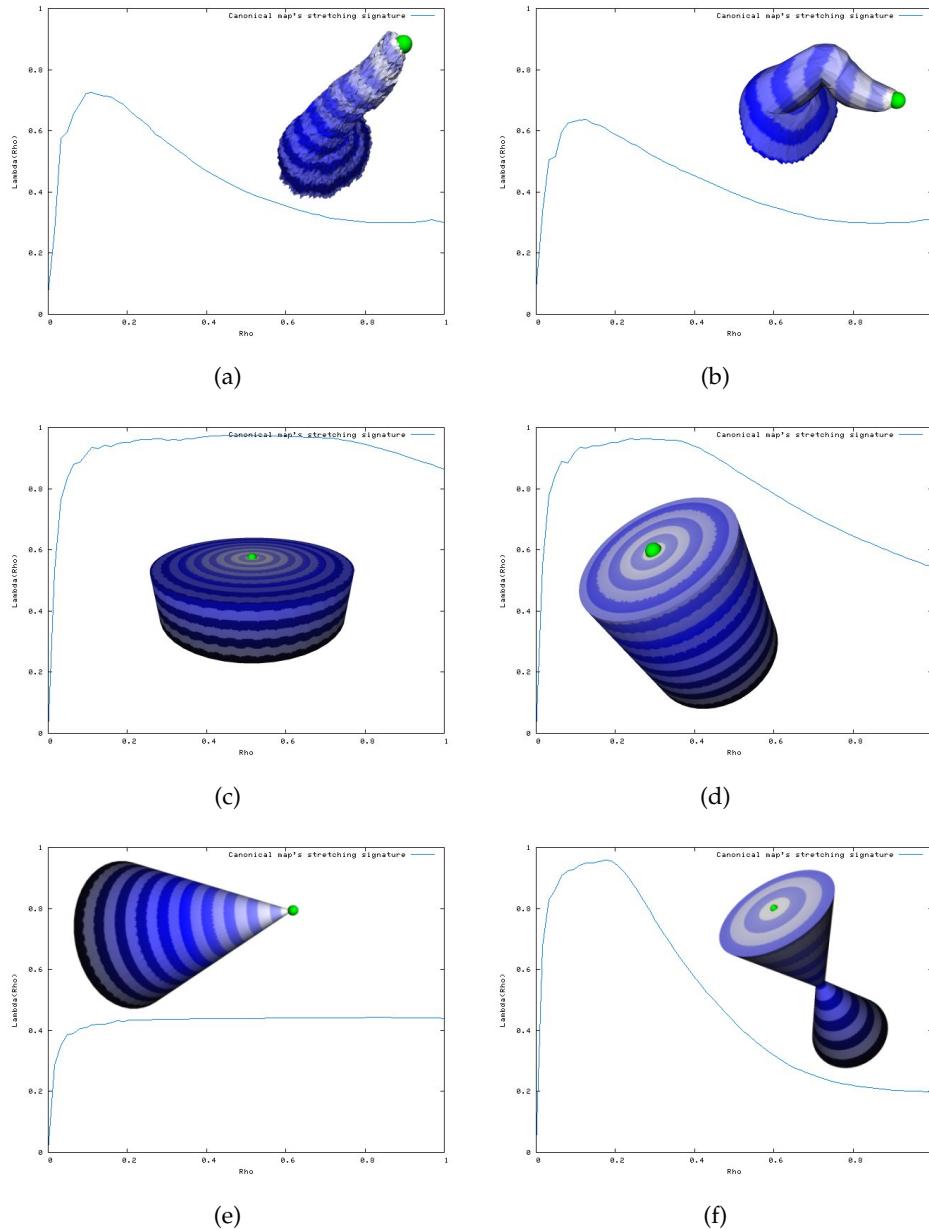


Figure 7.5 – Example of stretching signatures for altered versions of the thumb chart and other primitive charts.

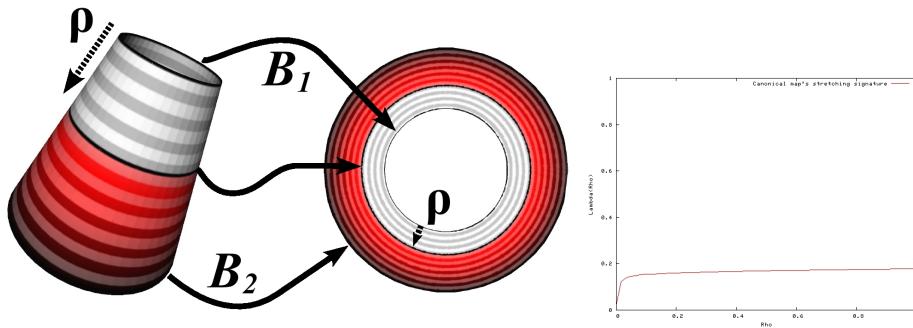


Figure 7.6 – Annulus-like Reeb chart unfolding signature computation.

Figure 7.5 shows some examples of stretching signatures for various disk-like primitives. As ρ is based on f , it inherits from its affine and isometric invariance (f is based on geodesic distances), as shown in figure 7.5(b), where the signature of a bent version of the thumb gives a signature nearly identical to the original (figure 7.4, right). Moreover, it is also robust to random noise, as shown in figure 7.5(a).

The close relationship between the stretching signature and the represented geometry can be underlined by the following remarks. In figure 7.5(c), \mathcal{C}_i is planar until $\rho = 0.65$, thus $\lambda_{\phi_i}(\rho)$ tends to 1 until $\rho = 0.65$. For a cone, $\lambda_{\phi_i}(\rho) = \sin(\alpha)$ where $\alpha = \arctan(\frac{r}{h})$ where r and h stand for the radius and the height of the cone. Hence, $\lambda_{\phi_i}(\rho)$ tends to a constant term, as shown in figure 7.5(e).

Annulus-like Reeb chart unfolding

An analog reasoning can be applied for annulus-like Reeb charts. Let B_1 be the boundary component of shortest perimeter of an annulus-like chart \mathcal{C}_j and B_2 the other one. In this case, we let ϕ_j map B_1 to the inner boundary component of the unit planar annulus \mathcal{D}_A (the canonical domain of annulus-like surfaces) and B_2 to its outer boundary component, as shown in figure 7.6. By defining the ρ parameter similarly to the previous paragraph, we can state that the sub-level set of ρ have also annulus topology, as illustrated by the grey sub-level set in figure 7.6. Hence, we introduce the stretching signature λ_{ϕ_j} of ϕ_j as follows (1 is the inner radius of the unit annulus):

$$\lambda_{\phi_j}(\rho) = \frac{\mathcal{A}_{\mathcal{C}_j}(\rho)}{\mathcal{A}_{\mathcal{D}_A}(\rho)} = \frac{\mathcal{A}_{\mathcal{C}_j}(\rho)}{\pi(\rho + 1)^2 - \pi} \quad (7.2)$$

Notice that in figure 7.6, the chart is based on a truncated version

of a cone; thus its signature is similar to that in figure 7.5(e) (modulo a multiplicative factor which depends on the openness of the cone).

As a conclusion, Reeb chart unfolding signatures are intrinsic and stable geometry characterizations of the Reeb charts. Moreover, thanks to their conciseness (λ_{ϕ_i} is a vector) they are definitely suited for fast geometry comparison in the context of partial 3D shape retrieval.

7.3 REEB PATTERNS IN REEB GRAPHS

At this stage of the framework, the input orientable, connected and closed triangulated surface \mathcal{S} is concisely represented by a dual Reeb graph \mathcal{R} , whose nodes are accompanied with the unfolding signatures of corresponding Reeb charts.

The next step of the framework consists in shape partial similarity estimation. Given two input triangulated surfaces \mathcal{S}_1 and \mathcal{S}_2 , the goal is to identify their similar sub-parts and to evaluate their geometrical similarity.

In the following paragraphs, we introduce the notions of Reeb pattern, Reeb graph structural distortion and propose a structural signature for Reeb patterns which is insensitive to such a distortion (avoiding the use of error-tolerant graph matching algorithms). As in the previous paragraph, for the sake of intuition, we first present new notions in the smooth setting.

7.3.1 Reeb patterns and structural distortion

Definition 89 (Reeb pattern) *Let \mathbb{M} be an orientable, compact, smooth, connected and closed 2-manifold, $R(f)$ its Reeb graph relatively to a simple Morse function f and C_i an annulus-like Reeb chart. Let B^- be the boundary component of C_i with lowest f value, noted f_{B^-} . Let \mathbb{M}^+ be the sub-set of \mathbb{M} such that $\mathbb{M}^+ = \{m \in \mathbb{M} \mid f(m) > f_{B^-}\}$. The Reeb pattern P_i associated to the annulus-like Reeb chart C_i is the connected component of \mathbb{M}^+ having B^- as only boundary component.*

An analog definition of Reeb patterns in the discrete setting, noted \mathcal{P}_i can be easily extended from the above definition.

Figure 7.7 shows an Armadillo triangulated surface with the symbolic embedding of its enhanced topological skeleton with regard to the PL Morse function f to feature vertices. In particular, figure 7.7 zooms in two Reeb patterns \mathcal{P}_1 and \mathcal{P}_2 and their related sub-graphs \mathcal{G}_1 and \mathcal{G}_2 . Moreover, the annulus-like Reeb charts associated to \mathcal{P}_1 and \mathcal{P}_2 have been marked with a red circle. Roughly speaking, a Reeb pattern \mathcal{P}_i is a surface

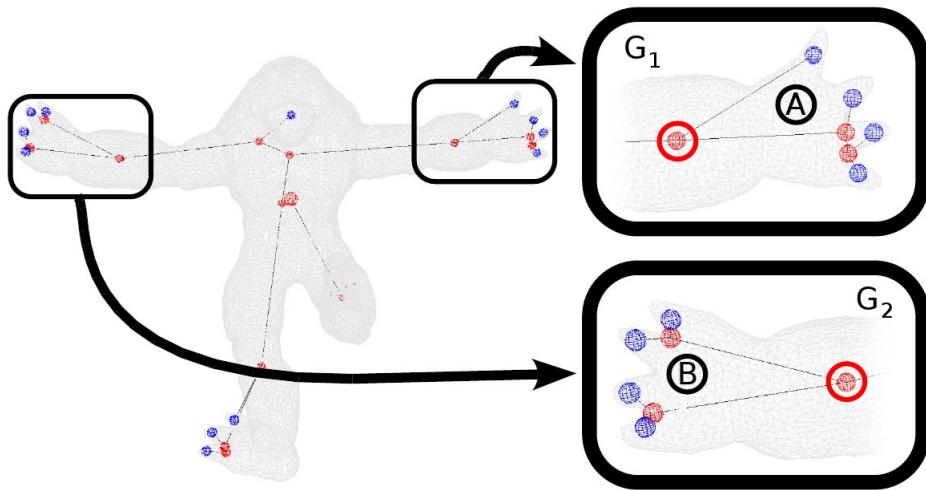


Figure 7.7 – Structural distortion on two visually similar and topology equivalent Reeb patterns.

sub-part which includes nearby protrusions (this is a set of annulus and disk-like Reeb charts, delimited by \mathcal{C}_i upper boundary component).

Notice Reeb patterns are not necessarily defined for each annulus-like Reeb chart (they are not defined for pairs of charts forming handles).

Reeb graphs can suffer from noise and distortion when they represent real world objects. From our experiences, feature vertex extraction is very stable on similar 3D shapes. This means very few noise due to inconsistent feature vertex extraction will appear. However, Reeb graphs still suffer from distortion.

An important result of Morse theory is that any smooth real valued function defined on compact smooth manifold can be transformed into a Morse function by a slight perturbation, transforming its degenerate critical points into non-degenerate ones. This result depicts the fact that small perturbations on the function can change the property of its critical point set. In particular, in our case, a slight perturbation on the surface (and thus on the function) can change the sequence of bifurcations and junctions in the Reeb graph. As a consequence, in figure 7.7, the sub-graph \mathcal{G}_1 first bifurcates in the A -labelled region while the sub-graph \mathcal{G}_2 first bifurcates in the B -labelled region. Consequently, no isomorphism exists between \mathcal{G}_1 and \mathcal{G}_2 while the two related Reeb patterns (the hands of the Armadillo) are visually similar and topology equivalent. We refer to this phenomenon as Reeb graph *structural distortion*.

7.3.2 Reeb pattern topology

To overcome this issue, in order to compare topology equivalent Reeb patterns despite structural distortion, we introduce the notion of *structural signature* of a Reeb pattern.

Definition 90 (Reeb pattern structural signature) *Let \mathbb{M} be an orientable, compact, smooth, connected and closed 2-manifold, $R(f)$ its Reeb graph relative to a simple Morse function f and P_i a Reeb pattern associated with an annulus-like Reeb chart C_i . Let $n_D(P_i)$ and $n_A(P_i)$ be respectively the number of disk-like and annulus-like Reeb charts included in P_i . The couple $(n_D(P_i), n_A(P_i))$ is the structural signature of the Reeb pattern P_i .*

Property 12 (Structural signature topological invariance) *The structural signature $(n_D(P_i), n_A(P_i))$ of P_i fully describes P_i topology since $n_D(P_i)$ and $n_A(P_i)$ are linked by the following relation:*

$$n_D(P_i) = n_A(P_i) + 1 - 3g \quad (7.3)$$

where g is the genus of the Reeb pattern.

Proof. Let f be a simple Morse function defined on a compact closed and orientable 2-manifold \mathbb{M} and $R(f)$ its Reeb graph.

Let P be a Reeb pattern defined relatively to $R(f)$. By definition, P is a compact 2-manifold with one boundary component.

Let P' be the compact and orientable 2-manifold without boundary obtained by the closure of the boundary of P (where the unique boundary component of P is glued by contraction to a point, as shown in figures 7.8 and 7.9). Let f' be a simple Morse function defined on P' such that it has the same set of critical points and critical values on P' than f on P , plus one minimum (due to the closure of the boundary component of P). f' has distinctly valued critical points and all its critical points are non-degenerate. f' is indeed a simple Morse function. For example, in figures 7.8 and 7.9, f and f' are the height functions. Let $R(f')$ be the Reeb graph of f' (depicted by black arrows on figures 7.8 and 7.9, right).

On the first hand, the Euler characteristic $\chi(P')$ of P' is given by the generalized Euler relation (FK97) (where $g_{P'}$ is the genus of P'):

$$\chi(P') = 2 - 2g_{P'} \quad (7.4)$$

On the other hand, the Euler characteristic $\chi(P')$ of P' can also be expressed by the Morse-Euler formula (FK97) since P' is a closed manifold

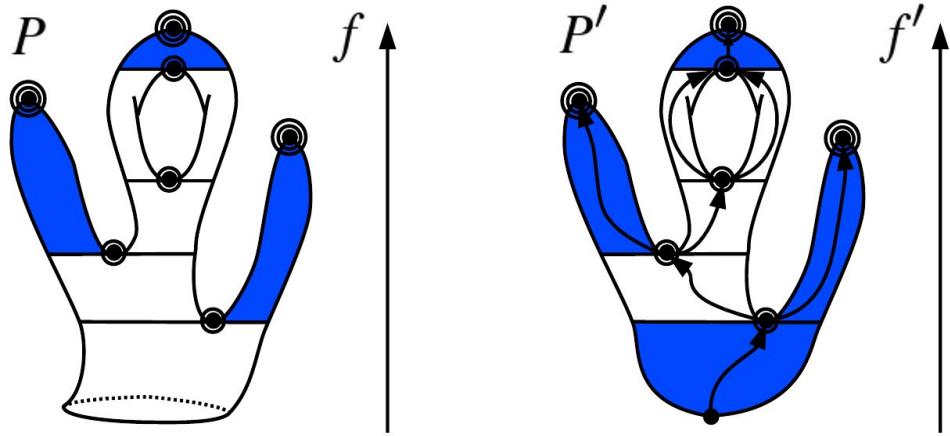


Figure 7.8 – Enumeration of the disk-like Reeb charts of a Reeb pattern. Left: Original Reeb pattern P . Right: Reeb pattern P' after the closure of the unique boundary component of P . Disk-like Reeb charts are in blue.

(where $k = 2$ is the dimension of the manifold):

$$\chi(P') = \sum_{i=0}^k (-1)^i \mu_i(f') = \mu_0(f') - \mu_1(f') + \mu_2(f') \quad (7.5)$$

In equation 7.5, $\mu_i(f')$ stands for the i^{th} Morse number of f' , which is equal to the number of f' critical points of index i . In particular, $\mu_0(f')$, $\mu_1(f')$ and $\mu_2(f')$ are respectively the number of f' local minima, saddles and maxima. In figures 7.8 and 7.9, each critical point is marked with a number of concentric circles equal to its index i .

Disk-like Reeb chart enumeration

Let $n_D(P')$ be the number of disk-like Reeb charts of P' . By definition, each disk-like Reeb chart of P' is exactly adjacent to one local extremum of f' . Moreover, the degree in $R(f')$ of equivalence classes of minima and maxima equals one (Ree46) (see figure 7.8, right). Then, a local extremum of f' can only be adjacent to one disk-like Reeb chart of P' . Thus:

$$n_D(P') = \mu_0(f') + \mu_2(f') \quad (7.6)$$

Moreover, by definition, f' has exactly the same set of critical points and critical values on P' than f on P , plus one minimum due to the closure of the boundary component of P (see figure 7.8). Consequently, P and P' have the same Reeb chart decomposition at the exception of the chart adjacent to the additional minimum of f' , which is transformed from an annulus-like Reeb chart (in P , figure 7.8, left) to a disk-like Reeb chart (in P' , figure 7.8, right) by the closure of the unique boundary component of

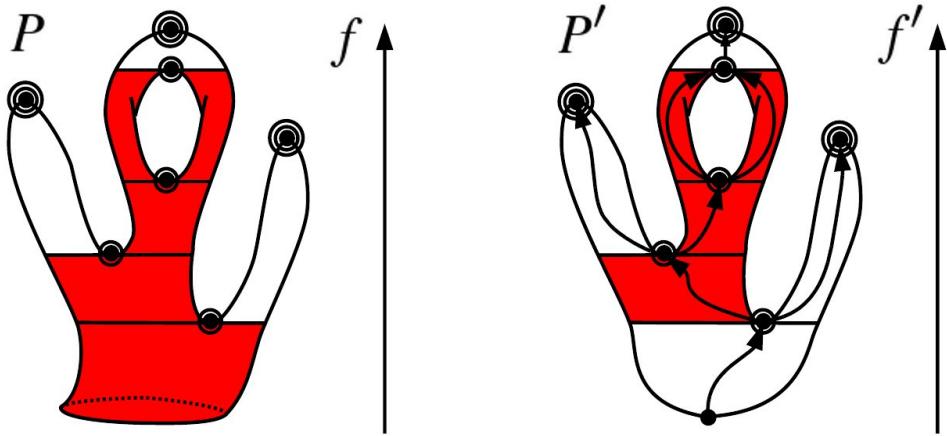


Figure 7.9 – Enumeration of the annulus-like Reeb charts of a Reeb pattern. Left: Original Reeb pattern P . Right: Reeb pattern P' after the closure of the unique boundary component of P . Annulus-like Reeb charts are in red.

P . Hence:

$$n_D(P) = n_D(P') - 1 = \mu_0(f') + \mu_2(f') - 1 \quad (7.7)$$

Annulus-like Reeb chart enumeration

Let $n_A(P')$ be the number of annulus-like Reeb charts of P' . $R(f')$ is a finite and connected one-dimensional simplicial complex (Ree46). Consequently, it can be considered as a planar graph, whose vertices are the equivalence classes corresponding to critical points of f' and whose edges correspond to the Reeb charts of P' . Then, the Euler relation for planar graphs holds:

$$\chi(R(f')) = V_{R(f')} - E_{R(f')} + F_{R(f')} = 2 \quad (7.8)$$

where $V_{R(f')}$ is the number of critical points of f' :

$$V_{R(f')} = \mu_0(f') + \mu_1(f') + \mu_2(f') \quad (7.9)$$

where $E_{R(f')}$ is the number of Reeb charts of P' :

$$E_{R(f')} = n_D(P') + n_A(P') \quad (7.10)$$

and where $F_{R(f')}$ is the number of faces of the planar graph (with $L_{R(f')}$ the number of loops in $R(f')$):

$$F_{R(f')} = L_{R(f')} + 1 \quad (7.11)$$

Moreover, $R(f')$ has $g_{P'}$ loops (CMEH*03) (with $g_{P'}$ the genus of P'). Then, equations 7.8, 7.9, 7.10 and 7.11 become:

$$\begin{aligned} \chi(R(f')) &= 2 = \mu_0(f') + \mu_1(f') + \mu_2(f') \\ &\quad - (n_D(P') + n_A(P')) + g_{P'} + 1 \end{aligned} \quad (7.12)$$

Thanks to equation 7.6, we have:

$$\begin{aligned} 2 &= \mu_1(f') - n_A(P') + g_{P'} + 1 \\ n_A(P') &= \mu_1(f') + g_{P'} - 1 \end{aligned} \quad (7.13)$$

Moreover, by definition, f' has exactly the same set of critical points and critical values on P' than f on P , plus one minimum due to the closure of the unique boundary component of P (see figure 7.9). Consequently, P and P' have the same Reeb chart decomposition at the exception of the chart adjacent to the additional minimum of f' , which is transformed from an annulus-like Reeb chart (in P , figure 7.9, left) to a disk-like Reeb chart (in P' , figure 7.9, right) by the closure of the unique boundary component of P . Hence:

$$\begin{aligned} n_A(P) &= n_A(P') + 1 \\ n_A(P) &= \mu_1(f') + g_{P'} \end{aligned} \quad (7.14)$$

Relation between $n_D(P)$ and $n_A(P)$

Consequently to equation 7.7 and 7.14, equation 7.5 becomes:

$$\begin{aligned} \chi(P') &= \mu_0(f') - \mu_1(f') + \mu_2(f') \\ \chi(P') &= n_D(P) + 1 - (n_A(P) - g_{P'}) \end{aligned} \quad (7.15)$$

Therefore, thanks to the equation 7.4, we have the following relations:

$$\begin{aligned} n_D(P) + 1 - n_A(P) + g_{P'} &= 2 - 2g_{P'} \\ n_D(P) &= n_A(P) + 1 - 3g_{P'} \end{aligned} \quad (7.16)$$

Moreover, by definition, the boundary of a Reeb pattern is required to be composed of a single connected component. Then, the boundary of a Reeb pattern cannot lie on a topological handle. Consequently, the closure of the unique boundary component of P cannot modify its genus g_P . Then:

$$g_P = g_{P'} \quad (7.17)$$

Thanks to the equation 7.17, we have the final result:

$$n_D(P) = n_A(P) + 1 - 3g_P \quad (7.18)$$

□

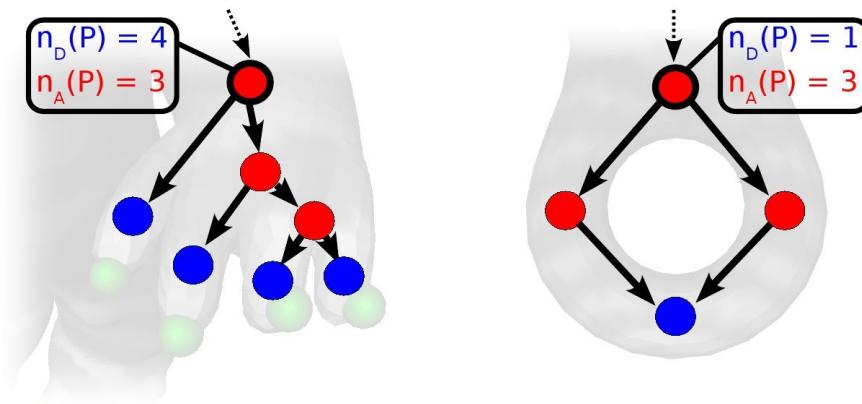


Figure 7.10 – Structural signatures of a genus-0 and a genus-1 Reeb pattern.

This property is mainly due to the combinatorial properties of smooth simple Morse functions and of Reeb graphs. As simple PL Morse functions and Reeb graphs in the discrete setting have similar combinatorial properties (see the Morse-Euler relation for triangulated surfaces page 86), the above reasoning also holds in the discrete setting.

By definition, Reeb patterns are orientable, compact, smooth and connected 2-manifolds with only one boundary component. In term of topology description, only their genus differs. Thus their topology is fully described through their structural signature $(n_D(P_i), n_A(P_i))$. Moreover, this signature does not encode the way the bifurcations and junctions are sequenced in the Reeb pattern. Thus it does not depict structural distortion. These properties also hold in the discrete setting.

Figure 7.10 gives two examples of structural signatures of a genus-0 and a genus-1 Reeb pattern. In the rest of the approach, structural signatures help both overcoming structural distortion and reducing the search space of the problem.

7.4 PARTIAL SIMILARITY COMPUTATION BASED ON REEB PATTERN UNFOLDING

In the following paragraphs, we introduce an efficient approximation of the brute-force approach for the maximally similar sub-graph computation problem. It does not consider all the node-to-node combinations but the topology equivalent Reeb pattern to topology equivalent Reeb pattern combinations only.

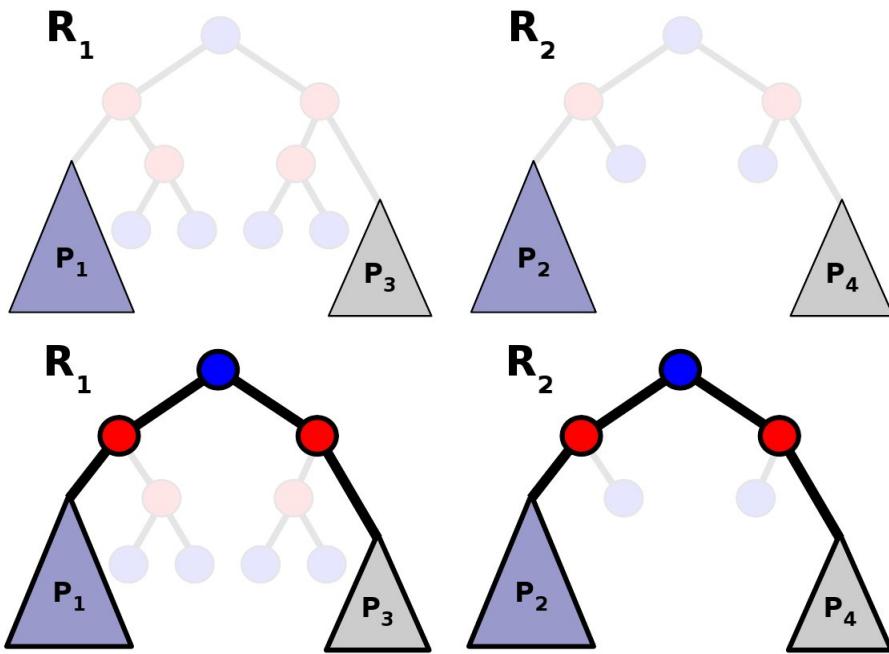


Figure 7.11 – Expansion example: for a given combination of topology equivalent Reeb patterns ((P_1, P_2) and (P_3, P_4)), the expansion algorithm tries to expand as much as possible the common sub-graphs (in bold in the second row).

7.4.1 Reeb pattern combination enumeration

First, given two shape signatures \mathcal{R}_1 and \mathcal{R}_2 to compare, the structural signature of each Reeb pattern \mathcal{P}_i of \mathcal{R}_1 and \mathcal{R}_2 is computed.

Then, for each identified Reeb pattern \mathcal{P}_i , we compute the set of topology equivalent Reeb patterns (whose structural signatures are identical) in the other graph. Let $\mathcal{T}(\mathcal{P})_1$ and $\mathcal{T}(\mathcal{P})_2$ be respectively the sets of Reeb patterns of \mathcal{R}_1 and \mathcal{R}_2 that have a *homologue* in the other graph (a pattern whose structural signature is identical).

Next, we compute the set \mathcal{M} of all the possible injective mappings $m : \mathcal{T}(\mathcal{P})_1 \rightarrow \mathcal{T}(\mathcal{P})_2$ that map a Reeb pattern $\mathcal{P}_1 \in \mathcal{T}(\mathcal{P})_1$ to a homologue Reeb pattern $\mathcal{P}_2 \in \mathcal{T}(\mathcal{P})_2$. In our experiments, the average cardinality of \mathcal{M} is 20 (which has no comparison with the $\frac{n!}{(n-k)!}$ combinations of the brute-force approach).

7.4.2 Expansion of common sub-graphs

Next, given an association m of Reeb patterns, the common sub-graphs $\mathcal{G}_1^m \subset \mathcal{R}_1$ and $\mathcal{G}_2^m \subset \mathcal{R}_2$ are computed thanks to an expansion algorithm which simultaneously visits \mathcal{R}_1 and \mathcal{R}_2 , starting by Reeb pattern associations, and which adds visited node associations to m .

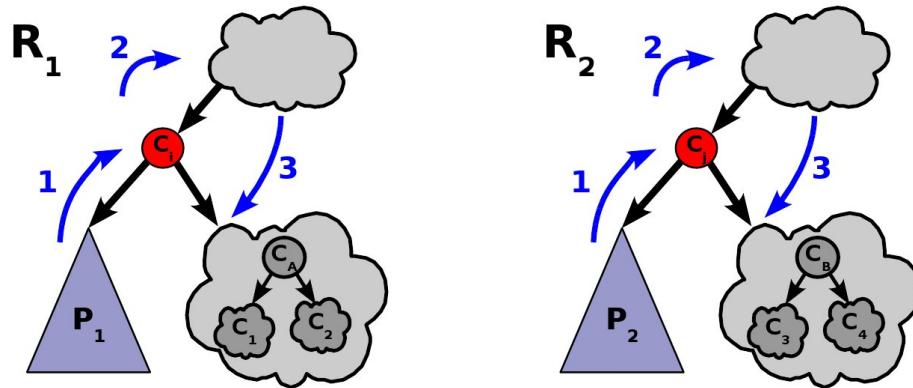


Figure 7.12 – Expansion process from two topology equivalent Reeb patterns \mathcal{P}_1 and \mathcal{P}_2 . Steps 1, 2 and 3 are depicted by blue arrows.

Figure 7.11 shows an example of expansion: the algorithm expands the initial mapping m by recursively visiting nodes that are adjacent to the associated Reeb patterns.

The expansion algorithm is composed of the following steps, denoted by blue arrows in figure 7.12.

Step 1 Considering \mathcal{R}_1 and \mathcal{R}_2 as directed acyclic graphs (the direction goes towards increasing f values), the algorithm first visits the parent nodes C_i and C_j of \mathcal{P}_1 and \mathcal{P}_2 (figure 7.12, step 1). If the degree or the type (disk or annulus) of C_i and C_j differs, the expansion stops at this point and restarts from another unvisited pair of topology equivalent Reeb patterns. Otherwise, the expansion continues.

Step 2 The step 2 recursively repeats steps 1, 2 and 3 in this order. This means expansion continues the same way as far as visited nodes have the same degree and same type. Moreover, it stops when Reeb patterns formerly matched by m are reached.

Step 3 In step 3, the expansion continues on the unvisited children nodes of C_i and C_j with the same stopping condition (the degree or the type of visited nodes differs). However, let C_A and C_B be two nodes that have been matched in step 3. If their degree equals 3, a decision has to be taken as for the matching of their children C_1, C_2 and C_3, C_4 (see figure 7.12). Children whose Reeb patterns have been formerly matched by m are by definition associated. Remaining nodes are matched according to their degree and their structural signature. At this point, this expansion also stops if nodes have distinct degree or type.

The output of this expansion algorithm are two common sub-graphs $\mathcal{G}_1^m \subset \mathcal{R}_1$ and $\mathcal{G}_2^m \subset \mathcal{R}_2$, along with their Reeb pattern and node matches.

7.4.3 Intra-Reeb pattern node-to-node matching

The previous algorithm outputs node and Reeb pattern associations. In order to have a full node-to-node correspondence between the two sub-graphs to compare, we have to find a node-to-node mapping for each pair of topology equivalent Reeb patterns matched by m (like \mathcal{P}_1 and \mathcal{P}_2 in figure 7.12). Due to possible structural distortion, there might be no isomorphism between the sub-graphs of \mathcal{P}_1 and \mathcal{P}_2 .

Consequently, intra-Reeb pattern node matching is driven by geometry similarity only, running a bipartite matching algorithm (Mun57).

In particular, the bipartite matching algorithm is run twice: once for the sets of disk-like Reeb charts and once for the sets of annulus-like Reeb charts of the two Reeb patterns \mathcal{P}_1 and \mathcal{P}_2 . In each case, the assignment score $s \in [0, 1]$ between two Reeb charts \mathcal{C}_1 and \mathcal{C}_2 is the following:

$$s(\mathcal{C}_1, \mathcal{C}_2) = 1 - L_{N1}(\mathcal{C}_1, \mathcal{C}_2) \quad (7.19)$$

where $L_{N1}(\mathcal{C}_1, \mathcal{C}_2)$ stands for the normalized L_1 distance between the unfolding signatures λ_{ϕ_1} and λ_{ϕ_2} of \mathcal{C}_1 and \mathcal{C}_2 .

Given two Reeb patterns \mathcal{P}_1 and \mathcal{P}_2 , the bipartite matching algorithm matches the pair of topology equivalent charts that maximize their respective assignment score (with regard to the unfolding signatures), while maximizing the overall sum of assignment scores on the Reeb patterns \mathcal{P}_1 and \mathcal{P}_2 . In other words, the pairs of most geometrical similar Reeb charts of \mathcal{P}_1 and \mathcal{P}_2 are matched by this algorithm.

At this stage of the approach, m is a full node-to-node correspondence between the sub-graphs $\mathcal{G}_1^m \subset \mathcal{R}_1$ and $\mathcal{G}_2^m \subset \mathcal{R}_2$.

7.4.4 Partial similarity estimation

In this step of the comparison, each mapping m is scored relatively to the geometrical similarity S of associated common sub-graphs $\mathcal{G}_1^m \subset \mathcal{R}_1$ and $\mathcal{G}_2^m \subset \mathcal{R}_2$. At this point, several similarity functions can be defined. In our experiments, we use the following S function:

$$S(m) = \frac{\sum_{\forall \mathcal{C}_i \in \mathcal{G}_1^m} (1 - L_{N1}(\mathcal{C}_i, m(\mathcal{C}_i)))^\alpha}{|\mathcal{R}_q|} \quad (7.20)$$

In equation 7.20, $L_{N1}(\mathcal{C}_i, m(\mathcal{C}_i))$ stands for the normalized L_1 distance between the unfolding signature of \mathcal{C}_i and $m(\mathcal{C}_i)$. α stands for a corrective

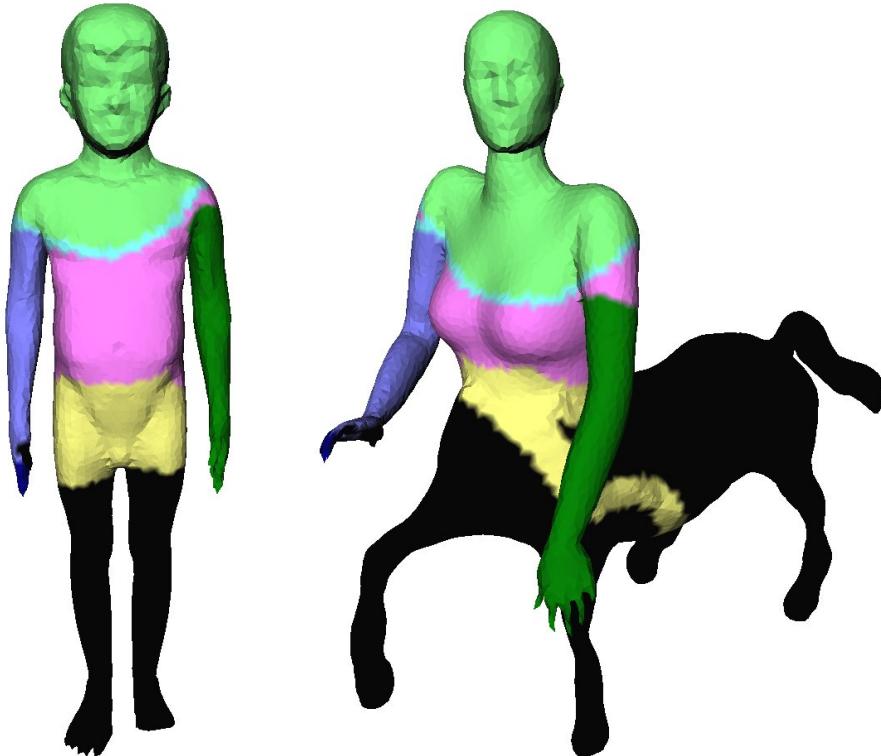


Figure 7.13 – Reeb chart (bright colors) and Reeb pattern (dark colors) matching between a boy and a centaur. Unmatched charts are black.

parameter whose aim is to amplify the Reeb chart geometry similarity contribution (after training, α is set to 4). Finally, $|\mathcal{R}_q|$ stands for the number of nodes of the dual Reeb graph that represents the query triangulated surface \mathcal{S}_q presented to the system.

Finally, the mapping \widehat{m}^* that maximizes S is the approximation of the optimal solution m^* . Moreover, $\widehat{\mathcal{G}}_1^*$ and $\widehat{\mathcal{G}}_2^*$ are the approximations of the most similar sub-parts of the two triangulated surfaces \mathcal{S}_1 and \mathcal{S}_2 , whose partial similarity is given by $S(\widehat{m}^*)$.

Figure 7.13 shows an example of the approximated optimal mapping \widehat{m}^* between two sub-graphs $\widehat{\mathcal{G}}_1^*$ and $\widehat{\mathcal{G}}_2^*$ of the two triangulated surfaces, along with the node-to-node and pattern-to-pattern matching.

7.5 EXPERIMENTAL STUDY

To assess the efficiency of the approach we have described, we evaluate its performance on the Shape REtrieval Contest 2007 (SHREC 2007) par-

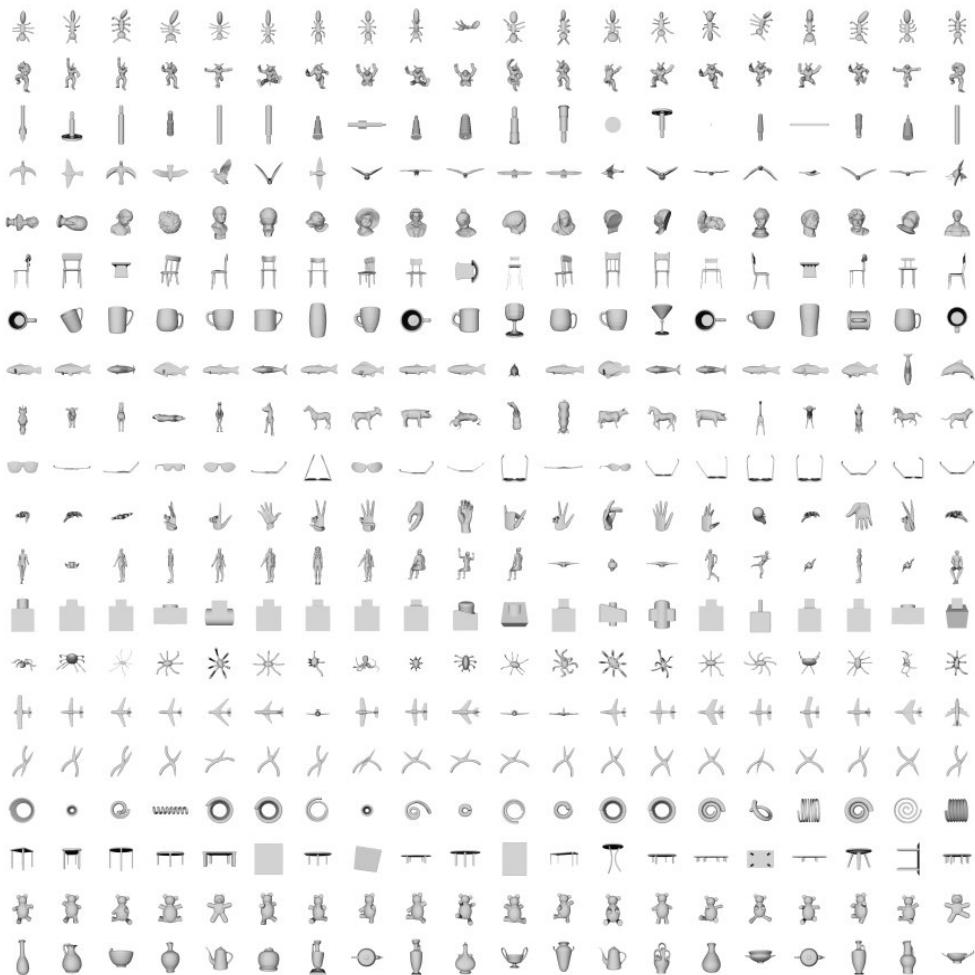


Figure 7.14 – SHREC 2007 data-set snapshot: one class per row: ants, armadillos, bearings, birds, busts, chairs, cups, fishes, four-legs, glasses, hands, humans, mechanics, octopuses, planes, pliers, springs, tables, teddies and vases.

tial retrieval benchmark (MPBo7)², and make some comparisons with the methods competing in this contest.

7.5.1 SHREC 2007 partial shape retrieval benchmark

This benchmark is composed of a data-set of 400 triangulations of closed 2-manifolds (grouped in 20 classes, as shown in figure 7.14) and of a query-set of 30 triangulations of closed 2-manifolds, as shown in figure 7.15.

The data-set exhibits diverse variations within a same class: affine and isometric transformations, Boolean operations, under-sampling or topology variation (notice 4 of the 20 classes contain non zero genus surfaces). Consequently, this data-set is a challenging test for evaluating the invariance of the proposed 3D shape modeling approach.

²No partial shape retrieval benchmark has been proposed at SHREC 2008.

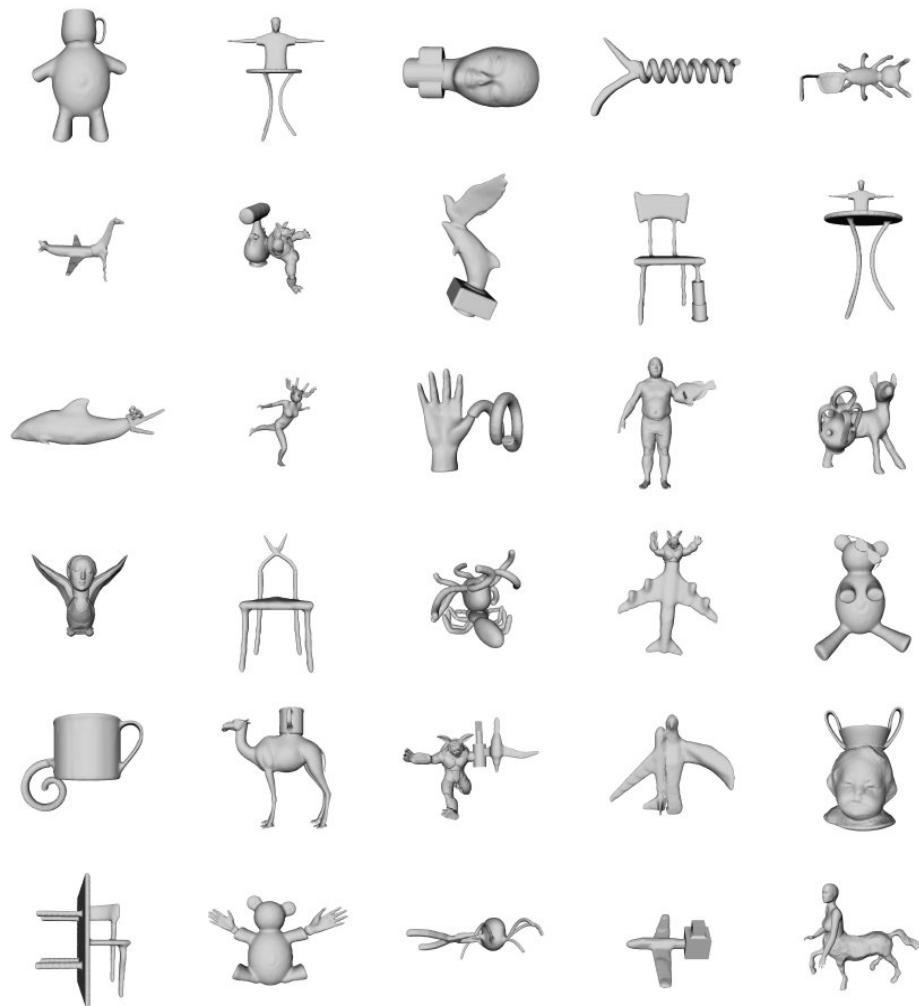


Figure 7.15 – SHREC 2007 query-set snapshot.

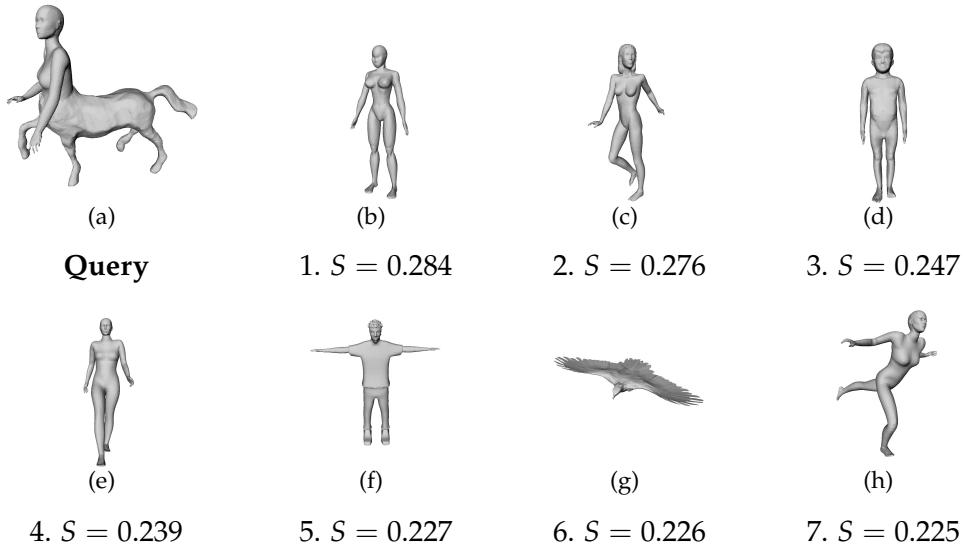


Figure 7.16 – A query from the SHREC 2007 query-set (a centaur) and the top-7 results retrieved by our system.

Each query of the query-set shares visually similar sub-parts with at least two classes of the data-set.

Figures 7.16, 7.17 and 7.18 show three examples of typical queries and the triangulated surfaces retrieved by our system.

Moreover, for a given query, the benchmark provides a ground-truth which divides the data-set into *Highly Relevant*, *Marginally Relevant* and *Non Relevant* class groups. This granularity enables to precisely evaluate the relevance of the results returned by the system. In particular, each relevance group is associated to a specific score, used in the computation of the Normalized Discounted Cumulative Gain (NDCG) vector. Roughly speaking, the higher is the $NDCG[i]$ value, the more relevant are the top- i results.

Such a performance measure provides a relevance overview over the whole data-set. Moreover, it can take into account several classes for scoring the query results, which is important in partial similarity since a query can be partially similar to several classes of objects. Furthermore, it is the only performance measurement taken into account in the SHREC 2007 contest. For more details about the benchmark, we defer the reader to the contest proceedings (MPBo7).

7.5.2 Experimental setup

First, the 400 triangulated surfaces of the data-set are indexed off-line. During the off-line process, the enhanced topological skeleton and the

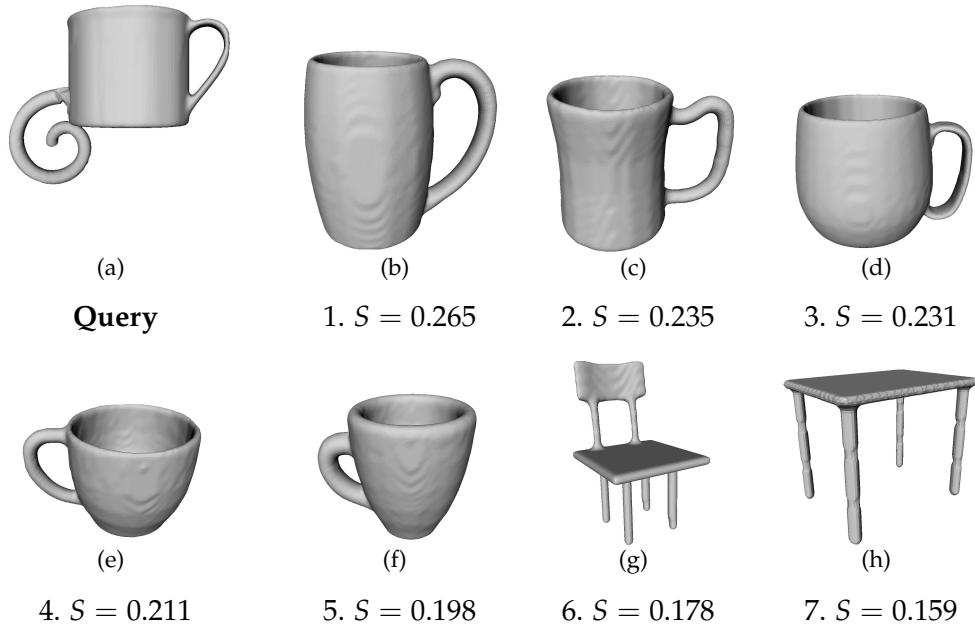


Figure 7.17 – A genus-1 query from the SHREC 2007 query-set and the top-7 results retrieved by our system.

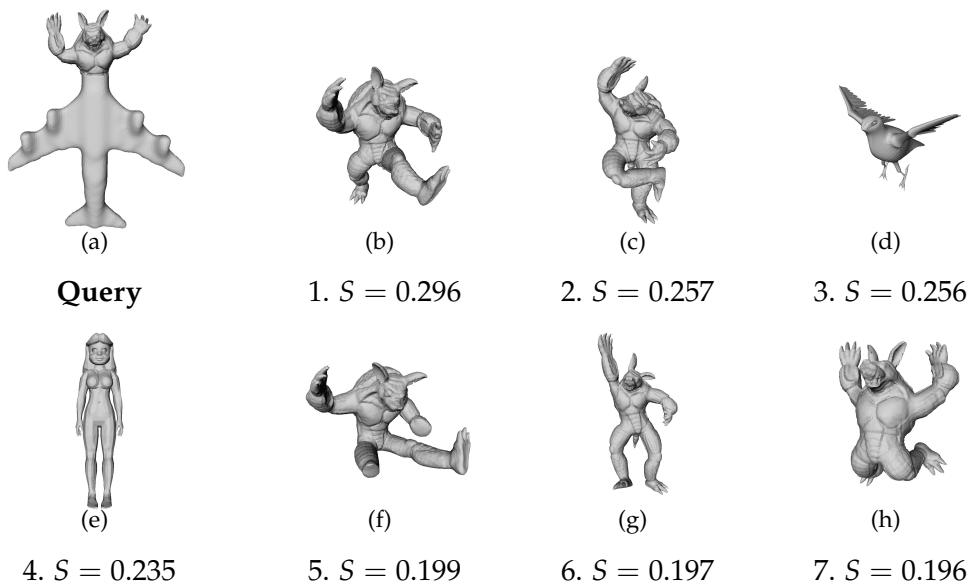


Figure 7.18 – Another query from the SHREC 2007 query-set (an Armadillo composed with a plane) and the top-7 results retrieved by our system. Notice Armadillos have been retrieved despite near-isometric transformations.

related unfolding signatures are computed for each triangulated surface and stored into the index file of the data-set.

During the on-line process, the enhanced topological skeleton and the unfolding signatures of the query \mathcal{S}_q are first extracted. Then, the shape signature \mathcal{R}_q of \mathcal{S}_q is compared with each graph of the index as described previously.

Finally, the entries of the data-set are sorted by decreasing value of partial similarity. Depending on the number of vertices in the query triangulated surface \mathcal{S}_q and the size of its signature graph \mathcal{R}_q , the full processing time of a query varies from **4 to 30 seconds** with the experimental setup described in section 5.3.5, page 120.

7.5.3 Performance evaluation

First, from a qualitative point of view, figures 7.16, 7.17 and 7.18 give a good overview of the efficiency of the proposed approach.

For example, in figure 7.16, the query is a centaur (half-horse, half-human) and thus most of the top results are humanoid models (first horses are retrieved at rank 15), even if they are not globally similar to the query. Figure 7.17 demonstrates the method's ability to handle non zero genus surfaces.

From a more quantitative point of view, in the first experiment, we compare the average NDCG vector of our approach with those of the methods competing in the contest (CDS*05, BMSFo6) (as reported in the contest proceedings (MPB07)). Such a vector is the average of the 30 NDCG vectors corresponding to the 30 queries of the query-set.

Figure 7.19 shows the curves corresponding to these vectors. As the *Reeb pattern unfolding* (RPU) curve is higher than the others, it follows that our approach outperforms the other methods.

Moreover, to quantify its improvement, we introduce the NDCG vector gain G , which is the ratio of the area between two curves A and B and the area below the lowest one (here B):

$$G(A, B) = \frac{\sum_{i=1}^{400} (NDCG[i]_A - NDCG[i]_B)}{\sum_{i=1}^{400} NDCG[i]_B} \quad (7.21)$$

With such a measure, the gain on the methods by Biasotti et al. (BMSFo6) and Cornea et al. (CDS*05) is respectively of **14,1%** and **40,9%**.

In the second experiment, we evaluate the impact of the choice of the similarity function S (see section 7.4.4, page 170) on the retrieval performances. In particular, we consider the function S_1 that returns the size

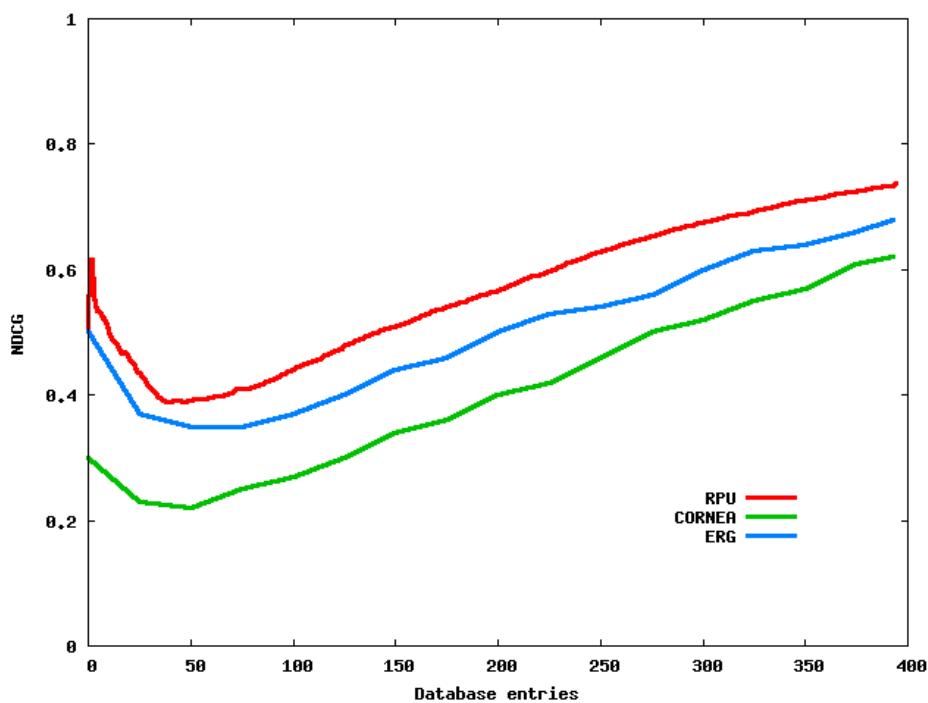


Figure 7.19 – Average Normalized Discounted Cumulative Gain (NDCG) vectors for Reeb pattern unfolding (RPU), (BMSFo6) (ERG) and (CDS*o5) (CORNEA) on the SHREC 2007 data-set.

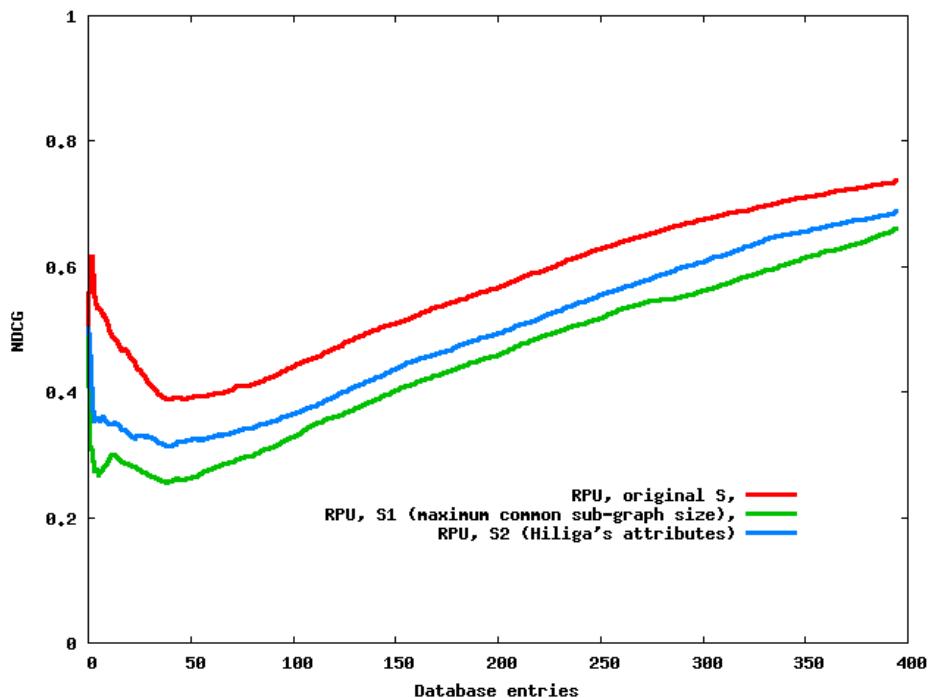


Figure 7.20 – Contribution of the Reeb chart unfolding signature to the the performances.

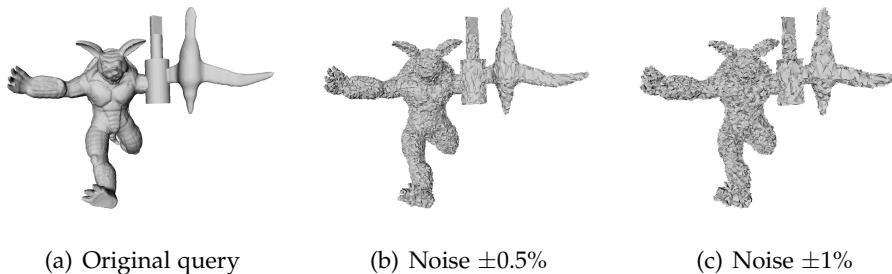


Figure 7.21 – Random noise on a SHREC 2007 query.

of the common sub-graphs (the maximally similar common sub-graphs become then the maximum common sub-graphs). We also consider the function S_2 where the unfolding signatures have been replaced by the geometrical attributes used by Hilaga et al. (HSKK01). The related curves are reported in figure 7.20. This figure shows that performances comparable to the other methods of the contest are obtained with S_2 . Moreover, it clearly illustrates the gain provided by the Reeb chart geometry modeling by unfolding signatures.

7.5.4 Robustness evaluation

As the enhanced topological skeleton relatively to the PL Morse function f to feature vertices and the unfolding signature computation is based on intrinsic surface measurements (geodesic distances), the approach is guaranteed to be invariant against affine and isometric transformations. In particular, figure 7.18 demonstrates its robustness against near-isometric transformations, since Armadillos in different poses have been retrieved as top results.

In the third experiment, we investigate the approach robustness against random noise. For each element of the query-set, the vertices' location is randomly moved with an amplitude bounded by respectively $\pm 0.5\%$ and $\pm 1\%$ of the lengths of the bounding box of the triangulated surface, as shown in figure 7.21.

The NDCG vectors with these corrupted query-sets have been reported in figure 7.22. These curves demonstrate the stability of the algorithm despite random noise. Moreover, even with a random noise of $\pm 0.5\%$, it still outperforms the scores on clean data of the method presented by Biasotti et al. (BMSFo6).

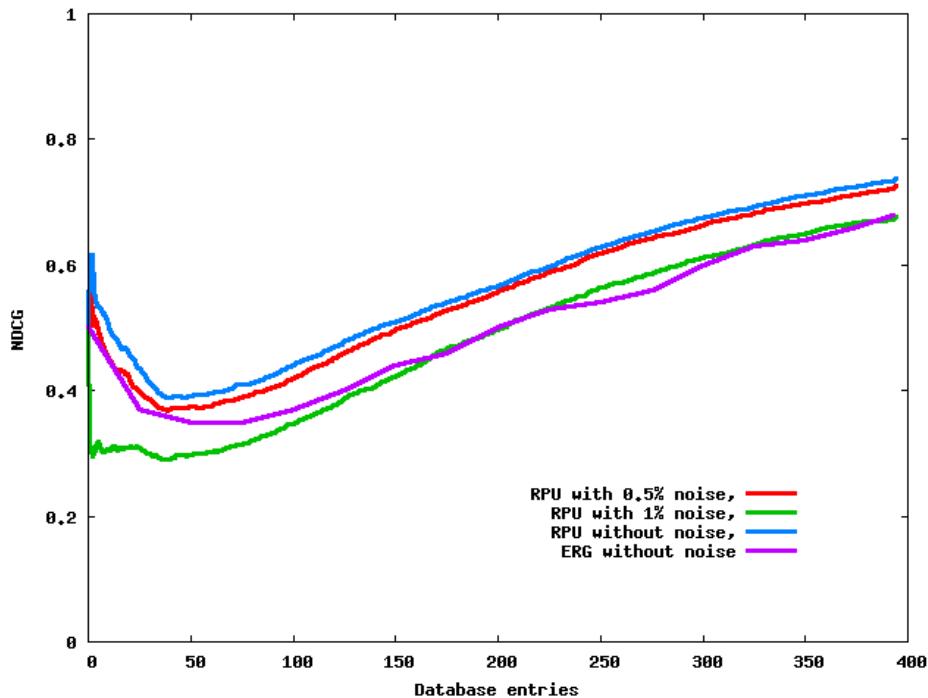


Figure 7.22 – Robustness evaluation of RPU with a noisy version of the SHREC 2007 query-set.

7.5.5 Discussion

As other structural based approaches, the surface decomposition step of the framework introduces a bias in the comparison process. To guarantee stability and performance, this decomposition has to be stable within a same class of objects and moreover consistent with the data-set ground-truth.

The bias introduced in the presented technique is based on feature vertex extraction, which drives the enhanced topological skeleton construction. Hence, this stage is a critical part of the framework and impacts its performances.

In practice, with the SHREC data-set, feature vertex extraction turns out to be homogeneous within most classes. This stability definitely contributes to the good performances of our approach on the SHREC 2007 partial shape retrieval benchmark. Reciprocally, the performances of our approach on this benchmark assess quantitatively the good invariance and expressive properties of 3D shape modeling by enhanced topological skeletons and feature vertex extraction.

7.6 APPLICATION TO MODELING BY EXAMPLE

To demonstrate the applicative interest of the proposed approach, we designed an intelligent modeling-by-example system (FKS^{*}04). Figures 7.23 and 7.24 show typical use examples.

First, the user queries the data-set through our partial shape retrieval system. Then, he/she selects on the query the Reeb patterns (or Reeb pattern sub-parts) he/she wants to remove (in red, green and blue in fig. 7.23(a)).

Next, the system automatically highlights the parts of the retrieved results candidates for swapping, based on our graph matching algorithm. The user then selects the retrieved results and cut the desired parts (highlighted or not, fig. 7.23(c), 7.23(d) and 7.23(f)).

After scaling, rotating and translating the parts according to his/her needs, the cut sub-parts are pasted back onto the query model (fig. 7.23(b)). In our experiments, boundaries are simply re-meshed but more visually appealing methods can be employed for gluing parts (SBSCOo6).

Thanks to the partial shape retrieval and the part swapping suggestions, a novice user can intuitively and rapidly create new 3D shapes with high geometric details based on the example parts of interest retrieved by our system.

7.7 CHAPTER CONCLUSION

In this chapter, we presented an application of enhanced topological skeleton based 3D shape modeling to partial 3D shape retrieval. This application is particularly interesting in the sense it can provide a quantitative measurement (through performance benchmarks) of the efficiency of the shape modeling approach.

By investigating Reeb graph theory, we both improved the 3D shape description and comparison steps with regard to state-of-the-art partial retrieval techniques.

Extensive experiments demonstrated the improvement of each of these steps, resulting in an overall gain of respectively **14,1%** and **40,9%** on the methods by Biasotti et al. (BMSFo6) and Cornea et al. (CDS^{*}05). Moreover, the invariance to affine and isometric transformations as well as the robustness to surface random noise have been shown. In practice, queries are processed in interactive time (from 4 to 30 seconds). Finally, in order

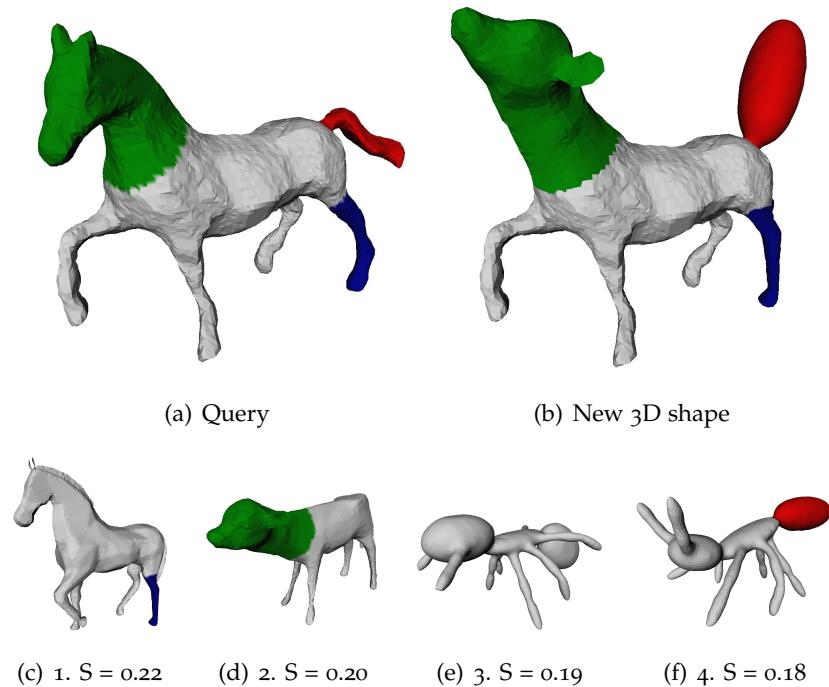


Figure 7.23 – Modeling a cow-horse by example: the user selects on the query, then on the retrieved results (second row) the Reeb patterns to be exchanged.

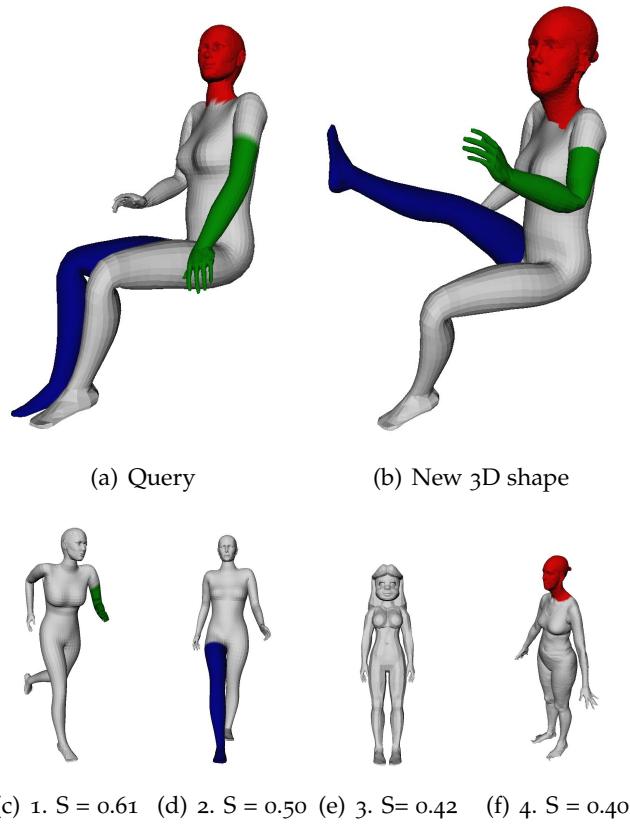


Figure 7.24 – Modeling a new woman 3D shape by example, composing both synthetic and scanned data.

to enable comparisons to future methods for partial 3D shape retrieval, a full implementation of our approach is freely available on the Web³.

In particular, along the resolution of this research problem, we developed the new concepts of *Reeb charts* and *Reeb patterns* on Reeb graphs. We also provided theoretical results on their respective topological properties. This topology control then served our geometry modeling approach to propose intrinsic geometry signatures based on surface parameterization techniques, independently from the genus of the overall input triangulated surface. This example definitely illustrates the interest of differential topology based modeling as a preliminary for 3D shape geometry modeling. In this example, differential topology based modeling enabled to deal with parameterization intrinsic topology related limitations by providing a full topology control of the surface elements to compare.

Moreover, experiments showed our approached overcame the methods competing in the SHREC 2007 partial 3D shape retrieval benchmark, relatively to its pre-defined ground-truth. The good performances of our approach definitely assess quantitatively the good invariance and expressive properties of 3D shape modeling by enhanced topological skeletons and feature vertex extraction, as described in chapter 5, page 91.

7.8 FRENCH CHAPTER ABSTRACT

Dans ce chapitre, nous avons présenté une application des squelettes topologiques avancés à l'indexation 3D par similarité partielle. Cette application est particulièrement intéressante dans le sens où elle fournit des mesures quantitatives (au travers de *benchmarks*) de l'efficacité de l'approche de modélisation de forme.

En étudiant la théorie des graphes de Reeb, nous avons à la fois amélioré l'étape de description et de comparaison des formes par rapport aux techniques de l'état de l'art.

De nombreuses expériences ont montré l'amélioration apportée par chacune de ces étapes, aboutissant à un gain global de respectivement **14.1%** et **40.9%** sur les méthodes de Biasotti et al. (BMSFo6) et de Cornea et al. (CDS*05) sur le *benchmark* de l'édition 2007 du concours international d'indexation 3D (SHREC 2007) dans la catégorie indexation par similarité partielle. De plus, l'invariance aux transformations affines et isométriques ainsi que la robustesse au bruit aléatoire ont été montrés. En pratique, les

³ <http://www.lifl.fr/~tierny/sinamis.html>

requêtes sont traitées en temps interactif (de 4 à 30 secondes). Enfin, pour permettre des comparaisons de performance avec les futures méthodes d'indexation 3D par similarité partielle, une implémentation complète de notre approche est librement disponible sur le Web⁴.

Au cours de la présentation de notre solution à ce problème, nous avons développé les nouveaux concepts de *carte de Reeb* et de *motifs de Reeb* sur les graphes de Reeb. Nous avons également fournis des résultats théoriques sur leurs propriétés topologiques respectives. Ce contrôle de topologie nous a servi dans notre stratégie pour proposer des signatures intrinsèques de géométrie basées sur des techniques de paramétrisation de surface, ce indépendamment du genre de la surface en entrée. Cet exemple illustre résolument l'intérêt de la topologie différentielle comme préliminaire à toute modélisation géométrique. Dans cet exemple, la topologie différentielle nous a permis de contenir les limitations intrinsèques liées à la topologie des techniques de paramétrisation de surface, en proposant un caractérisation complète de la topologie des éléments de surface à comparer.

Par ailleurs, les expériences ont montré que notre approche surpassé les méthodes participant à SHREC 2007 dans la catégorie indexation par similarité partielle, relativement à la vérité terrain prédéfinie. Ces bonnes performances attestent également quantitativement des bonnes propriétés d'invariance et d'expressivité de l'extraction de sommets caractéristiques et de la modélisation 3D par squelette topologique avancé.

⁴<http://www.lifl.fr/~tierny/sinamis.html>

8

APPLICATION TO SEMANTIC-ORIENTED SEGMENTATION

CONTENTS

8.1	SCIENTIFIC CONTEXT	187
8.1.1	Related work	187
8.1.2	Motivations	187
8.1.3	Topology driven 3D mesh segmentation	188
8.1.4	Chapter overview	188
8.2	TOPOLOGY DRIVEN HIERARCHICAL SEGMENTATION	189
8.2.1	Semantic-oriented heuristics	189
8.2.2	Fine segmentation computation	190
8.2.3	Segmentation hierarchy	195
8.2.4	Experiments and results	197
8.3	DYNAMIC SURFACE MESH ANALYSIS	203
8.3.1	Related work	203
8.3.2	Motion characterization	205
8.3.3	Surface decomposition based on motion analysis	206
8.3.4	Early experiments	208
8.4	CHAPTER CONCLUSION	211
8.5	FRENCH CHAPTER ABSTRACT	213

THIS chapter introduces an application of enhanced topological skeleton based 3D shape modeling to the automatic functional decomposition of 3D shapes represented by triangulated surfaces.

First, by reasoning at an abstract level (directly on the enhanced topological skeleton), we define a decomposition strategy integrating heuristics inspired from human perception theory. Qualitative experiments based on a recent survey are provided to assess the validity of the approach.

Second, by analyzing the temporal evolution of dynamic meshes, we propose to infer the functional parts of a 3D shape by decomposing into parts of uniform motion along an animated sequence. We provide early results showing the qualitative and quantitative efficiency of the proposed approach with regard to existing techniques.

8.1 SCIENTIFIC CONTEXT

8.1.1 Related work

As we reviewed in the survey chapter of this manuscript (chapter 3, page 29), surface mesh segmentation aims at decomposing 3D shapes into segments of homogeneous properties. In particular, state-of-the-art techniques can be classified into two categories.

On the first hand, *low-level* methods aim at producing segments of homogeneous geometry or topology properties. Examples of low-level methods are those presented in (LPRM02, BM03, CSAD04, LDB05).

On the other hand, *high-level* techniques aim at decomposing the shape into segments of homogeneous perceptual properties. These works are motivated by psychological research (Bie87) which states that human perception tends to decompose shapes into shape primitives for their understanding. Consequently, the underlying ambition of *high-level* approaches is to provide a functional decomposition of the 3D shape, or to retrieve the semantic layout of the 3D shape, so as to provide the computer with a shape representation similar to those involved in human perception. This is the purpose of 3D shape understanding research.

High-level segmentation automatic techniques thus inspire from human perception theory (Bie87) to define perceptually consistent heuristics to drive the surface mesh decomposition. For example, Katz et al. (KLT05) and Liu and Zhang (LZ07) drive the segmentation by the notion of feature points. Podolak et al. (PSG*06) influence the process by the notion of perceived symmetry. Moreover, most of these techniques come in hierarchical flavor, providing multiple level of details in their decompositions (KTo3, KLT05). We defer the reader to the survey chapter of this manuscript (page 29) and to the survey articles (AKM*06, Shao08) for a broader overview of semantic-oriented segmentation techniques.

8.1.2 Motivations

In a similar fashion to the control skeleton driven paradigm for shape pose edition, semantic-oriented segmentation can serve as an automatic functional decomposition proposition for several tasks in Computer Graphics.

In the context of surface mesh artistic painting for example, a functional decomposition of the 3D shape provided automatically by the system can help the user in the specific painting of these parts.

Semantic-oriented segmentation has also shown to an interesting pre-

processing step in surface parameterization for texture mapping (ZMT05). Thanks to the provided decomposition, the user can map specific textures to each of the parts identified by the semantic-oriented segmentation.

Finally, among other applications, semantic-oriented segmentation also served as a pre-processing stage for 3D shape morphing (STK02).

8.1.3 Topology driven 3D mesh segmentation

Among the existing bias employed for semantic-oriented segmentation, several techniques recently proposed to drive the segmentation with topology considerations. Indeed, as underlined by human perception theory (Bie87), the structure of a shape plays an important role in human perception and thus topology shape modeling can provide high level information about the shape.

Li et al. (LTZ01) propose to construct a curve skeleton of the object to use this high-level structural information to drive the segmentation. Lien et al. (LKA06) extend this idea by constructing a hierarchical skeleton, analyzing the principal axis of its convex hull. However, this method is reported to be time consuming when dealing with non genus zero surfaces. Moreover, the convex hull of a 3D shape is an extrinsic surface characterization and varies under isometric transformations. Berretti et al. (BDBP06) propose a method that overcomes this issue using a derived version of the multi-resolution Reeb graphs proposed by Hilaga et al. (HSKK01). However their algorithm is not hierarchical and only consider the topology information for its decomposition computation.

8.1.4 Chapter overview

In this chapter, we investigate the use of enhanced topological skeleton based 3D shape modeling in the context of *high-level* 3D mesh automatic segmentation. In particular, we explore two main directions of research:

- 1. Human perception theory based heuristics** First, by defining human perception theory based heuristics, mostly inspired by previous automatic segmentation methods, we propose an algorithm for estimating a functional decomposition of the 3D shape represented by a triangulated surface. We analyze its pertinence according to the evaluation criteria enumerated in recent segmentation surveys (AKM*06).

2. Motion analysis of dynamic meshes Second, we present early results dealing with time varying 3D shape analysis. In particular, in this specific approach, we try to retrieve the 3D shape functional layout by decomposing it into parts of homogeneous motion along the time. Early experimental results are then presented and discussed.

8.2 TOPOLOGY DRIVEN HIERARCHICAL SEGMENTATION

In this section, we present a *high-level* 3D mesh segmentation automatic technique based on enhanced topological skeleton 3D shape modeling.

8.2.1 Semantic-oriented heuristics

First, to estimate the functional parts of 3D shape according to perceptual criteria, we enumerate a set of heuristics inspired from human perception theory (Bie87) and from previous state-of-the-art approaches.

Feature points

As mentioned by Katz et al. (KLT05) and Liu and Zhang (LZ07), the feature points of a surface (the points located at the extremity of the shape protrusions) are points of important geometrical and perceptual interest. Consequently, like in these approaches, we propose to drive the decomposition by feature points, thanks to the feature vertex computation algorithm presented in section 5.2.1, page 96.

Curvature analysis

Psychological research (Bie87) claims that human perception tends to decompose objects along its most concave regions. This hypothesis has been exploited in number of segmentation approaches so far to compute the boundaries of interest between the segments of the decomposition.

Katz and Tal (KT03) proposed an automatic *high-level* segmentation technique for triangulated surfaces based on a triangle clustering algorithm. In particular, the clustering step considers a triangle-to-triangle metric integrating both geodesic distance (for isometric invariance) and surface curvature, so that the boundaries of the clusters automatically follow the most concave parts of the surface. Here, we present a similar approach by also adding a curvature term in the computation of the PL Morse function f to feature vertices, so that f level lines follow the most

concave parts of the surface. Then, we also exploit curvature information and compute contour constrictions of this modified function to delimit the segments of the decomposition.

Perceived segment importance

Like in most of state-of-the-art techniques, decomposition segments which have a particularly small perceived size are merged with neighbor segments so as to prevent the so called *over-segmentation*. Moreover, Katz et al. (KLT05) notice that the core of a 3D object has a particular perceived importance and propose to isolate it as an individual segment. For example, in their approach, the *core* segment of a humanoid 3D shape will be its torso and it will form a unique segment in the decomposition. In our approach, we also integrate this perceptual hypothesis.

Topology characterization

High-level segmentation automatic techniques drive their decomposition by high-level notions like feature points, perceived symmetry or curve-skeletons. In this latter case, the topology characterization of the shape provides an interesting information about its structure. In our approach, we propose to use the enhanced topological skeleton of the input triangulated surface as initial shape representation and to achieve the segmentation by considerations on the enhanced topological skeleton. Moreover, to provide several level of details in the decomposition, we define a segmentation hierarchy which also integrates topological constraints.

8.2.2 Fine segmentation computation

PL Morse function

First, to characterize the protrusions of the input triangulated surface \mathcal{S} , its feature vertices are computed with the algorithm described in section 5.2.1 (page 96).

Then, the initial PL Morse function f_I (geodesic distances to feature vertices) is computed. In particular, we inspire from Katz and Tal (KT03) by adding a curvature term in the metric at the base of geodesic computation.

Given an edge $e \in \mathcal{S}$, we introduce the following metric between its

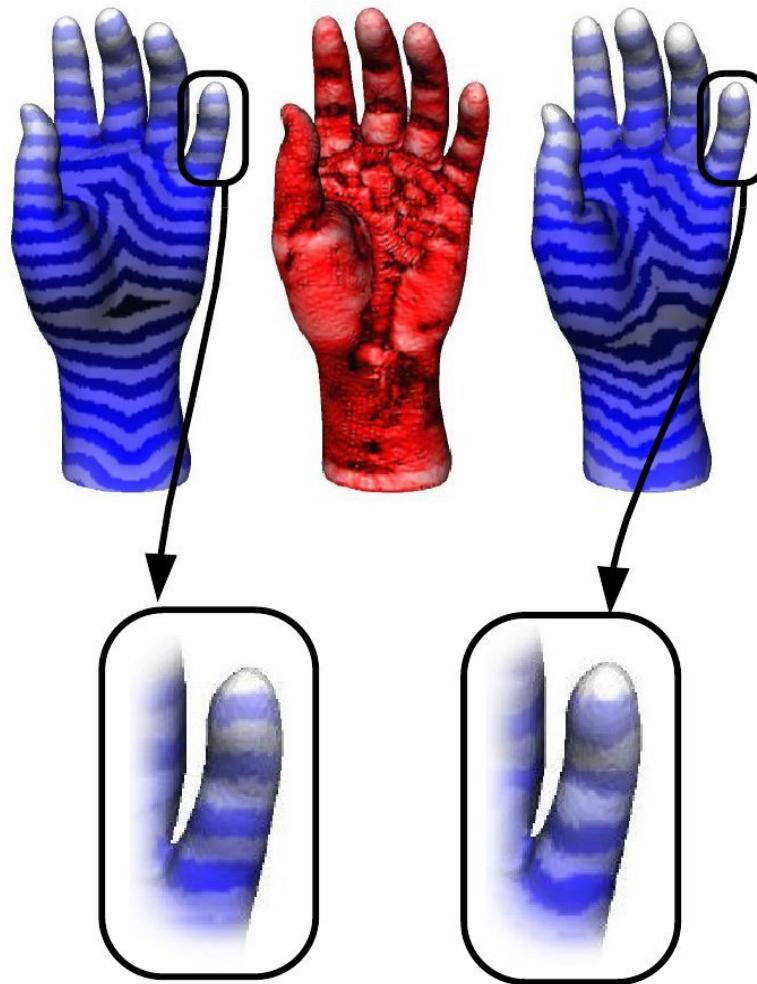


Figure 8.1 – Initial geodesic PL function to feature vertices (top left), curvature index $\kappa_{\mathcal{I}}$ (top middle) and PL function with the curvature term (top right).

vertices v_1 and v_2 (KT03):

$$m(e) = \frac{d(v_1, v_2)}{\bar{e}} + \omega \frac{|\kappa_{\mathcal{I}}(v_1) - \kappa_{\mathcal{I}}(v_2)|}{\Delta \kappa_{\mathcal{I}}} \quad (8.1)$$

where v_1, v_2 are the vertices of e , $d(v_1, v_2)$ the Euclidean distance between v_1 and v_2 , \bar{e} is the average edge length in \mathcal{S} and $\Delta \kappa_{\mathcal{I}}$ the average curvature index difference between adjacent vertices of \mathcal{S} . Moreover, like in (KT03), when $\kappa_{\mathcal{I}}(v_1)$ and $\kappa_{\mathcal{I}}(v_2)$ are both negative (which corresponds to concavity), ω is set to ω_0 and 0 otherwise.

This metric integrating a curvature term is then used in the computation of geodesics on \mathcal{S} . In practice, geodesics are computed with the Dijkstra algorithm (Dij59) considering the term $m(e)$ as edge weight.

Figure 8.1 shows the effect of computing geodesics with such a metric. Roughly speaking, the curvature term in equation 8.1 increases the distance between two vertices when a concave region separates them. The

curvature term thus makes the traversal of concave regions more *difficult* than convex ones, by adding a curvature based distance term. As a consequence, the geodesic distance function will first visit convex regions before crossing concave ones, forcing its level lines to align with concave regions, as shown in figure 8.1 (bottom right). Consequently, this function contours will be interesting candidates for the definition of segment boundaries located along the concave regions of the shape. In practice, ω_0 is set to 0.05 for all the experiments reported in the rest of the manuscript.

Once the curvature based PL geodesic function to feature vertices is computed, the perturbation strategy described in section 5.2.2 (page 103) is applied so that the PL function becomes a simple PL Morse function, noted $f_{\delta,\kappa}$ (then the Reeb graph of such a function is defined and can be computed).

Topological boundaries

The next step of our segmentation algorithm consists in computing the enhanced topological skeleton relatively to the previously defined simple PL Morse function $f_{\delta,\kappa}$. Then, the decomposition is achieved by reasoning directly on the enhanced topological skeleton.

In particular, in our decomposition approach, we focus on the computation of the set B of segment boundaries. First, we isolate the equivalence classes of the Reeb graph corresponding to saddle vertices. The related discrete contours are referred to as *topological boundaries* and are added to the set B of boundaries. The topological boundaries then correspond for example to the boundaries between the shape protrusions and the rest of the object. Moreover, they tend to be aligned with the concavity of the surface.

Concavity boundaries

Then, discrete contour computation is achieved, exactly as described in section 6.2.2 (page 135). As the $f_{\delta,\kappa}$ PL Morse function is based on geodesic and curvature computation, its level lines follow the concavity of the surface. Thus, the contour constrictions of $f_{\delta,\kappa}$ are also pertinent segment boundaries, according to the curvature based perception hypothesis stated in the previous section. Consequently, $f_{\delta,\kappa}$ contour constrictions are referred to as *concavity boundaries* and are added to the set B of segment boundaries.



Figure 8.2 – Segmentation abstract graph and resulting segmentation.

Segmentation abstract graph

The set B of segments boundaries is a set of closed curves (*concavity boundaries*) and of self-intersecting closed curves (*topological boundaries*). These curves partition the input triangulated surface into a set of connected components.

Let \mathcal{G} be the *segmentation abstract graph*, whose nodes depict the connected components delimited by the set B of segment boundaries and whose edges correspond to the adjacency relation between them. Figure 8.2 shows a symbolic embedding of \mathcal{G} (left) and the resulting shape decomposition (right) where each connected component delimited by segment boundaries have been displayed with a distinct color.

First, we classify the nodes N of \mathcal{G} into three types, according to their degree $\deg(N)$:

- $\deg(N) = 1$: *extremity node* (in green in figure 8.2, left). The corresponding surface segment contains a feature vertex.
- $\deg(N) = 2$: *tubular node* (in blue in figure 8.2, left). The corresponding surface segment is a connected surface element of genus zero and having two boundary components. Let B_i and B_o be respec-

tively its *inner* and *outer* boundary components. The outer boundary component B_o is directed towards the extremity nodes and have a higher $f_{\delta,\kappa}$ value than B_i which is directed towards the center of the object.

- $\deg(N) > 2$: *junction node* (in red in figure 8.2, left).

Then, we refer to the maximally connected sets of *junction nodes* as *junction areas*.

Core segment definition

As discussed by Katz et al. (KLT05), the core of a 3D shape plays an important role in its perceived layout. Consequently, they propose a technique to isolate a segment corresponding to the center of the object (like the torso of a humanoid). Here, we follow the same approach by defining the *core segment* as the junction area that contains $f_{\delta,\kappa}$ (only) minimum.

To conform to the perception heuristics (enumerated in the previous section), we simplify the segmentation abstract graph \mathcal{G} with the following strategy, by focusing on each *tubular node* $N_A \in \mathcal{G}$ which is adjacent to the core segment node N_C .

- If $\hat{g}(B_o) < \hat{g}_0$ and $\hat{g}(B_i) > \hat{g}_0$ (where \hat{g} is the average curvature index along the corresponding contour, see section 6.2.2, page 137), then N_A is merged with N_C (B_o is more concave than B_i , $\hat{g}_0 = -0.2$).

At the end of this process, the core segment's boundaries are guaranteed to be localized on concave parts of the shape.

- If $f_{\delta,\kappa}(B_o) - f_{\delta,\kappa}(B_i) < \Delta_{f_{\delta,\kappa}}$, then N_A is merged with N_C .

At the end of this process, segments of small size are merged with the core segment to prevent *over-segmentation* ($\Delta_{f_{\delta,\kappa}} = 0.1$).

Figure 8.3 shows the result of this perception heuristics based simplification. The junction area corresponding to the core segment is displayed with a red node (with bigger radius) in the segmentation abstract graph (figure 8.3 left). In this example, all the tubular nodes adjacent to the core segment have been merged with it. As a result, the core segment represents the palm of the hand and the other segments correspond to the wrist and the fingers (each finger is subdivided into phalanges).

At this stage of the approach, the algorithm provides a surface decomposition conformed to the perception heuristics defined in the previous

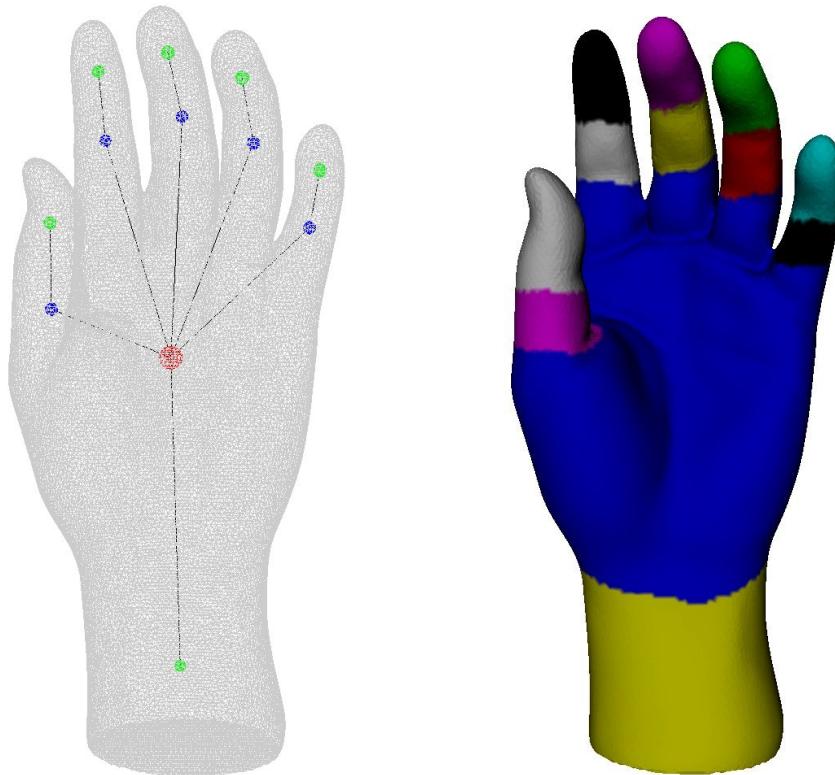


Figure 8.3 – Segmentation abstract graph (after perception heuristics based simplification) and resulting segmentation.

section. This segmentation corresponds to the finest level of decomposition of the approach. In the following, we present a method for the traversal of the simplified segmentation abstract graph, which provides a hierarchy of coarser decompositions.

8.2.3 Segmentation hierarchy

To conform to psychological research (Bie87), which states that human perception tends to decompose shapes into shape primitives for their understanding, we also give a special emphasis on the topology of the shape to define the segmentation hierarchy. In particular, along the hierarchy, we consider the topology of a segment as a more important decomposition criterion than its geometry.

Coarsest level

In the coarsest level of the segmentation hierarchy, we consider the difference S_d between the input triangulated surface S and its core segment. Each of the connected component S_d forms an individual segment at this level of hierarchy.

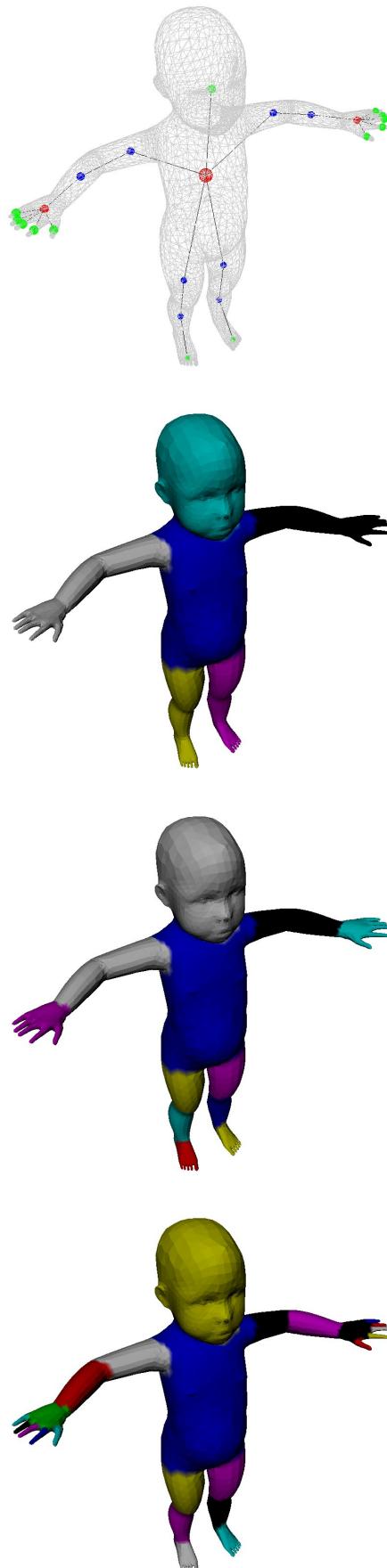


Figure 8.4 – Segmentation abstract graph (after simplification) and the hierarchy of resulting decompositions (coarsest to finest).

Roughly speaking, at this level, the segmentation decomposes the shape into its core and its other connected parts, as shown in figure 8.4 (second row). Equivalently, each segment of this level correspond to a branch of the segmentation abstract graph (figure 8.4, first row).

Intermediary levels

In the intermediary levels, the segments of the coarsest level are recursively subdivided according to the following strategy. For each segment of the previous level of hierarchy, the corresponding nodes of the segmentation abstract graph \mathcal{G} are swept by increasing order of $f_{\delta,k}$ (from the *center* of the object towards the extremity of its protrusions). If a junction area is encountered, the previously swept nodes are gathered into a new segment and the remaining nodes are gathered in a second new segment.

For example, in figure 8.4, the left arm segment of the first level (in grey, second row) is swept from the shoulder until a junction area is encountered at the palm of the hand (in red in the segmentation abstract graph, first row). Consequently, the swept nodes are gathered in a new segment (in grey) and the remaining ones in a second one (see the hand in pink).

If no junction area is encountered, the segment is sub-divided into the nodes it is composed of. For example, in figure 8.4, the left leg (in yellow, second row) has no junction area. Consequently, at the next level of hierarchy, this segment is subdivided according to its composing nodes in the segmentation abstract graph.

Finest level

The recursive segment subdivision process continues until there is no more possible subdivision. In other word, it stops when each node of the segmentation abstract graph correspond to an individual segment in the decomposition (figure 8.4, bottom row).

8.2.4 Experiments and results

In this section, we present experimental results of *high-level* decompositions of 3D shapes represented by triangulated surfaces using enhanced topological skeletons. In particular, we discuss the pertinence of the presented approach according to the seven following evaluation criteria, presented in a recent survey (AKM*06).

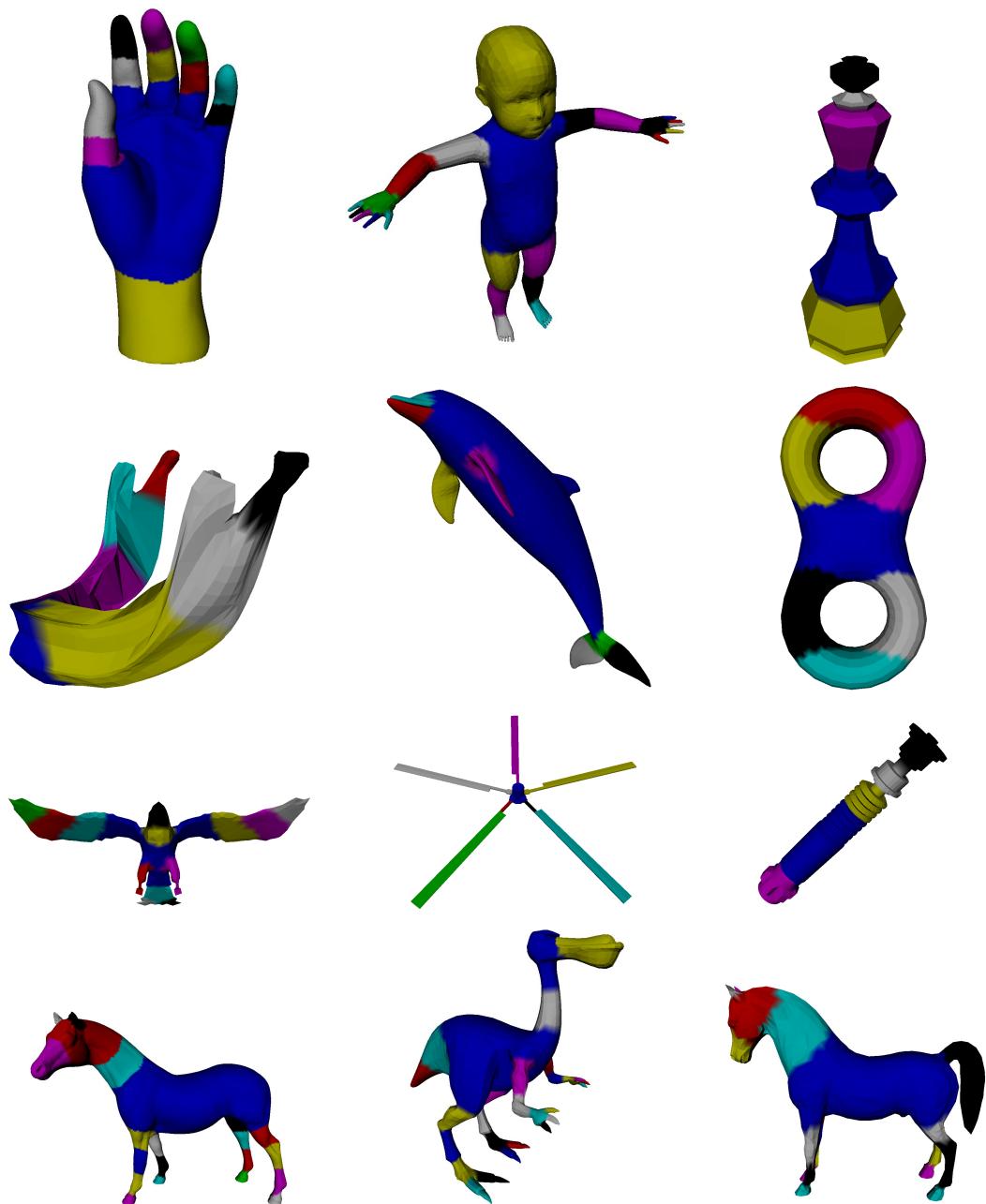


Figure 8.5 – Finest level of hierarchy for various 3D shapes.

Versatility

Figure 8.5 shows the ability of the approach to handle various categories of 3D shapes, like primitive shapes (with genus 2), animals, humanoids, anatomic shapes or mechanical elements.

Decomposition semantic

This criterion evaluates the pertinence of the heuristics involved in the decomposition. Figure 8.5 shows that animals and humanoids are decomposed into core and limbs and that limbs are further decomposed at plausible articulations, which correspond to functional shape decompositions.

Boundary location

This criterion evaluates the precision of the location of the boundaries, with regard to the actual concavity of the surface. As we introduce a curvature term in the computation of geodesic distances (in a similar way to Katz and Tal (KT03)), boundaries are supposed to be located on the surface concave regions. This result can be indeed observed on shapes with frank or coarse concavities (like the chess piece or the bird). However, when no contour constriction is identified, the triangulated surface is segmented only at its *topological boundaries*, which might not clearly reflect the functional layout of the object (like for the bi-torus).

Hierarchy semantic

On the contrary to related work, our hierarchy scheme is driven by high level notions, inferred from the surface topology. Figure 8.6 shows a comparison between the hierarchical decompositions of the *dinopet* 3D shape obtained with the algorithm of Katz et al. (KLT05) and our algorithm. The first level of the hierarchy (top left) provides the same number of segments: in both cases, the dinopet is decomposed into its core and its limbs (head, arms, legs and tail). However, with our algorithm, thanks to high-level topology based considerations, at the second level of the hierarchy, limbs are divided into arms and hands and legs and feet, which makes sense from a functional point of view. On the contrary, the method by Katz et al. (KLT05) divides these limbs in the middle. As a consequence, the hierarchy provided by our algorithm better depicts the semantic of the shape.

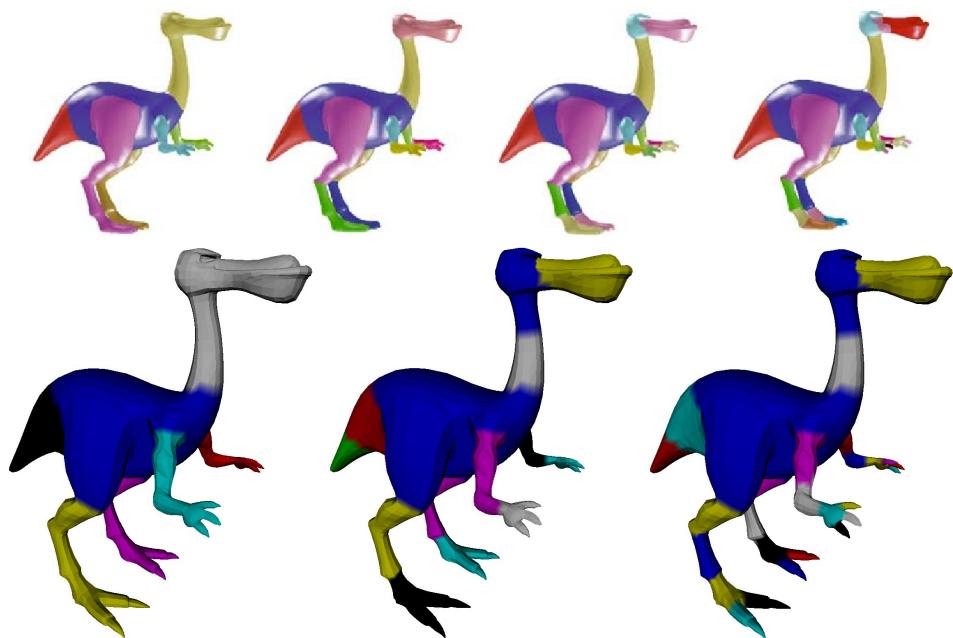


Figure 8.6 – Comparison between the hierarchical decompositions obtained with the algorithm of Katz et al. (KLT05) (top row) and our algorithm (bottom row).

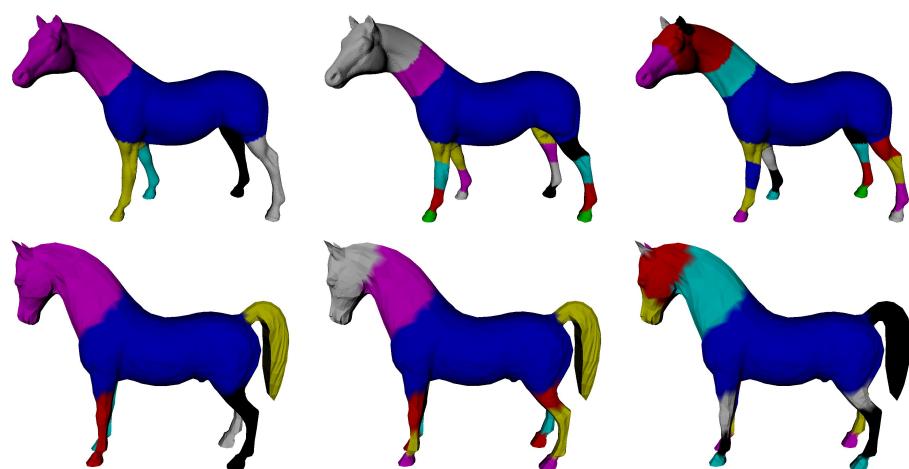


Figure 8.7 – Compatible hierarchies of decomposition for two similar objects.

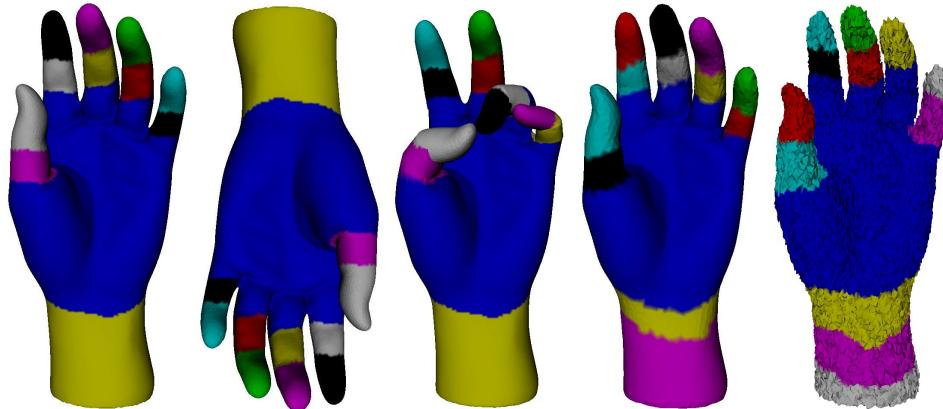


Figure 8.8 – Finest decompositions of the hand triangulated surface after affine and near-isometric transformations, under-sampling and random noise.

Moreover, figure 8.7 shows that our hierarchy scheme is compatible when similar objects are considered. First, horses are decomposed into core and limbs, then limbs are subdivided and the head is decomposed at the last level in both cases. Notice that the algorithm is not affected by the fact that the second horse (bottom row) has an additional feature (its tail).

Robustness

As the segmentation boundary definition is based on reasonings on the enhanced topological skeleton, the decomposition inherits from its affine and isometric invariance properties, as shown in figure 8.8. However, the boundary definition turns out to be slightly sensible to surface undersampling (the number of extracted segments varies) and quite sensible to random surface noise.

Time complexity

The overall complexity of enhanced topological skeleton and contour constriction computation is $O(N_V^2)$ in the worst case (with N_V the number of vertices in the input triangulated surface \mathcal{S}). Then the complexity of feature boundary selection and hierarchy definition only depends on the number of nodes in the segmentation abstract graph \mathcal{G} , which depends on the shape geometrical complexity and which is negligible with regard to N_V .

Table 8.1 shows running times obtained with the experimental setup described in section 5.3.5 (page 120). For comparison, a similarly sampled version of the dinopet 3D shape is segmented in 28 seconds in (KLT05), while our algorithm takes 3.9 seconds.

Triangulated surface	Triangles	Coarsest level	Finest level	Time (s.)
Hand	52000	7	12	107
Baby	10000	6	24	5.4
Chess piece	600	3	6	0.05
Human jaw	2300	3	7	0.43
Dolphin	4200	6	8	0.89
Bi-torus	6004	5	7	1.3
Bird	1996	6	14	0.4
Fan	456	6	8	0.06
Mechanical	2800	3	5	0.53
Horse 1	39996	6	21	34
Dinopet	9000	7	28	3.9
Horse 2	5000	7	20	1.35

Table 8.1 – Number of extracted surface segments and computation times for several triangulated surfaces.

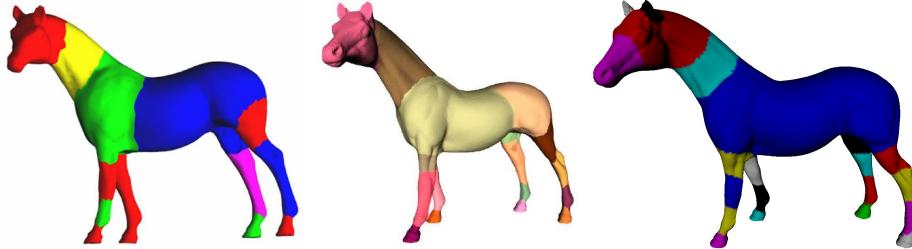


Figure 8.9 – Comparison to other topology driven segmentation algorithms (left: (ZMT05), middle: (LKA06)).

Control parameters

Some low level parameters have been introduced in the described approach (ω_0 , $\Delta f_{\delta,\kappa}$ and \hat{g}_0) but they have been fixed once experimentally for all the 3D shapes presented in this section. Therefore, the algorithm is fully automatic.

In comparison to other state-of-the-art topology driven techniques (see figure 8.9), thanks to the feature vertex computation algorithm, our method can isolate small shape features such as the ears of the Cyberware horse 3D shape, on the contrary to the algorithms described in (ZMT05, LKA06). Notice that our algorithm subdivides limbs more regularly than in (ZMT05) (where the front right leg, in red, is not subdivided). Moreover, as discussed above, our hierarchical scheme better depicts the functional layout of the shape than previous methods (KLT05). Finally,

running times are significantly lower than clustering based approaches (KTo3, KLT05).

8.3 DYNAMIC SURFACE MESH ANALYSIS

In this section, we explore another type of approach for the extraction of the functional layout of a 3D shape. The presented technique is based on the motion analysis of time-varying 3D shapes. In particular, the functional layout of the shape is inferred by decomposing the surface into segments of homogeneous motion along an animated sequence. This hypothesis is indeed a faithful hypothesis in the sense that segments that are affected by distinct motions might indeed ensure distinct functions of the represented object.

The modeling of time-varying 3D shapes is quite a new research topic and is gaining more and more interest in the 3D shape modeling community. In this manuscript, we provide early results for the functional decomposition of *dynamic* 3D shape using enhanced topological skeleton based shape modeling. This discussion gives a good perspective to the work presented so far in this manuscript, opening new research directions.

8.3.1 Related work

Dynamic triangulated surfaces

With the ongoing development of 3D technologies, 3D dynamic shapes (time-varying 3D data) are becoming a media of increasing importance. Such data can be provided by scientific simulations, animation software, video games or security systems.

Most of the time, 3D dynamic shapes are represented by time-varying surface meshes of constant connectivity. In particular, we focus here on dynamic triangulated surfaces.

Input dynamic 3D shapes are thus represented by a triangulated surface \mathcal{S} (see definition 59, page 80). However, the vertices v of \mathcal{S} are no longer associated to a unique position vector $\mathbf{p} \in \mathbb{R}^3$, but to a **sequence** of position vectors $\{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_i, \dots, \mathbf{p}_{N_F}\}$, where N_F is the number of frames in the animated sequence.

Homogeneous decompositions of dynamic triangulated surfaces

While static *high-level* 3D shape segmentation methods aim at extracting a functional layout of the represented object by decomposing it into segments of uniform perceptual properties, 3D dynamic shape segmentation methods propose to exploit the temporal information to decompose the shape into parts of uniform motion along the sequence. Consequently, state-of-the-art techniques first try to characterize the motion uniformity over the triangulated surface and then use clustering techniques to decompose it.

As most of real life objects' motions can be defined in term of local rigid transformations (translations and local rotations), Mamou et al. (MZPo6), in the context of dynamic shape compression, propose to locally compute the optimal rigid transformations along the sequence frames for small surface neighborhoods, using the least square method. Then, surface neighborhoods affected by the same rigid transformations along the sequence are gathered using the canonical k -means algorithm (DH73).

Lee et al. (LWCo6) adopt a similar strategy by first computing the so-called *deformation gradient* to characterize rigid transformation prediction error over the surface and then employ a clustering-based static mesh segmentation algorithm (KTo3) on this error field.

Skeleton extraction techniques for 3D dynamic surface meshes (SY07, dATTSo8) also characterize surface motion with rigid transformation predictions, combined with clustering algorithms.

However, predicting local rigid transformations turns out to be computationally expensive. Moreover, from our experience, if the considered surface neighborhoods are too small, an important proportion of the involved matrices turn out to be non-invertible, which prevents rigid transformation prediction error computation of a large number of surface neighborhoods. Furthermore, most of existing techniques only focus on the temporal information and thus cannot extract immobile shape features which still correspond to plausible functional parts.

In this section, we investigate the use of enhanced topological skeleton based 3D shape modeling for the decomposition of dynamic triangulated surfaces. In particular, we propose a simple, fast and efficient technique for motion characterization which is not based on affine transformation prediction. Then, by using calculus on the Reeb graph relative to the motion analysis, we identify *elastic contours* corresponding to the articulation of the object identified in the sequence analysis.

8.3.2 Motion characterization

Analysis overview

State-of-the-art techniques identify surface segments of uniform motion by focusing on affine transformation prediction. In particular, surface neighborhoods that are affected by the same affine transformation are gathered in the same segment of the final decomposition. It follows from this approach that the segmentation boundaries will be located on surface portions of irregular affine transformation repartition. Due to this irregularity, these surface portions (which match the final segmentation boundaries) will actually exhibit high affine transformation prediction error.

Consequently, in our approach, we try not to focus on the segments themselves, but on their boundaries, and propose to extract segment boundaries in the surface portions that exhibit a high affine transformation prediction error. Moreover, to avoid the affine transformation prediction task (which is computationally expensive) we start from the observation that affine transformations are a sub-group of isometric transformations. Consequently, surface portions that exhibit high affine transformation prediction error will also exhibit high isometric transformation prediction error, which is far easier to evaluate from a computational point of view.

Edge-length deviation based motion analysis

Isometric transformations are transformations which preserve surface metrics (defined through an inner product). In other words, isometric transformations preserve in particular the length of curves embedded on the surface. Notice this is the main motivation for the use of geodesic computation in the context of isometry invariant shape modeling. Thus, length preservation is also an invariant of affine transformations.

Consequently, we measure the length preservation error through the frames of the dynamic sequence to isolate the segmentation boundaries. In particular, to reduce the computational effort, we focus on the quadratic edge length deviation, noted \mathcal{L}_d :

$$\mathcal{L}_d(v) = \sum_{t=0}^{N_F-1} \sum_{\forall v_i \in Lk_0(v)} \frac{1}{|Lk_0(v)|} (d(v^t, v_i^t) - d(v^{t+1}, v_i^{t+1}))^2 \quad (8.2)$$

where N_F stands for the number of frames in the animated sequence and $d(v^t, v_i^t)$ for the Euclidean distance between v and v_i at frame t .

Figure 8.10 shows example frames of a dynamic triangulated surface and the repartition of \mathcal{L}_d (color gradient) on \mathcal{S} (first frame). Notice in this

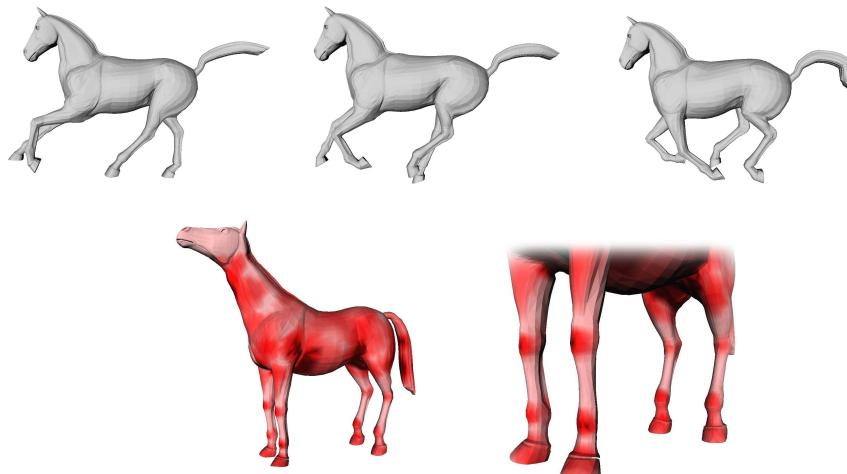


Figure 8.10 – Example frames of the horse dynamic triangulated surface (top row) and its quadratic edge-length deviation (bottom row).

figure that light pink surface portions (low edge-length deviation, bottom row) have been affected by local rotations in the sequence (top row) and correspond to rigid portions of the object. Moreover, the articulations of the legs exhibit a high edge-length deviation (dark red, bottom row) as they have been affected by elastic transformations.

8.3.3 Surface decomposition based on motion analysis

In the following, we define a surface decomposition strategy based on enhanced topological skeletons, which follows the approach proposed in the case of static shapes, described in the previous section.

Topological boundaries

First, the enhanced topological skeleton relatively to the PL Morse function f to feature vertices (defined in section 5.2, page 96) is computed for \mathcal{S} at the first frame. In the case of dynamic triangulated surfaces, decomposition hypothesis are based on motion characterization only. Thus no curvature consideration is integrated in the computation of the PL Morse function and the former f function is chosen for further computations.

By definition, \mathcal{S} has constant connectivity along the animated sequence. Thus, its topology does not evolve through time. Consequently, the enhanced topological skeleton of f is computed only once, for the first frame.

Let B be the set of segment boundaries of the shape decomposition. First we isolate the equivalence classes of the Reeb graph $\mathcal{R}(f)$ corre-

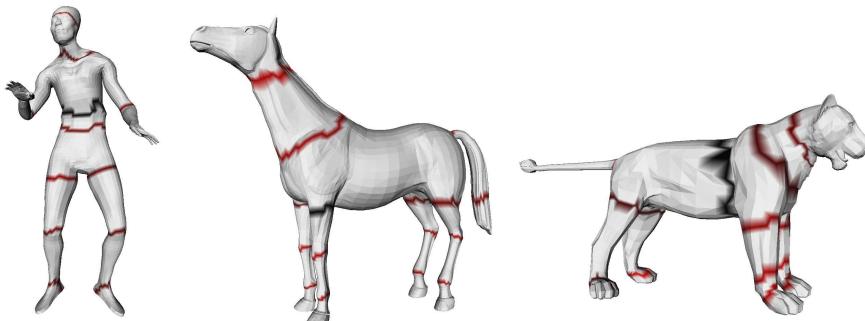


Figure 8.11 – *Elastic boundaries for several dynamic triangulated surfaces (reported on the first frame).*

sponding to saddle vertices. The related contours are self-intersecting closed curves and are referred to as *topological boundaries*. Like in the case of static shape decomposition, these boundaries delimit the protrusions of the shape.

Elastic boundaries

To identify the surface portions which are affected by elastic transformations, and thus corresponding to plausible articulations, we use a variant of the contour constriction algorithm presented in section 6.2.2 (page 135). In particular, we use exactly the same approach at the exception that the curvature term in the geometrical measurement function g is substituted by the quadratic edge-length deviation \mathcal{L}_d .

Then, for each edge \mathcal{E}_i of $\mathcal{R}(f)$, we identify the contours that maximize the \mathcal{L}_d based geometrical measurement as *elastic boundaries* and are added to the set B of segment boundaries.

Figure 8.11 shows examples of *elastic boundaries* computed on several dynamic triangulated surfaces and reported on their first frame. Notice that the elastic boundaries of the horse sequence are indeed located in the surface regions of highest quadratic edge-length deviation (see figure 8.10). From a functional point of view, these boundaries indeed correspond to the articulations of the horse highlighted by the animated sequence.

Segmentation abstract graph

The set B of segment boundaries is a set of closed curves (*elastic boundaries*) and of self-intersecting closed curves (*topological boundaries*). These

curves partition the input triangulated surface \mathcal{S} into a set of connected components.

Let \mathcal{G} be the *segmentation abstract graph* of \mathcal{S} , whose nodes depict the connected components delimited by the set B of segment boundaries and whose edges correspond to the adjacency relations between them.

In the framework of time-varying 3D shape analysis (which is a research perspective of this thesis), we will not go further in the reasoning on the enhanced topological skeleton and discuss the properties of the *segmentation abstract graph* obtained at this stage by our motion analysis strategy.

8.3.4 Early experiments

The computed *segmentation abstract graph* \mathcal{G} can be seen as the *kinematic skeleton* of the dynamic 3D shape. Indeed, kinematics is a branch of dynamics which studies the motion of objects without considering the physical forces implying the motion. de Aguiar et al. (dATTS08) thus use the term of *kinematic skeleton* in the sense that such a skeleton provides the necessary information about the understanding of the 3D shape motion, particularly precising its articulation points.

Figure 8.12 shows example of dynamic triangulated surfaces and the corresponding segmentation abstract graphs computed with our dynamic shape motion analysis approach. In particular, the segmentation abstract graph \mathcal{G} has been embedded using the following strategy.

The equivalence classes of f critical vertices have been contracted to a blue point, located at the barycenter of the related contour. *Elastic boundaries* have been contracted to *elastic nodes* (in red in figure 8.12). Finally, the connected components delimited by the segment boundaries B have been represented with black edges, linking the nodes of their related boundaries.

In figure 8.12, we can notice that *elastic nodes* are located at plausible articulations of the object, with regard to the input animated sequence. Figure 8.13 presents a visual comparison between the functional decompositions obtained with the algorithms presented in (LWCo06) and (dATTS08) and ours. Notice that with the first algorithm (left), the hooves of the horse are not extracted as individual segments while their motion is different from that of the rest of the leg. On the contrary, our method (right) identifies these articulations and provides a decomposition compatible with that provided by de Aguiar's algorithm (middle). Table 8.2 provides the com-

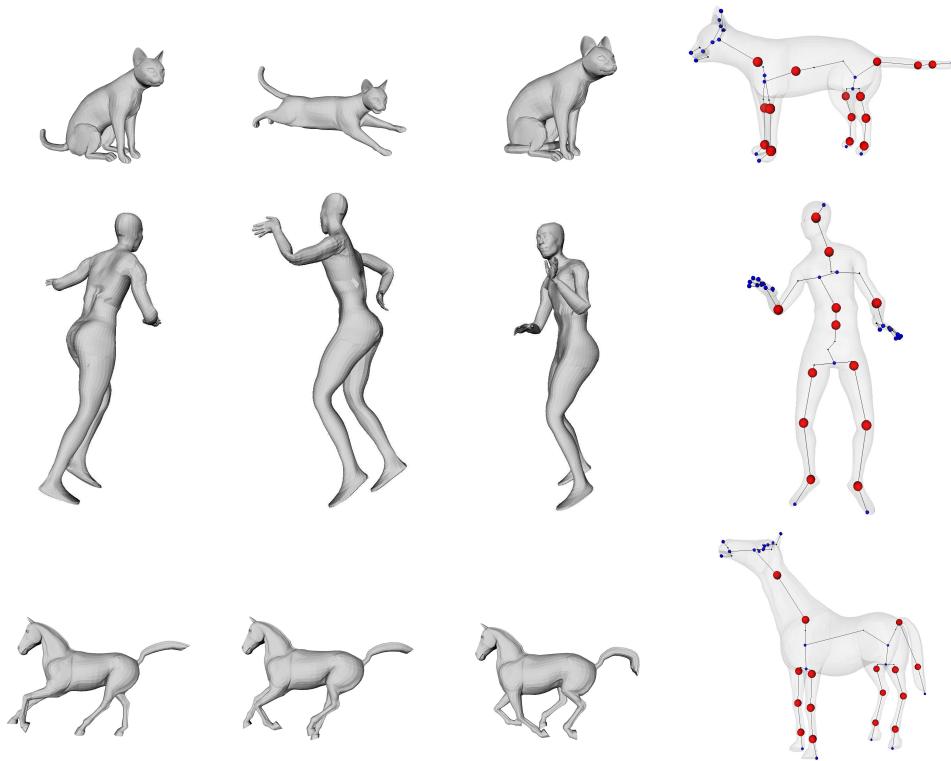


Figure 8.12 – Examples of dynamic triangulated surfaces and the corresponding segmentation abstract graphs.

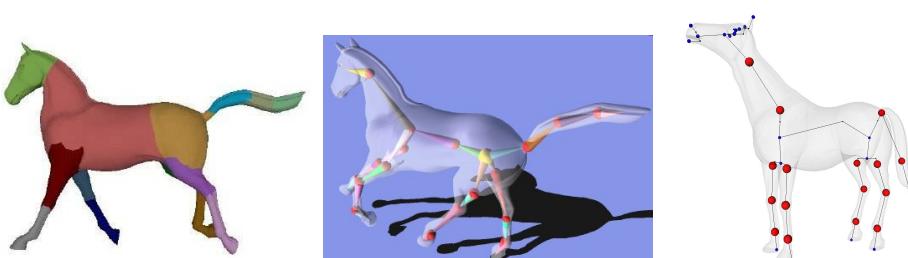


Figure 8.13 – Comparison with the functional dynamic 3D shape decomposition of (LWCo6) (left) and of (dATTS08) (middle).

Algorithms	Time (s.)
Our algorithm	12.7
de Aguiar's algorithm (dATTS08)	21
Lee's algorithm (LWCo6)	806

Table 8.2 – Computation times for the horse dynamic 3D shape decomposition.

Dynamic triangulated surface	Triangles	Frames	Time (s.)
Cat	14410	10	9.6
Dancer	14118	201	21.2
Horse	16843	49	12.7
Lion	9996	10	6.0
Snake	18354	134	20.2

Table 8.3 – Computation times for several dynamic triangulated surface decompositions.

putation time for the horse dynamic 3D shape decomposition with our algorithm (with the experimental setup described in section 5.3.5, page 120) in comparison to state-of-the-art techniques.

As our motion analysis strategy is not based on affine transformation prediction, our algorithm runs faster than state-of-the-art solutions. Table 8.3 provides additional computation times for the decomposition of other dynamic triangulated surfaces. In particular, quadratic edge-length deviation computation runs in linear time and requires $O(N_V \cdot N_F)$ steps where N_V is the number of vertices in \mathcal{S} and N_F is the number of frames in the animated sequence. The overall time complexity is bounded by that of enhanced topological skeleton computation, which requires $O(N_V^2)$ steps (the Reeb graph is only computed for the first frame).

In order to give a quantitative evaluation of the efficiency of our approach, we propose the following experiment, summarized in figure 8.14. Given a skeleton-driven animation of a static 3D shape (realized with an animation software), we propose to retrieve the kinematic model of the animation by our decomposition strategy. Figure 8.14 shows the original designed skeleton (top row, left) and the resulting animation (top row). Quadratic edge-length deviation is reported with a color gradient (bottom row). Notice that the index and the middle fingers have been slightly articulated but still exhibit edge-length deviation. Finally, the decomposition computed with the proposed approach appears on the bottom row (right).

Notice that each joint of the original skeleton has been retrieved in

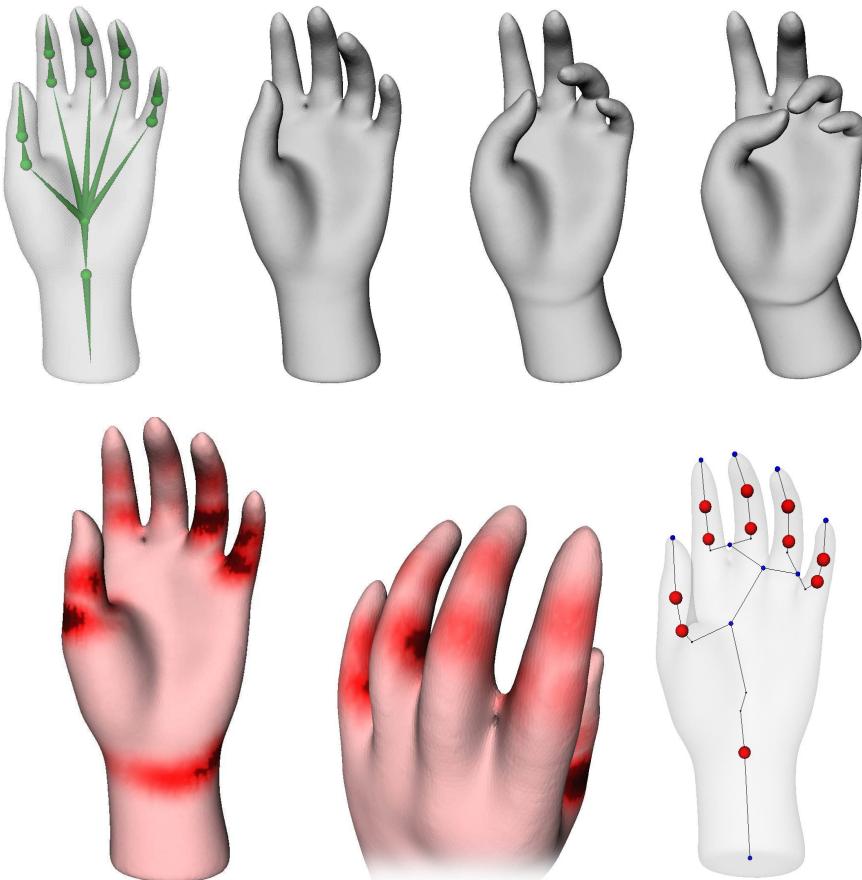


Figure 8.14 – *Animation reverse engineering: a dynamic triangulated surface is generated (top row) from a predefined skeleton of a static mesh (top row, left). Edge-length deviation (bottom row) and resulting decomposition (bottom row, right). Elastic node location error: 0.44%.*

the output decomposition. Moreover, the average height (y) difference between the original skeleton's joints and the extracted skeleton's elastic nodes is only **0.44%** of the height of the object bounding box, which seems to be a satisfactory precision. For comparison, de Aguiar's algorithm achieves a precision of 1.7% on similar experiments.

8.4 CHAPTER CONCLUSION

In this chapter, we investigated the use of enhanced topological skeleton based 3D shape modeling for recovering a functional decomposition of a 3D object for shape understanding.

On the first hand, following state-of-the-art approaches, we exploited heuristics inspired from human perception theory (Bie87) to decompose the triangulated surface representing the input 3D shape into plausible functional parts. In particular, we moved the automatic 3D mesh seg-

mentation problem to a more abstract level by reasoning directly on the enhanced topological skeleton and showed that heuristics were easily integrable at this level. We qualitatively evaluated the pertinence of the proposed technique according to the evaluation criteria enumerated in a recent survey (AKM*06). By comparing to state-of-the-art techniques, we showed that topology considerations were an interesting bias for recovering the global structure of the shape. Then, thanks to the enhanced topological skeleton data-structures, we refined the decomposition with curvature considerations. Obtained decompositions are compatible with expected segmentation visual properties (AKM*06), are robust to a variety of surface degradations and are computed with low computational effort (in comparison to state-of-the-art techniques). Finally, we also showed that integrating topology considerations in the definition of the decomposition hierarchy better depicted the functional object's layout than previous approaches and that the decomposition hierarchy was compatible when dealing with visually similar distinct objects.

On the other hand, we also proposed to exploit the temporal information provided by dynamic meshes to decompose the 3D shape into parts of uniform motion for recovering its functional layout. We proposed a new simple and efficient technique for motion characterization for the computation of segment boundaries. We provided early results where the approach was evaluated from both a qualitative and quantitative point of view, showing its pertinence with regard to existing techniques.

Along this chapter, we systematically tried to evaluate the pertinence of the proposed techniques either from a qualitative or (when possible) quantitative point of view. These experiments underlined the cruel lack of quantitative experimentation protocol for the evaluation of automatic *high-level* segmentation techniques (like it exists in the framework of shape retrieval, as described in the previous chapter). In the context of 2D image segmentation, the research community already provided such quantitative experimentation protocols based on segmentation ground-truth. We believe that in the future, the definition of 3D segmentation ground-truth data is a necessary step for a significant development of automatic 3D segmentation techniques. This statement is the main motivation of a new research project started in 2008 with my advisors in partnership with the LIRIS laboratory in Lyon and the INRIA institute in Grenoble, called MADRAS and partially funded by the French National Research Agency (ANR grant 07-MDCO-015). By defining user segmentation ground-truth data and perception based similarity measures between 3D shape decom-

positions, this project opens new research perspectives for the automatic functional decomposition of 3D shapes, enabling the design of new approaches integrating machine-learning techniques, so as to really learn and understand the semantic of a shape.

8.5 FRENCH CHAPTER ABSTRACT

Dans ce chapitre, nous avons exploré l'utilité de la modélisation de forme 3D par squelette topologique avancé pour retrouver une décomposition fonctionnelle d'un objet 3D dans le cadre de la compréhension de formes.

Tout d'abord, en suivant les approches de l'état de l'art, nous avons exploité des heuristiques inspirées de la théorie de la perception humaine (Bie87) pour décomposer une surface triangulée en parties fonctionnelles plausibles. En particulier, nous avons déplacé le problème de la segmentation automatique de surfaces maillées à un niveau plus abstrait en raisonnant directement sur le squelette topologique et nous avons montré que les heuristiques étaient facilement intégrable à ce niveau. Nous avons évalué qualitativement la pertinence de la technique proposée selon les critères d'évaluation énumérés dans un état de l'art récent (AKM*06). Par comparaison avec les approches précédentes, nous avons montré que la topologie étaient un biais intéressant pour retrouver la structure globale d'une forme. Ensuite, grâce aux structures de données associées aux squelettes topologiques avancés, nous avons affiné la décomposition avec des considérations sur les courbures de la surface. Les décompositions obtenues sont compatibles avec les propriétés visuelles attendues (AKM*06), sont robustes à une variété de dégradations de surface et leur calcul requiert un faible effort de calcul (par comparaison aux techniques de l'état de l'art).

Ensuite, nous avons également proposé d'exploiter l'information temporelle fournie par des maillages dynamiques pour décomposer une forme 3D en segments de mouvement homogène, afin d'extraire sa structure fonctionnelle. Nous avons proposé une technique simple et efficace pour la caractérisation du mouvement, utilisée pour délimiter les segments. Nous avons présenté des résultats préliminaires où l'approche a été évaluée qualitativement et quantitativement, mettant en évidence sa pertinence au regard des quelques techniques existantes.

Au cours de ce chapitre, nous avons systématiquement essayé d'évaluer la pertinence des techniques proposées, que ce soit qualitativement ou (si possible) quantitativement. Ces expériences ont mis en

évidence le manque notoire de protocole d'évaluation quantitatif des techniques de segmentation de surfaces maillées. Dans le cadre de la segmentation d'images 2D, la communauté de recherche a déjà fourni des protocoles d'expérimentation quantitative basée sur des vérités terrain. Nous pensons qu'à l'avenir, la définition de vérités terrain pour la segmentation 3D est une étape nécessaire pour le développement significatif des méthodes de segmentation. Cette observation est la principale motivation d'un nouveau projet de recherche lancé en 2008 avec mes encadrants, en collaboration avec le laboratoire LIRIS de Lyon l'INRIA de Grenoble, nommé MADRAS et financé par l'Agence Nationale de la Recherche (projet 07-MDCO-015). En définissant des vérités terrain de segmentation et des mesures de similarités entre segmentations, ce projet ouvre de nouvelles perspectives de recherche pour la décomposition fonctionnelle automatique de formes 3D, permettant de concevoir de nouvelles approches intégrant par exemple des techniques d'apprentissage automatique, pour ainsi réellement apprendre et comprendre la sémantique d'une forme.

CONCLUSION

In this thesis, we explored the use of differential topology for the intrinsic and expressive modeling of 3D shapes represented by triangulated surfaces and proposed solutions to the control skeleton extraction for shape pose edition, partial 3D shape retrieval and semantic-oriented segmentation problems.

First, we introduced in chapter 2 the notion of 3D shape from a computer scientist's point of view. In particular, we focused on boundary surface discrete approximations by surface meshes and detailed the technical reasons of their widespread use in 3D related applications. We also showed that this surface representation is extrinsic and suffers from high variability towards shape-preserving transformations, motivating research for more sophisticated shape modeling techniques.

Second, in chapter 3, we reviewed recent techniques for intrinsic and expressive 3D shape modeling. We divided this survey into two main categories, geometry and topology related techniques, to underline the complementarity of these two notions. In particular, we showed that geometry based modeling techniques have intrinsic limitations related to topology and that topology based modeling could help in their overcoming. We finally focused on differential topology techniques, especially using Reeb graphs, whose theoretical foundations have been developed in chapter 4, concentrating theoretical preliminaries. From a theoretical point of view, Reeb graphs indeed appear as an interesting shape abstraction, capturing efficiently its topology and enabling a full topology control for further geometry modeling.

Then, we presented the contributions of this thesis as follows.

9.1 SUMMARY OF CONTRIBUTIONS

First, in chapter 5, we developed a formalization of Reeb graphs for triangulated surfaces, by focusing on the notion of discrete contour. Then

we proposed a new algorithm for the computation of Reeb graphs which provides output data-structures more compliant with their definition in the smooth setting than previous approaches. We showed that, thanks to the combinatorial properties of simple PL Morse functions (which are compatible with those of smooth Morse functions), computed Reeb graphs preserved the topological properties of their smooth counterparts. Moreover, the output data-structure, referred to as *enhanced topological skeleton*, also keeps track of the surface geometry by storing explicitly the contours corresponding to each equivalence class of the discrete Reeb graph. Such a surface representation enables the formulation of geometrical calculus on Reeb graphs, largely used in the resolution of the research problems addressed later on.

Moreover, to guarantee the invariance of the representation towards shape-preserving transformations, its robustness towards surface degradations and its expressive properties, we proposed an algorithm for the computation of an intrinsic, stable and expressive simple PL Morse function. The solution we presented is based on feature vertex extraction, geodesic distance computation, minimum cancellation, global perturbation and degenerate critical vertex unfolding. The definition of this simple PL Morse function enables the construction of intrinsic Reeb graphs, whose structure captures the most important geometrical and perceptual features of the shape.

To assess the validity of the presented 3D shape modeling technique, we showed its efficiency in the resolution of three research problems related to Computer Graphics and Computer Vision, discussed by increasing level of abstraction.

Control skeleton extraction for shape-pose edition (chapter 6)

This problem deals with the automatic extraction of shape control skeletons defined such that the locations of the joints and the segments of the skeletons enable intuitive user shape handling for pose edition.

We took advantage of the enhanced version of Reeb graphs we formerly introduced (where each equivalence class is equipped with its corresponding contour) to define geometrical calculus on the Reeb graph, both for the location computation of the segments (by contour contraction) and of the joints (by contour constriction computation).

We discussed the qualitative properties of the computed shape control skeletons with regard to criteria enumerated in a recent state-of-the-art

(WML^{*06) and showed the validity of the approach in a skeleton driven shape articulation application, where the user intuitively articulate an input shape by handling directly its articulated skeleton.}

Partial 3D shape retrieval (chapter 7)

Partial 3D shape retrieval engines are systems which are expected to retrieve, in interactive processing times, the entries of a 3D shape collection which share visually similar sub-parts with an input query shape, even if they visually differ globally. This problem has recently focused the shape modeling community's interest because of its potential to provide a quantitative evaluation of a shape modeling approach and of its difficulty with regard to the canonical global shape similarity estimation problem.

We took advantage of the topological properties of our Reeb graph representation to improve both the 3D shape description and comparison steps. We introduced the new notions of *Reeb charts* and *Reeb patterns* and provided results on their topological properties. This topology control enabled the definition of intrinsic geometry signatures based on surface parameterization and the definition of a sub-graph matching technique which is robust to Reeb graphs' specific distortion.

Extensive experiments showed the superiority of our approach on the SHREC 2007 partial shape retrieval benchmark (MPB07) with regards to the state-of-the-art methods competing in this contest, assessing the validity of enhanced topological skeleton based 3D shape modeling.

Semantic-oriented segmentation (chapter 8)

We proposed two automatic techniques based on enhanced topological skeleton 3D shape modeling for the functional decomposition of 3D shapes represented by triangulated surfaces. By reasoning directly on the enhanced topological skeleton, we moved the automatic semantic-oriented segmentation problem to a more abstract level. First, we exploited heuristics based on human perception theory as decomposition constraints (mainly involving curvature). We discussed the qualitative properties of the obtained decompositions with regard to the evaluation criteria enumerated in a recent segmentation survey (AKM^{*06) and showed differential topology brought a pertinent bias in this task. Second, we exploited the temporal information provided by time-varying triangulated surfaces, *dynamic meshes*, to decompose the shape into parts of uniform motion along the animated sequence to retrieve a functional lay-}

out of the shape. We provided early results where this technique was evaluated from both a qualitative and quantitative point of view, showing its pertinence with regard to the few existing techniques.

For each of these problems, we followed the same shape modeling strategy, consisting in computing the feature vertices of the input triangulated surface, in computing a simple PL Morse function based on geodesic distances to the feature vertices and in computing the corresponding contour-enhanced Reeb graph, referred to as enhanced topological skeleton.

For each of these problems, we underlined the interest of Reeb graph based modeling for their resolution. Thanks to the invariance, the stability and the expressive properties of the simple PL Morse function we proposed, 3D shapes have been intrinsically abstracted in the form of a graph having strong properties on the degree of its nodes. In this strategy, geometry problems have been abstracted to graph problems, already widely studied in Computer Science.

Moreover, we showed the Reeb graph computation algorithm we proposed was more compliant with the Reeb graph definition in the smooth setting than previous approaches. We specifically took advantage of this surface geometry capture through the explicit encoding of equivalence classes' contours to bring new solutions to the addressed problems.

In particular, we introduced geometrical calculus on the Reeb graphs to compute contour contraction and contour constrictions for control skeleton positioning and semantic-oriented segmentation.

In the context of shape comparison, we showed Reeb graphs provided a full topology control of the input triangulated surface and enabled to overcome the topology related issues of parameterization based geometry modeling techniques.

9.2 OPEN PROBLEMS AND DIRECTIONS

In this thesis, we presented new solutions based on enhanced topological skeletons to three research problems and assessed their validity with regard to the state-of-the-art techniques.

Towards anatomically correct skeletons

We proposed a pure geometry and topology based solution to the control skeleton for shape pose edition problem. We believe that kind of solution could be improved to guarantee the validity of the shape animation from an anatomical point of view in the case of 3D shapes representing vertebrate characters. In this sense, a pioneer work has been recently proposed by Aujay et al. (AHLD07) where anatomy considerations were integrated in an interactive skeleton extraction technique. We believe that such an approach can be extended and automated in a framework adjusting by Reeb graph driven geometry optimisation the localization of pre-defined anatomically correct skeleton models.

Towards topology independent unbiased geometry comparison

In the context of shape comparison, we introduced a topology based bias for partial similarity estimation. Even if this bias turned out to provide good performances relatively to the pre-defined ground-truth, unbiased solutions for partial similarity estimation (based on surface parameterization for example) are worth being investigated. The main actual bottleneck is the computational effort required to achieve local comparison. In this framework, sophisticated local search techniques will have to be proposed to tackle this issue.

Moreover, to provide topology independent geometry characterization, we introduced a divide and conquer strategy based on Reeb graphs, where the input triangulated surface was decomposed into a set of charts of controlled topology. Then, specific parameterization strategies were applied for geometry characterization. This decomposition step introduces a (relevant) bias in the process. In order to reach unbiased solutions for topology independant geometry characterization, level set approaches (Set99) or topology oblivious cross parameterization techniques are worth being investigated. In this sense, a pioneer work has recently been proposed by Bennett et al. (BPJ07), who presented a solution based on topology control and edition by Reeb graphs for the cross parameterization of surfaces of possibly different topology. We believe that these techniques should be deeply studied and automated to significantly extend current purely geometry based solutions for shape comparisons.

Towards shape semantic machine learning

In the context of automatic semantic-oriented decomposition of 3D shapes, we first proposed a solution based on enhanced topological skeletons and human perception theory heuristics. We discussed the qualitative properties of the resulting decompositions but also evoked the lack of quantitative evaluation protocol based on ground-truth definitions. We believe that in the future, the definition of segmentation ground-truth(s) is a necessary step for a significant development of segmentation solutions. In this sense, given a segmentation ground-truth, machine-learning techniques are worth being studied in order to really learn and understand the *semantic* of a 3D shape.

Towards a unified topological and geometrical understanding of dynamic shapes

Moreover, we also proposed a solution based on the analysis of time-varying 3D data and showed it provided useful information for shape functional decomposition.

From a more fundamental point of view, we believe that the modeling of time-varying 3D shapes is definitely the next challenge to be addressed by the shape modeling community. In particular, in the framework of Riemannian geometry, time-varying shapes can be efficiently described as paths in shape space. Consequently, nice problems can be envisaged in this framework, like animation recognition (for security, accessibility or virtual reality applications), learning, compression, transfer, edition, etc. Once again, due to the possible topology transitions of a shape along the time, we believe that Reeb graphs have an important role to play in topology control for the time varying 3D shape modeling techniques that will be proposed in the future.

9.3 FRENCH CONCLUSION

Dans cette thèse, nous avons exploré l'utilisation de la topologie différentielle pour la modélisation intrinsèque et expressive de formes 3D représentées par des surfaces triangulées et nous avons proposé des solutions pour les problèmes d'extraction de squelette de contrôle pour l'édition de pose d'objets déformables, pour l'indexation de formes 3D par similarité partielle et pour la segmentation orientée sémantique.

Premièrement, nous avons introduit dans le chapitre 2 la notion de forme 3D du point de vue informatique. En particulier, nous nous sommes focalisés sur les approximations discrètes de surfaces par surfaces maillées et nous avons détaillé les raisons techniques expliquant leur utilisation largement répandue dans les applications 3D. Nous avons également montré que cette représentation est extrinsèque et souffre de grande variabilité face aux transformations préservant les formes, motivant la recherche de techniques de modélisation de forme plus sophistiquées.

Deuxièmement, dans le chapitre 3, nous avons passé en revue les techniques récentes pour la modélisation intrinsèque et expressive de forme 3D. Nous avons divisé cette étude en deux catégories principales, les techniques basées sur la géométrie et celle sur la topologie, pour souligner la complémentarité de ces deux notions pour la modélisation. En particulier, nous avons montré que les approches géométriques ont des limitations propres liées à des considérations de topologie et que la modélisation topologique pouvait contribuer à les surmonter. Enfin, nous nous sommes focalisés sur les techniques basées sur la topologie différentielle, en particulier sur les graphes de Reeb, dont les fondements théoriques ont été développés dans le chapitre 4, regroupant tous les préliminaires théoriques. D'un point de vue théorique, les graphes de Reeb apparaissent effectivement comme une abstraction de forme intéressante, qui capture efficacement sa topologie et qui permet un contrôle topologique pour d'éventuelles traitements géométriques ultérieurs.

Ensuite, nous avons présenté les contributions de cette thèse comme suit.

9.3.1 Résumé des contributions

Tout d'abord, dans le chapitre 5, nous avons développé une formalisation des graphes de Reeb pour les surfaces triangulées, en se focalisant sur la notion de contour discret. Ensuite, nous avons proposé un nouvel algorithme de calcul de graphes de Reeb, qui fournit en sortie des struc-

tures de données plus compatibles avec leur définition dans le continu que les approches précédentes. Nous avons montré que, grâce aux propriétés combinatoires des fonctions de Morse simples linéaires par morceaux (qui sont compatibles avec celles des fonctions de Morse dans le continu), les graphes de Reeb calculés préservent les propriétés topologiques de leurs homologues continus. De plus, les structures de données en sortie, désignées par le terme de *squelettes topologiques avancés*, rendent également compte de la géométrie de la surface par la sauvegarde explicite des contours correspondant à chaque classe d'équivalence des graphes de Reeb discrets. Une telle représentation de surface permet la formulation de calcul géométrique sur les graphes de Reeb, largement utilisé dans la résolution des problèmes de recherche abordés ensuite.

Par ailleurs, pour garantir l'invariance de la représentation face aux transformations préservant la forme, sa robustesse face aux dégradations de surface et son expressivité, nous avons proposé un algorithme pour le calcul d'une fonction de Morse simple linéaire par morceaux intrinsèque, stable et expressive. La solution que nous avons présentée est basée sur l'extraction de sommet caractéristique, le calcul de distances géodésiques, l'annulation de minima, la perturbation globale et le dépliage de sommets critiques dégénérés. La définition de cette fonction de Morse simple linéaire par morceaux permet la construction de graphes de Reeb intrinsèques, dont la structure capture les composantes géométriques les plus importantes de la forme.

Pour témoigner de la validité de la technique de modélisation 3D présentée, nous avons montré son efficacité dans la résolution de trois problèmes de recherche lié à l'Informatique Graphique et la Vision par Ordinateur, abordés par niveau d'abstraction croissant.

Extraction de squelettes de contrôle pour l'édition de pose (chapitre 6)

Ce problème traite de l'extraction automatique de squelettes de contrôle d'objets 3D déformables, définis de telle sorte que les positions des articulations et des segments des squelettes permettent une manipulation intuitive et interactive de la pose de l'objet par un utilisateur.

Nous avons tiré parti de la version avancée des graphes de Reeb que nous avions précédemment introduite (où chaque classe d'équivalence est accompagnée du contour correspondant) pour définir la notion de calcul géométrique sur le graphe de Reeb, à la fois pour le calcul de localisation

des segments (par contraction de contour) et des articulations (par calcul de constrictions de contour).

Nous avons discuté des propriétés qualitatives des squelettes de contrôle calculés au regard des critères énumérés dans l'état de l'art de Wu et al. (WML*06) et nous avons montré la validité de notre approche par une application à la déformation d'objets 3D guidée par squelette, où l'utilisateur articule intuitivement une forme en manipulant directement le squelette.

Indexation 3D par similarité partielle (chapitre 7)

Les moteurs de recherche 3D par similarité partielle sont des systèmes devant retrouver, dans des temps d'exécution interactifs, les entrées d'une collection de formes 3D qui partagent des sous-parties visuellement similaires à celles d'une forme requête, ce même si les formes diffèrent globalement. Ce problème a récemment attiré l'attention de la communauté de recherche en modélisation 3D, étant donné son potentiel à fournir des évaluations quantitatives des approches de modélisation et sa difficulté intrinsèque vis à vis du problème traditionnel d'estimation de similarité globale.

Nous avons tiré parti des propriétés topologiques de notre représentation par graphes de Reeb pour améliorer à la fois les étapes de description et de comparaison de formes 3D. Nous avons introduit les nouvelles notions de *cartes de Reeb* et de *motifs de Reeb* et nous avons fourni des résultats sur leurs propriétés topologiques respectives. Ce contrôle topologique nous a permis de définir des signatures de géométrie intrinsèques basées sur la paramétrisation de surface ainsi qu'une technique de mise en correspondance de sous-graphes robuste à la distorsion spécifique des graphes de Reeb.

De nombreuses expériences ont montré la supériorité de notre approche sur le *benchmark* d'indexation par similarité partielle de SHREC 2007 (MPB07) face aux méthodes de l'état de l'art participant à ce concours, ce qui témoigne de la validité de la modélisation de formes 3D par squelettes topologiques avancés.

Segmentation orientée sémantique (chapitre 8)

Nous avons proposé deux techniques automatiques basées sur les squelettes topologiques avancés pour la décomposition fonctionnelle de formes 3D représentées par des surfaces. En raisonnant directement

sur le squelette topologique avancé, nous avons déplacé le problème de la segmentation automatique orientée sémantique à un niveau plus abstrait. Tout d'abord, nous avons exploité des heuristiques basées sur la théorie de la perception humaine comme contraintes de décomposition (impliquant principalement la courbure). Nous avons discuté les propriétés qualitatives des décompositions obtenues au travers des critères d'évaluation énumérés dans un état de l'art récent (AKM*06) et nous avons montré que la topologie différentielle apportait un biais pertinent dans cette tâche. Ensuite, nous avons exploité l'information temporelle fournie par des surfaces triangulées variant dans le temps pour décomposer la forme en sous-parties de mouvement uniforme au cours de la séquence animée, pour retrouver la structure fonctionnelle de la forme. Nous avons également fourni des résultats préliminaires où cette technique a été évaluée à la fois qualitativement et quantitativement, montrant sa pertinence vis à vis des quelques méthodes existantes.

Pour chacun de ces problèmes, nous avons suivi la même approche de modélisation, consistant à calculer les sommets représentatifs de la surface triangulée en entrée, à calculer la fonction de Morse simple linéaire par morceaux basée sur l'évaluation de distances géodésiques aux sommets caractéristiques et à calculer la version avancée (enrichies par les contours) du graphe de Reeb.

Pour chacun de ces problèmes, nous avons mis en évidence l'intérêt de la modélisation par graphe pour leur résolution. Grâce à l'invariance, la stabilité et l'expressivité de la fonction de Morse simple linéaire par morceaux que nous avons introduite, les formes 3D sont intrinsèquement abstraite sous la forme d'un graphe. Dans cette stratégie, les problèmes de géométrie sont alors abstraits en problèmes sur les graphes, largement étudiés en Informatique.

De plus, nous avons montré que l'algorithme de calcul de graphe de Reeb que nous avons proposé était plus fidèle à la définition des graphes de Reeb dans le continu que ceux des approches précédentes. Nous avons spécifiquement tiré parti de cette capture de la géométrie de la surface par l'encodage explicite des contours associés aux classes d'équivalence pour apporter de nouvelles solutions aux problèmes abordés.

En particulier, nous avons introduit la notion de calcul géométrique sur le graphe de Reeb pour calculer les contractions et les constrictions de contour pour le positionnement du squelette de contrôle et pour la segmentation orientée sémantique.

Dans le contexte de la comparaison de formes, nous avons montré que

les graphes de Reeb fournissaient un contrôle complet sur la topologie de la surface triangulée en entrée et qu'ils permettaient de surmonter les limitations liées à la topologie des techniques de modélisation géométrique basée sur la paramétrisation.

9.3.2 Problèmes ouverts et directions de recherche

Dans cette thèse, nous avons présenté de nouvelles solutions basées sur les squelettes topologiques avancés pour trois problèmes de recherche et nous avons montré leur validité par rapport aux techniques de l'état de l'art.

Vers des squelettes anatomiquement corrects

Nous avons proposé une solution purement basée sur la géométrie et la topologie pour le problème d'extraction de squelettes de contrôle pour l'articulation de forme. Nous pensons que ce type de solution peut être amélioré pour garantir la validité de l'articulation d'un point de vue anatomique dans le cas de formes 3D représentant des vertébrés. Dans ce sens, un travail novateur a récemment été proposé par Aujay et al. (AHLDo7) où des considérations anatomiques ont été intégrées dans une technique interactive d'extraction de squelette. Nous pensons qu'une telle approche peut être étendue et automatisée en ajustant à la forme le positionnement de squelettes anatomiques correctes par un procédé d'optimisation géométrique basé sur les graphes de Reeb.

Vers des techniques de comparaison géométrique indépendantes de la topologie

Dans le contexte de la comparaison de forme, nous avons introduit un biais basé sur la topologie pour l'estimation de similarité partielle. Même si ce biais conduit à de bonnes performances relativement à la vérité terrain prédéfinie, des solutions non biaisées (basées sur la paramétrisation de surface par exemple) méritent d'être étudiées. En terme de calcul, le principal goulot d'étranglement est l'effort requis pour les comparaisons locales. Dans ce cadre, des considérations sur la topologie pourraient être utiles pour le développement de solutions basées sur la géométrie mais néanmoins, des techniques sophistiquées pour la comparaison locale devront être proposées pour réduire le temps de calcul.

De plus, pour caractériser la géométrie des formes indépendamment de leur topologie, nous avons introduit une stratégie "*diviser pour mieux*

régner" basée sur les graphes de Reeb, où la surface triangulée en entrée est d'abord décomposée en cartes de topologie contrôlée. Ensuite, des stratégies de paramétrisation spécifiques sont appliquées pour la caractérisation de la géométrie. L'étape de décomposition introduit un biais (pertinent) dans ce processus. Pour atteindre des solutions non biaisées, les techniques de paramétrisation croisée méritent d'être approfondies. Dans ce sens, un travail novateur a récemment été proposé par Bennett et al. (BPJ07), qui ont présenté une solution basée sur le contrôle et l'édition de topologie par graphes de Reeb pour la paramétrisation croisée entre surfaces de topologie éventuellement distincte. Nous pensons que ces techniques devraient être profondément étudiées et automatisées pour étendre de manière significative les solutions pour la comparaison de forme basées sur la modélisation géométrique.

Vers l'apprentissage automatique de la sémantique des formes

Dans le contexte de la décomposition de formes 3D automatique orientée sémantique, nous avons d'abord proposé une solution basée sur les squelettes topologiques avancés et sur des heuristiques inspirées de la théorie de la perception humaine. Nous avons discuté des propriétés qualitatives des décompositions résultantes mais nous avons également évoqué le manque de protocole d'évaluation quantitative basé sur la définition de vérité terrain. Nous pensons qu'à l'avenir, la définition de vérité(s) terrain pour la segmentation est une étape nécessaire pour un développement significatif des solutions de segmentation. Dans ce sens, étant donné une vérité terrain, les techniques d'apprentissage automatique méritent d'être étudiées afin de réellement apprendre et comprendre la sémantique d'une forme 3D.

Vers la compréhension unifiée topologique-géométrique des formes dynamiques

De plus, nous avons également proposé une solution de décomposition basée sur l'analyse de données 3D variant dans le temps et nous avons montré que ces données apportaient une information utile pour la décomposition fonctionnelle de la forme.

D'un point de vue plus fondamental, nous pensons que la modélisation de formes 3D variant dans le temps est résolument le prochain défi scientifique auquel la communauté de recherche en modélisation 3D devra faire face. En particulier, dans le cadre de la géométrie Riemannienne,

les formes variant dans le temps peuvent être efficacement représentées par des chemins dans l'espace des formes. Par conséquent, de beaux problèmes peuvent être envisagés avec cette approche, comme la reconnaissance (pour des applications de sécurité, d'accessibilité ou de réalité virtuelle), l'apprentissage, la compression, le transfert, ou encore l'édition d'animations. Là encore, à cause d'éventuelles transitions topologiques de la forme au cours du temps, nous pensons que les graphes de Reeb ont un rôle important à jouer pour le contrôle topologique dans les futures techniques de modélisation de formes 3D variant dans le temps.

INDEX

- Bijection, 64
Boundary components, 67
Christoffel symbols, 70
Closed manifold, 67
Closed set, 63
Compact manifold, 64
Connected components, 67
Connected components of a simplicial complex, 80
Connected level set, 76
Connected manifold, 67
Connected simplicial complex, 80
Continuous function, 64
Contour, 76
Covering, 64
Critical point, 73
Critical value, 73
Critical vertex, 85
Degenerate critical vertex, 86
Differentiable manifold, 65
Discrete contour, 111
Discrete Gaussian curvature, 83
Discrete level line, 110
Discrete mean curvature, 83
Discrete principal curvatures, 83
Discrete sub-level set, 110
Edge, 79
Embedding, 66
Euler relation I, 69
Euler relation II, 81
Face, 79
Feature points, 101
Feature vertices, 101
First fundamental form, 69
Function, 64
Gaussian curvature, 71
General piecewise linear function, 84
Genus of a surface, 68
Genus of a triangulated surface, 81
Geodesic distance on a triangulated surface, 81
Geodesic on a surface, 70
Geodesic on a triangulated surface, 81
Gradient, 76
Half space, 67
Hessian matrix, 73
Homeomorphic spaces, 64
Homeomorphism, 64
Index of a non-degenerate critical point, 74
Injection, 65
Isolated critical point, 73
Level line, 76
Level manifold, 76
Level set, 76

- Link, 82
 Lower link, 84
 Manifold, 64
 Manifold with boundary, 67
 Manifold's boundary, 67
 Map, 64
 Maximum vertex, 85
 Mean curvature, 71
 Minimum vertex, 85
 Morse function, 73
 Morse number, 74
 Morse theorem, 75
 Morse-Euler relation, 74
 Non-degenerate critical point, 73
 Normal curvature, 71
 Normal vector field, 68
 One-to-one, 65
 One-to-one and onto, 64
 Open set, 63
 Orientable surface, 68
 Piecewise linear function, 84
 Piecewise linear Morse function, 86
 PL function, 84
 PL Morse function, 86
 Principal curvatures, 71
 Reeb chart, 154
 Reeb graph, 77
 Reeb graph edge, 116
 Reeb graph node, 116
 Reeb graph of a triangulated surface, 111
 Reeb pattern, 161
 Reeb pattern structural signature, 163
 Regular point, 73
 Regular value, 73
 Regular vertex, 85
 Saddle vertex, 85
 Second fundamental form, 70
 Simple Morse function, 77
 Simple saddle, 85
 Simplex, 79
 Simplicial complex, 79
 Smooth function, 65
 Smooth manifold, 65
 Star, 82
 Sub-level set, 109
 Surface, 64
 Topological space, 63
 Topology, 63
 Triangle, 79
 Triangulated surface, 80
 Triangulated surface with boundary, 80
 Triangulated surface's boundary, 80
 Triangulated surface's boundary components, 80
 Triangulation, 80
 Underlying space, 80
 Upper link, 84
 Vertex, 79

BIBLIOGRAPHY

- [ABE07] ATTALI D., BOISSONNAT J.-D., EDELSBRUNNER H.: *Mathematical Foundations of Scientific Visualization, Computer Graphics and Massive Data Exploration*. Springer, 2007, ch. Stability and computation of medial axes: a state of the art report. (Cited pages xxi and 46.)
- [Abi81] ABIKOFF W.: The uniformization theorem. *American Mathematical Monthly* 88 (1981), 574–592. (Cited pages 38 and 40.)
- [AHLD07] AUJAY G., HÉTROY F., LAZARUS F., DEPRAZ C.: Harmonic skeletons for realistic character animation. In *Eurographics Symposium on Computer Animation* (2007), pp. 151–160. (Cited pages 55, 131, 219, and 225.)
- [aimo06] AIM@SHAPE Shape Repository. <http://shapes.aim-at-shape.net/> (2006). (Cited pages 16, 21, and 120.)
- [AKM*06] ATTENE M., KATZ S., MORTARA M., PATANÉ G., SPAGNUOLO M., TAL A.: Mesh segmentation: A comparative study. In *IEEE Shape Modeling International* (2006), pp. 14–25. (Cited pages 49, 51, 52, 187, 188, 197, 212, 213, 217, and 224.)
- [ATC*08] AU O. K.-C., TAI C.-L., CHU H.-K., COHEN-OR D., LEE T.-Y.: Skeleton extraction by mesh contraction. *ACM Transactions on Graphics To appear* (2008). (Cited pages 46, 47, 48, and 132.)
- [Ban67] BANCHOFF T. F.: Critical points and curvature for embedded polyhedra. *Journal of Differential Geometry* 1 (1967), 257–268. (Cited pages 84 and 86.)
- [BB97] BLOOMENTHAL J., BAJAJ C.: *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers, 1997. (Cited page 17.)
- [BCDS94] BUSER P., CONWAY J., DOYLE P., SEMMLER K.-D.: Some planar isospectral domains. In *International Mathematics Research Notices* (1994), pp. 391–403. (Cited pages 32 and 36.)

- [BCGo8] BEN-CHEN M., GOTSMAN C.: Characterizing shape using conformal factors. In *Eurographics Workshop on 3D object retrieval* (2008), pp. 1–8. (Cited pages xx, 38, 40, 151, 152, and 157.)
- [BCGBo8] BEN-CHEN M., GOTSMAN C., BUNIN G.: Conformal flattening by curvature prescription and metric scaling. *Computer Graphics Forum* 27 (2008), 449–458. (Cited page 39.)
- [BDBPo6] BERRETI S., DEL BIMBO A., PALA P.: Partitioning of 3D meshes using Reeb graphs. In *IEEE International Conference on Pattern Recognition* (2006), pp. 19–22. (Cited page 188.)
- [BEHPO4] BREMER P.-T., EDELSBRUNNER H., HAMANN B., PASCUCCI V.: Topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics* 10 (2004), 385–396. (Cited page 54.)
- [BGSF08] BIASOTTI S., GIORGI D., SPAGNUOLO M., FALCIDIENO B.: Reeb graphs for shape analysis and applications. *Theoretical Computer Science* 392 (2008), 5–22. (Cited page 55.)
- [BHGS06] BOUBEKEUR T., HEIDRICH W., GRANIER X., SCHLICK C.: Volume-surface trees. *Computer Graphics Forum* 25 (2006), 399–406. (Cited page 22.)
- [BHK03] BEN HAMZA A., KRIM H.: Topological modeling of illuminated surfaces using reeb graph. In *IEEE International Conference on Image Processing* (2003), pp. 361–364. (Cited page 55.)
- [BI04] BRENNECKE A., ISENBERG T.: 3D shape matching using skeleton graphs. In *Simulation and Visualization* (2004), pp. 299–310. (Cited page 47.)
- [Bie87] BIEDERMAN I.: Recognition-by-components: A theory of human image understanding. *Psychological Review* 94 (1987), 115–147. (Cited pages 31, 50, 51, 187, 188, 189, 195, 211, and 213.)
- [BJ85] BESL P., JAIN R.: Intrinsic and extrinsic surface characteristics. In *IEEE Computer Vision and Pattern Recognition* (1985), pp. 226–243. (Cited page 24.)

- [BKS^{*}05] BUSTOS B., KEIM D. A., SAUPE D., SCHRECK T., VRANIC D. V.: Feature-based similarity search in 3D object databases. *ACM Computing Surveys* 37 (2005), 345–387. (Cited page 149.)
- [Bli82] BLINN J. F.: A generalization of algebraic surface drawing. *ACM Transactions on Graphics* 1 (1982), 235–256. (Cited page 17.)
- [Blo02] BLOOMENTHAL J.: Medial based vertex deformation. In *Eurographics Symposium on Computer Animation* (2002), pp. 147–151. (Cited page 48.)
- [Blu67] BLUM H.: A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form* (1967), M.I.T. Press, pp. 362–380. (Cited page 46.)
- [BM03] BOIER-MARTIN I. M.: Domain decomposition for multiresolution analysis. In *Eurographics Symposium on Geometry Processing* (2003), pp. 29–40. (Cited pages xxi, 49, 50, 52, and 187.)
- [BMK92] BESL P. J., MC KAY N. D.: A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), 239–256. (Cited page 22.)
- [BMS00] BIASOTTI S., MORTARA M., SPAGNUOLO M.: Surface compression and reconstruction using reeb graphs and shape analysis. In *Spring Conference on Computer Graphics* (2000), pp. 175–184. (Cited page 55.)
- [BMSF06] BIASOTTI S., MARINI S., SPAGNUOLO M., FALCIDIENO B.: Sub-part correspondence by structural descriptors of 3D shapes. *Computer-Aided Design Journal* 38 (2006), 1002–1019. (Cited pages xxv, 87, 150, 151, 176, 177, 178, 180, and 182.)
- [BPJo7] BENNETT J., PASCUCCI V., JOY K.: Genus oblivious cross parameterization: robust topological management of inter-surface maps. In *Pacific Graphics* (2007), pp. 238–247. (Cited pages 56, 219, and 226.)
- [BS00] BOUIX S., SIDDIQI K.: Divergence based medial surfaces. In *European Conference on Computer Vision* (2000), pp. 603–618. (Cited page 48.)

- [BTSGoo] BLANDING R. L., TURKIYYAH G. M., STORTI D. W., GANTER M. A.: Skeleton-based three-dimensional geometric morphing. *Computational Geometry* 15 (2000), 129–148. (Cited page 48.)
- [CDS*05] CORNEA N. D., DEMIRCI M. F., SILVER D., SHOKOUFANDEH A., DICKINSON S., KANTOR P. B.: 3D object retrieval using many-to-many matching of curve skeletons. In *IEEE Shape Modeling International* (2005), pp. 366–371. (Cited pages xxv, 47, 151, 176, 177, 180, and 182.)
- [CEF98] CHENG H. L., EDELSBRUNNER H., FU P.: Shape space from deformation. In *Pacific Graphics* (1998), pp. 104–113. (Cited page 41.)
- [CLR94] CORMEN T H., LEISERSON C E., RIVEST R L.: *Introduction to Algorithms*. MIT Press, 1994. (Cited page 117.)
- [CMEH*03] COLE-MCLAUGHLIN K., EDELSBRUNNER H., HARER J., NATARAJAN V., PASCUCCI V.: Loops in Reeb graphs of 2-manifolds. In *ACM Symposium on Computational Geometry* (2003), pp. 344–350. (Cited pages 55, 78, 87, 88, 95, 104, 109, 115, 117, 119, 156, and 165.)
- [CSAoo] CARR H., SNOEYLINK J., AXEN U.: Computing contour trees in all dimensions. In *ACM Symposium on Discrete Algorithms* (2000), pp. 918–926. (Cited page 87.)
- [CSADo4] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. *ACM Transactions on Graphics* 23 (2004), 905–914. (Cited pages xxi, 49, 50, 52, and 187.)
- [CSdPo4] CARR H., SNOEYINK J., DE PANNE M. V.: Simplifying flexible isosurfaces using local geometric measures. In *IEEE Visualization* (2004), pp. 497–504. (Cited pages 55 and 87.)
- [CSMo7] CORNEA N. D., SILVER D., MIN P.: Curve skeleton properties, applications and algorithms. *IEEE Transactions on Visualization and Computer Graphics* 13 (2007), 530–548. (Cited pages 46, 47, 48, 49, and 132.)
- [CSYBo5] CORNEA N. D., SILVER D., YUAN X., BALASUBRAMANIAN R.: Computing hierarchical curve-skeletons of 3D objects. *The*

- Visual Computer* 21 (2005), 945–955. (Cited pages 47, 48, and 132.)
- [CT65] COOLEY JAMES W., TUKEY JOHN W.: An algorithm for the machine calculation of complex fourier series. *Mathematical Computation* 19 (1965), 297–301. (Cited page 141.)
- [CTSOo3] CHEN D. Y., TIAN X. P., SHEN Y. T., OUHYOUNG M.: On visual similarity based 3D model retrieval. *Computer Graphics Forum* 22 (2003), 223–232. (Cited page 149.)
- [dATTS08] DE AGUIAR E., THEOBALT C., THRUN S., SEIDEL H.-P.: Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum* 27 (2008), To appear. (Cited pages xxvi, 204, 208, 209, and 210.)
- [DBG*06] DONG S., BREMER P.-T., GARLAND M., PASCUCCI V., HART J.: Spectral surface quadrangulation. *ACM Transactions on Graphics* 25 (2006), 1057–1066. (Cited pages 54 and 56.)
- [dC92] DO CARMO M. P.: *Riemannian geometry*. Birkhäuser, 1992. (Cited page 61.)
- [DG07] DONG S., GARLAND M.: Iterative methods for improving mesh parameterizations. In *IEEE Shape Modeling International* (2007), pp. 199–210. (Cited page 39.)
- [DGB03] DUPONT F., GILLES B., BASKURT A.: Lossless and scalable 3D object coding method based on medial axis transformation. In *Picture Coding Symposium* (2003), pp. 381–386. (Cited page 46.)
- [DH73] DUDA R., HART P.: *Pattern classification and scene analysis*. Wiley, 1973. (Cited pages 51 and 204.)
- [Die05] DIESTEL R.: *Graph Theory*. Springer, 2005. (Cited page 99.)
- [Dij59] DIJKSTRA E. W.: A note on two problems in connection with graphs. *Numerische Mathematik* 1 (1959), 269–271. (Cited pages 81, 99, 104, 118, and 191.)
- [DJ06] DEY T. K., JIAN S.: Defining and computing curve skeletons with medial geodesic function. In *Eurographics Symposium on Geometry Processing* (2006), pp. 143–152. (Cited pages xxi, 46, 47, and 132.)

- [DZMCo7] DYER R., ZHANG H., MÖLLER T., CLEMENTS A.: *An investigation of the spectral robustness of mesh Laplacians*. Tech. Rep. SFU-CMPT-17/07-TR2007-17, Simon Fraser University, 2007. (Cited page 36.)
- [Ede01] EDELSBRUNNER H.: *Geometry and topology for mesh generation*. Cambridge University Press, 2001. (Cited pages 61 and 71.)
- [EG79] EISENBERG M., GUY R.: A proof of the hairy ball theorem. *American Mathematical Monthly* 86 (1979), 571–574. (Cited page 54.)
- [EHZ01] EDELSBRUNNER H., HARER J., ZOMORODIAN A.: Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. In *ACM Symposium on Computational Geometry* (2001), pp. 70–79. (Cited pages 84, 86, 103, and 107.)
- [EHZ03] EDELSBRUNNER H., HARER J., ZOMORODIAN A.: Hierarchical morse-smale complexes for piecewise linear 2-manifolds. *Discrete Computational Geometry* 30 (2003), 173–192. (Cited page 54.)
- [EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003), 1285–1295. (Cited page 34.)
- [FHO4] FLOATER M. S., HORMANN K.: Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling* (2004), pp. 157–186. (Cited pages 37, 38, and 39.)
- [FK97] FOMENKO A., KUNII T.: *Topological Modeling for Visualization*. Springer, 1997. (Cited pages 61, 71, and 163.)
- [FKS*04] FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. *ACM Transactions on Graphics* 23 (2004), 652–663. (Cited pages 149, 150, and 180.)
- [FMK*03] FUNKHOUSER T., MIN P., KAZHDAN M., CHEN J., HALDERMAN A., DOBKIN D.: A search engine for 3D models. *ACM Transactions on Graphics* 22 (2003), 83–105. (Cited pages 149, 150, and 151.)

- [For98] FORMAN R.: Morse theory for cell complexes. *Advances in Mathematics* 134 (1998), 90–145. (Cited page 84.)
- [FS06] FUNKHOUSER T., SHILANE P.: Partial matching of 3D shapes with priority-driven search. In *Eurographics Symposium on Geometry Processing* (2006), pp. 131–142. (Cited page 150.)
- [gamo03] INRIA Gamma Research Group Shape Repository. <http://www-c.inria.fr/gamma/disclaimer.php> (2003). (Cited pages 16, 21, and 120.)
- [gcc08] GNU Compiler Collection. <http://gcc.gnu.org/> (2008). (Cited page 120.)
- [GCOo6] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics* 25 (2006), 130–150. (Cited page 150.)
- [GG90] GONG W., G B.: A simple parallel 3D thinning algorithm. In *IEEE International Conference on Pattern Recognition* (1990), pp. 188–190. (Cited page 46.)
- [GKo4] GIBLIN P., KIMIA B. B.: A formal classification of 3D medial axis points and their local geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (2004), 238–251. (Cited page 46.)
- [Gra71] GRAMAIN A.: *Topologie des surfaces*. Presses Universitaires de France, 1971. (Cited pages 61 and 78.)
- [GSCO07] GAL R., SHAMIR A., COHEN-OR D.: Pose oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics* 13 (2007), 261–271. (Cited page 149.)
- [gslo8] GNU Scientific Library (gsl) project. <http://www.gnu.org/software/gsl/> (2008). (Cited page 120.)
- [gtso6] GNU Triangulated Surface (gts) library project. <http://gts.sourceforge.net/> (2006). (Cited page 120.)
- [GVo4] GU X., VEMURI B. C.: Matching 3D shapes using 2D conformal representations. In *Medical Image Computing and Computer Assisted Intervention* (2004), pp. 771–780. (Cited pages xx, 37, 38, 39, 151, 152, and 157.)

- [GWC^{*}04] GU X., WANG Y., CHAN T. F., THOMPSON P. M., YAU S.-T.: Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging* 23 (2004), 949–958. (Cited page 40.)
- [GWK^{*}07] GU X., WANG S., KIM J., ZENG Y., WANG Y., QIN H., SAMARAS D.: Ricci flow for 3D shape analysis. In *IEEE International Conference on Computer Vision* (2007), pp. 1–8. (Cited page 41.)
- [GY03] GU X., YAU S.-T.: Surface classification using conformal structures. In *IEEE International Conference on Computer Vision* (2003), pp. 701–708. (Cited pages xx and 37.)
- [H05] HÉTROY F.: Constriction computation using surface curvature. In *Eurographics* (2005), pp. 1–4. (Cited pages 133, 135, 137, and 138.)
- [HA03a] HÉTROY F., ATTALI D.: From a closed piecewise geodesic to a constriction on a closed triangulated surface. In *Pacific Graphics* (2003), pp. 394–398. (Cited page 132.)
- [HA03b] HÉTROY F., ATTALI D.: Topological quadrangulations of closed triangulated surfaces using the reeb graph. *Graphical Models* 65 (2003), 131–148. (Cited page 55.)
- [Har98] HART J. C.: *Mathematical Visualization*. Springer, 1998, ch. Morse Theory for implicit surface modeling. (Cited pages xxi and 72.)
- [HGoo] HUBELI A., GROSS M.: *A Survey of Surface representations for geometric modeling*. Tech. Rep. 335, ETH Zurich, 2000. (Cited page 17.)
- [Hir76] HIRSCH M. W.: *Differential topology*. Springer-Verlag, 1976. (Cited page 54.)
- [HKDH04] HUBER D., KAPURIA A., DONAMUKKALA R., HEBERT M.: Parts-based 3D object classification. In *IEEE Computer Vision and Pattern Recognition* (2004), pp. 82–89. (Cited page 149.)
- [HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T.: Topology matching for fully automatic similarity estimation of 3D

- shapes. In *SIGGRAPH* (2001), pp. 203–212. (Cited pages 55, 87, 95, 149, 154, 178, and 188.)
- [IJL*05] IYER N., JAYANU S., LOU K., KALYANARAMAN Y., RAMANI K.: Three-dimensional shape searching: State-of-the-art review and future trends. *Computer-Aided Design Journal* 37 (2005), 509–530. (Cited page 149.)
- [Illo5] ILIC S.: *Implicit Meshes: unifying implicit and explicit surface representations for 3D reconstruction and tracking*. PhD thesis, EPFL, 2005. (Cited page 17.)
- [JH99] JOHNSON A. E., HEBERT M.: Using spin-images for efficient multiple model recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (1999), 433–449. (Cited page 150.)
- [JZo6] JAIN V., ZHANG H.: Robust 3D shape correspondence in the spectral domain. In *IEEE Shape Modeling International* (2006), pp. 118–129. (Cited page 34.)
- [JZo7] JAIN V., ZHANG H.: A spectral approach to shape-based retrieval of articulated 3D models. *Computer-Aided Design Journal* 39 (2007), 398–407. (Cited pages 34 and 149.)
- [Kac66] KAC M.: Can one hear the shape of a drum? *American Mathematical Monthly* 73 (1966), 1–23. (Cited page 32.)
- [KBHo6] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing* (2006), pp. 61–70. (Cited page 22.)
- [Ken84] KENDALL D. G.: Shape manifolds, procrustean metrics and complex projective spaces. *Bulletin of the London Mathematical Society* 16 (1984), 81–121. (Cited page 41.)
- [KGoo] KARNI Z., GOTSMAN C.: Spectral compression of mesh geometry. In *SIGGRAPH* (2000), pp. 279–286. (Cited pages 34 and 52.)
- [KLTo5] KATZ S., LEIFMAN G., TAL A.: Mesh segmentation using feature point and core extraction. *The Visual Computer* 21 (2005), 865–875. (Cited pages xxi, xxv, 50, 51, 53, 96, 101, 187, 189, 190, 194, 199, 200, 201, 202, and 203.)

- [KMP07] KILIAN M., MITRA N., POTTMANN: Geometric modeling in shape space. *ACM Transactions on Graphics* 26 (2007), NID 64. (Cited pages xx, 41, 42, 43, and 44.)
- [KSDD03] KESELMAN Y., SHOKOUFANDEH A., DEMIRCI M., DICKINSON S.: Many-to-many graph matching via metric embedding. In *IEEE Computer Vision and Pattern Recognition* (2003), pp. 850–857. (Cited page 151.)
- [KSMJ04] KLASSEN E., SRIVASTAVA A., MIO W., JOSHI S. H.: Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (2004), 372–383. (Cited page 41.)
- [KSNo8] KANAMORI Y., SZEGO Z., NISHITA T.: Gpu-based fast ray casting for a large number of metaballs. *Computer Graphics Forum* 27 (2008), 351–360. (Cited page 17.)
- [KTo03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics* 22 (2003), 954–961. (Cited pages xxi, 50, 51, 52, 187, 189, 190, 191, 199, 203, and 204.)
- [KvD92] KOENDERINK J. J., VAN DOORN A. J.: Surface shape and curvature scales. *Image and Vision Computing* 10 (1992), 557–565. (Cited pages xxiii and 136.)
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics* 21 (1987), 163–169. (Cited page 21.)
- [LDB05] LAVOUÉ G., DUPONT F., BASKURT A.: A new cad mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design Journal* 37 (2005), 975–987. (Cited pages 49, 52, and 187.)
- [Lee03] LEE J. M.: *Introduction to Smooth Manifolds*. Springer, 2003. (Cited pages xxi, 61, 65, and 67.)
- [LGQ06] LI X., GU X., QIN H.: Curves-on-surfaces: a general shape comparison framework. In *IEEE Shape Modeling International* (2006), pp. 38–43. (Cited page 39.)

- [LGQo8] LI X., GU X., QIN H.: Surface matching using consistent pants decomposition. In *ACM Symposium on Solid and Physical Modeling* (2008), p. To appear. (Cited page 56.)
- [LKAo6] LIEN J.-M., KEYSER J., AMATO N. M.: Simultaneous shape decomposition and skeletonization. In *ACM Symposium on Solid and Physical Modeling* (2006), pp. 219–228. (Cited pages xxv, 188, and 202.)
- [LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics* 21 (2002), 362–371. (Cited pages xx, xxi, 39, 49, 50, 52, and 187.)
- [LSLC005] LIPMAN Y., SORKINE O., LEVIN D., COHEN-OR D.: Linear rotation-invariant coordinates for meshes. *ACM Transactions on Graphics* 24 (2005), 479–487. (Cited pages xx and 35.)
- [LTZo1] LI X., TOON T. W., Z H.: Decomposing polygon meshes for interactive applications. In *Symposium on Interactive 3D Graphics* (2001), pp. 35–42. (Cited pages 52 and 188.)
- [LV99] LAZARUS F., VERROUST A.: *Level Set Diagrams of Polyhedral Objects*. Tech. Rep. 3546, Institut National de Recherche en Informatique et en Automatique (INRIA), 1999. (Cited pages 84, 94, 100, and 135.)
- [LWCo06] LEE T.-Y., WANG Y.-S., CHEN T.-G.: Segmenting a deforming mesh into near-rigid components. *The Visual Computer* 22 (2006), 729–739. (Cited pages xxvi, 204, 208, 209, and 210.)
- [LZo7] LIU R., ZHANG H.: Mesh segmentation via spectral embedding and contour analysis. *Computer Graphics Forum* 26 (2007), 385–394. (Cited pages xxi, 50, 51, 96, 187, and 189.)
- [LZQo6] LIU Y., ZHA H., QIN H.: Shape topics: a compact representation and new algorithms for 3D partial shape retrieval. In *IEEE Computer Vision and Pattern Recognition* (2006), pp. 2025–2032. (Cited page 150.)
- [Lévo6] LÉVY B.: Laplace-Beltrami eigenfunctions: towards an algorithm that “understands” geometry. In *IEEE Shape Modeling International* (2006), pp. 13–20. (Cited pages 32 and 33.)

- [MB98] MESSMER B. T., BUNKE H.: A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (1998), 493–504. (Cited page 152.)
- [MDSB02] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics* (2002), pp. 33–57. (Cited pages 34, 39, 82, and 83.)
- [Mil63] MILNOR J.: *Morse Theory*. Princeton University Press, 1963. (Cited pages 54, 61, and 71.)
- [MM99] MENCL R., MÜLLER H.: Interpolation and approximation of surfaces from three-dimensional scattered data points. In *IEEE Visualization* (1999), pp. 223–232. (Cited page 22.)
- [MM06] MICHOR P. W., MUMFORD D.: Riemannian geometries on spaces of plane curves. *Journal of the European Mathematical Society* 8 (2006), 1–48. (Cited page 41.)
- [Mor25] MORSE M.: Relations between the critical points of a real function of n independant variables. *Transactions of the American Mathematical Society* 27 (1925), 345–396. (Cited pages 54 and 71.)
- [Mor34] MORSE M.: The calculus of variations in the large. *American Mathematical Society Colloquium* 18 (1934). (Cited pages v, 56, and 71.)
- [MP02] MORTARA M., PATANÈ G.: Affine-invariant skeleton of 3D shapes. In *IEEE Shape Modeling International* (2002), pp. 245–252. (Cited pages 96 and 101.)
- [MPB07] MARINI S., PARABOSCHI L., BIASOTTI S.: SHape Retrieval Contest 2007: Partial matching track. In *SHREC (in conjunction with IEEE Shape Modeling International)* (2007), pp. 13–16. (Cited pages 172, 174, 176, 217, and 223.)
- [MSB*03] MA K.-L., STOMPEL A., BIELAK J., GHATTAS O., KIM E. J.: Visualizing large-scale earthquake simulations. In *Supercomputing conference* (2003), pp. 48–61. (Cited pages xix and 2.)

- [Mun57] MUNKRES J.: Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics* 5 (1957), 32–38. (Cited page 170.)
- [MWLo2] MA C.-M., WAN S.-Y., LEE J.-D.: Three-dimensional topology preserving reduction on the 4-subfields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), 1594–1605. (Cited pages 47 and 132.)
- [MWOo3] MA W.-C., WU F.-C., OUHYOUNG M.: Skeleton extraction of 3D objects with radial basis functions. In *IEEE Shape Modeling International* (2003), pp. 207–215. (Cited pages 47 and 132.)
- [MZPo6] MAMOU K., ZAHARIA T., PRÊTEUX F.: Multi-chart geometry video: A compact representation for 3D animations. In *IEEE 3DPVT* (2006), pp. 711–718. (Cited page 204.)
- [NGHo4] NI X., GARLAND M., HART J.: Fair Morse functions for extracting the topological structure of a surface mesh. *ACM Transactions on Graphics* 23 (2004), 613–622. (Cited pages 54, 56, 95, and 154.)
- [OMTo2] OHBUCHI R., MUKAIYAMA A., TAKAHASHI S.: A frequency domain approach to watermarking 3D shapes. *Computer Graphics Forum* 21 (2002), 373–382. (Cited page 34.)
- [prior03] Princeton Shape Benchmark.
<http://shape.cs.princeton.edu/benchmark/> (2003). (Cited pages 16 and 21.)
- [PSBMo7] PASCUCCI V., SCORZELLI G., BREMER P. T., MASCARENHAS A.: Robust on-line computation of Reeb graphs: simplicity and speed. *ACM Transactions on Graphics* 26 (2007), 58.1–58.9. (Cited pages xxi, 54, 55, 87, and 135.)
- [PSFo4] PATANÈ G., SPAGNUOLO M., FALCIDIENO B.: Para-graph: Graph-based parameterization of triangle meshes with arbitrary genus. *Computer Graphics Forum* 23 (2004), 783–797. (Cited pages 55 and 56.)
- [PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for

- 3D shapes. *ACM Transactions on Graphics* 25 (2006), 549–559. (Cited pages 51 and 187.)
- [RCB05] RICARD J., COEURJOLLY D., BASKURT A.: Generalizations of angular radial transform for 2D and 3D shape retrieval. *Pattern Recognition Letters* 26 (2005), 2174–2186. (Cited page 149.)
- [Ree46] REEB G.: Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes-rendus des Séances de l'Académie des Sciences* 222 (1946), 847–849. (Cited pages 54, 76, 78, 164, and 165.)
- [RLBo8] REVAUD J., LAVOUÉ G., BASKURT A.: Improving Zernike moments comparison for optimal similarity and rotation angle retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence To appear* (2008). (Cited page 149.)
- [RLL^{*}06] RAY N., LI W.-C., LÉVY B., SHEFFER A., ALLIEZ P.: Periodic global parameterization. *ACM Transactions on Graphics* 25 (2006), 1460–1485. (Cited page 39.)
- [RT07] RENIERS D., TELEA A.: Skeleton based hierarchical shape segmentation. In *IEEE Shape Modeling International* (2007), pp. 179–188. (Cited pages 47 and 132.)
- [Rus07] RUSTAMOV R. M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Eurographics Symposium on Geometry Processing* (2007), pp. 225–233. (Cited page 34.)
- [RWP05] REUTER M., WOLTER F. E., PEINECKE N.: Laplace-spectra as fingers for shape matching. In *ACM Symposium on Solid and Physical Modeling* (2005), pp. 101–106. (Cited page 34.)
- [SBSCO06] SHARF A., BLUMENKRANTS M., SHAMIR A., COHEN-OR D.: Snappaste: an interactive technique for easy mesh composition. *The Visual Computer* 22 (2006), 835–844. (Cited page 180.)
- [SCOL^{*}04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Eurographics Symposium on Geometry Processing* (2004), pp. 179–188. (Cited pages xx and 35.)

- [Set99] SETHIAN J. A.: *Level set methods and fast marching methods*. Cambridge University Press, 1999. (Cited page 219.)
- [SF02] STEINER D., FISCHER A.: Cutting 3D freeform objects with genus-n into single boundary surfaces using topological graphs. In *ACM Symposium on Solid and Physical Modeling* (2002), pp. 336–343. (Cited page 55.)
- [Shao08] SHAMIR A.: A survey on mesh segmentation techniques. *Computer Graphics Forum To appear* (2008). (Cited pages 49, 51, and 187.)
- [SKK91] SHINAGAWA Y., KUNII T. L., KERGOSIEN Y. L.: Surface coding based on morse theory. *IEEE Computer Graphics and Applications* 11 (1991), 66–78. (Cited pages 55 and 87.)
- [SLMB05] SHEFFER A., LÉVY B., MOGILNITSKY M., BOGOMYAKOV A.: Abf: fast and robust angle based flattening. *ACM Transactions on Graphics* 24 (2005), 311–330. (Cited page 39.)
- [SLS*06] SHARF A., LEWINER T., SHAMIR A., KOBBELT L., COHEN-OR D.: Competing fronts for coarse-to-fine surface reconstruction. *Computer Graphics Forum* 25 (2006), 389–398. (Cited page 22.)
- [SLSDo7] SHARF A., LEWINER T., SHAMIR A., DANIEL C.-O.: On the fly curve skeleton computation for 3D shapes. *Computer Graphics Forum* 26 (2007), 323–328. (Cited pages 47 and 132.)
- [Soro06] SORKINE O.: Differential representations for mesh processing. *Computer Graphics Forum* 25 (2006), 789–807. (Cited pages 34 and 36.)
- [SPo04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Transactions on Graphics* 23 (2004), 399–405. (Cited page 44.)
- [Spi79] SPIVAK M.: *A comprehensive introduction to differential geometry*. Publish or Perish, 1979. (Cited page 42.)
- [SPPo08] SPRINBORN B., PETER S., PINKALL U.: Conformal equivalence of triangle meshes. *ACM Transactions on Graphics* 27 (2008), 77.1–77.11. (Cited page 39.)

- [SSGDo3] SUNDAR H., SILVER D., GAGVANI N., DICKINSON S. J.: Skeleton based shape matching and retrieval. In *IEEE Shape Modeling International* (2003), pp. 130–142. (Cited page 47.)
- [SSJDo8] SRIVASTAVA A., SAMIR C., JOSHI S., DAOUDI M.: Elastic shape models for face analysis using curvilinear coordinates. *Journal of Mathematical Imaging and Vision* (2008), To appear. (Cited pages 41, 43, and 44.)
- [SSK^{*}05] SURAZHSKY V., SURAZHSKY T., KIRSANOV D., GORTLER STEVEN J., HOPPE H.: Fast exact and approximate geodesics on meshes. *ACM Transactions on Graphics* 24 (2005), 553–560. (Cited page 81.)
- [SSP94] SULLIVAN S., SANDFORD L., PONCE J.: Using geometric distance fits for 3D object modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16 (1994), 1183–1196. (Cited page 17.)
- [STK02] SHLAFMAN S., TAL A., KATZ S.: Metamorphosis of polyhedral surfaces using decomposition. *Computer Graphics Forum* 21 (2002), 219–228. (Cited pages 52 and 188.)
- [SY07] SCHAEFER S., YUKSEL C.: Example-based skeleton extraction. In *Symposium on Geometry Processing* (2007), pp. 153–162. (Cited page 204.)
- [Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *SIGGRAPH* (1995), pp. 351–358. (Cited page 32.)
- [TO99] TURK G., O'BRIEN J.: Shape transformation using variational implicit functions. *Computer Graphics* 33 (1999), 335–342. (Cited page 17.)
- [TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEICHER K.: Elastically deformable models. *Computer Graphics* 21 (1987), 205–214. (Cited page 17.)
- [TS05] TUNG T., SCHMITT F.: The augmented multiresolution Reeb graph approach for content-based retrieval of 3d shapes. *International Journal of Shape Modeling* 11 (2005), 91–120. (Cited pages 55, 87, and 149.)

- [TV98] TARASOV S., VYALYI M.: Construction of contour trees in 3D in $O(n \log(n))$ steps. In *ACM Symposium on Computational Geometry* (1998), pp. 68–75. (Cited pages 84 and 87.)
- [TV04] TANGELDER J. W. H., VELTKAMP R. C.: A survey of content based 3D shape retrieval methods. In *IEEE Shape Modeling International* (2004), pp. 145–156. (Cited page 149.)
- [TVD06a] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: 3D mesh skeleton extraction using topological and geometrical analyses. In *Pacific Graphics* (2006), pp. 85–94. (Cited page vii.)
- [TVD06b] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Invariant high level Reeb graphs of 3D polygonal meshes. In *IEEE 3DPVT* (2006), pp. 105–112. (Cited page vii.)
- [TVD07a] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Reeb chart unfolding based 3D shape signatures. In *Eurographics* (2007), pp. 13–16. (Cited pages vii and 157.)
- [TVD07b] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Topology driven 3D mesh hierarchical segmentation. In *IEEE Shape Modeling International* (2007), pp. 215–220. (Cited page vii.)
- [TVD08a] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Enhancing 3D mesh topological skeletons with discrete contour constrictions. *The Visual Computer* 24 (2008), 155–172. (Cited page vii.)
- [TVD08b] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Fast and precise kinematic skeleton extraction of 3D dynamic meshes (to appear). In *IEEE International Conference on Pattern Recognition* (2008). (Cited page vii.)
- [TVD08c] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Partial 3D shape retrieval by Reeb pattern unfolding. *Computer Graphics Forum To appear* (2008). (Cited page vii.)
- [vKvOB*97] VAN KREVELD M., VAN OOSTRUM R., BAJAJ C L., PASCUCCI V., SHCIKORE D R.: Contour trees and small seed sets for isosurface traversal. In *ACM Symposium on Computational Geometry* (1997), pp. 212–220. (Cited pages 55 and 88.)

- [VLo7] VALLET B., LÉVY B.: *Spectral geometry processing with manifold harmonics*. Tech. Rep. ALICE-2007-001, INRIA, 2007. (Cited pages xx, 34, 35, and 36.)
- [VLo8] VALLET B., LÉVY B.: Spectral geometry processing with manifold harmonics. *Computer Graphics Forum* 27 (2008). (Cited pages xx, 32, and 33.)
- [WDSBoo] WOOD Z J., DESBRUN M., SCHRÖDER P., BREEN D E.: Semi-regular mesh extraction from volumes. In *IEEE Visualization* (2000), pp. 275–282. (Cited page 55.)
- [WHDS04] WOOD Z J., HOPPE H., DESBRUN M., SCHRÖDER P.: Removing excess topology from isosurfaces. *ACM Transactions on Graphics* 23 (2004), 190–208. (Cited page 55.)
- [WLo8] WANG Y.-S., LEE T.-Y.: Curve skeleton extraction using iterative least squares optimization. *IEEE Transactions on Visualization and Computer Graphics To appear* (2008). (Cited pages 47, 48, and 132.)
- [WMKG07] WARDETZKY M., MATHUR S., KÄLBERER F., GRINSPUN E.: Discrete laplace operators: No free lunch. In *Eurographics Symposium on Geometry Processing* (2007), pp. 33–37. (Cited page 36.)
- [WML*06] WU F.-C., MA W.-C., LIANG R.-H., CHEN B.-Y., OUHYOUNG M.: Domain connected graph: the skeleton of a closed 3D shape for animation. *The Visual Computer* 22 (2006), 117–135. (Cited pages 48, 132, 143, 144, 151, 152, 157, 217, and 223.)
- [WW86] WYVILL G., WYVILL B.: Data structure for soft objects. *The Visual Computer* 2 (1986), 227–234. (Cited page 17.)
- [WWJ*06] WANG S., WANG Y., JIN M., GU X., SAMARAS D.: 3D surface matching and recognition using conformal geometry. In *IEEE Computer Vision and Pattern Recognition* (2006), pp. 2453–2460. (Cited pages 40 and 41.)
- [WWJ*07] WANG S., WANG Y., JIN M., GU X., SAMARAS D.: Conformal geometry and its applications on 3D shape matching, recognition and stitching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2007), 1209–1220. (Cited pages 38, 40, and 41.)

- [YBS04] YOSHIZAWA S., BELYAEV A. G., SEIDEL H.-P.: A fast and simple stretch-minimizing mesh parameterization. In *IEEE Shape Modeling International* (2004), pp. 200–208. (Cited page 39.)
- [YBS07] YOSHIZAWA S., BELYAEV A. G., SEIDEL H.-P.: Skeleton-based variational mesh deformations. *Computer Graphics Forum* 26 (2007), 255–264. (Cited pages xxi, 48, and 132.)
- [ZLS07] ZAYER R., LÉVY B., SEIDEL H.-P.: Linear angle based parameterization. In *Eurographics Symposium on Geometry Processing* (2007), pp. 135–141. (Cited page 39.)
- [ZMT05] ZHANG E., MISCHAIKOW K., TURK G.: Feature-based surface parametrization and texture mapping. *ACM Transactions on Graphics* 24 (2005), 1–27. (Cited pages xxv, 55, 56, 188, and 202.)
- [ZRS05] ZAYER R., RÖSSL C., SEIDEL H.-P.: Discrete tensorial quasi-harmonic maps. In *IEEE Shape Modeling International* (2005), pp. 278–287. (Cited page 39.)
- [ZTS02] ZUCKERBERGER E., TAL A., SHLAFMAN S.: Polyhedral surface decomposition with applications. *Computer Graphics* 26 (2002), 733–743. (Cited page 52.)
- [ZvKDo8] ZHANG H., VAN KAICK O., DYER R.: Spectral mesh processing. *Computer Graphics Forum To appear* (2008). (Cited pages 32, 34, and 36.)

Titre Modélisation de forme 3D par graphe de Reeb et applications

Résumé Avec le développement récent des technologies 3D, les formes 3D sont devenues un type de données multimedia interactives de première importance. Leur représentation la plus courante, le maillage de polygones, souffre cependant de grande variabilité face à des transformations canoniques préservant la forme. Il est donc nécessaire de concevoir des techniques de modélisation intrinsèque de forme.

Dans cette thèse, nous explorons la modélisation topologique par l'étude de structures basées sur les graphes de Reeb. En particulier, nous introduisons une nouvelle abstraction de forme, appelée squelette topologique avancé, qui permet non seulement l'étude de l'évolution topologique des lignes de niveau de fonctions de Morse mais aussi l'étude de leur évolution géométrique. Nous démontrons l'utilité de cette représentation intrinsèque de forme dans trois problèmes de recherche liés à l'Informatique Graphique et à la Vision par Ordinateur.

Tout d'abord, nous introduisons la notion de calcul géométrique sur les graphes de Reeb pour le calcul automatique et stable de squelettes de contrôle pour la manipulation interactive de forme.

Ensuite, en introduisant les notions de cartes de Reeb et de motifs de Reeb, nous proposons une nouvelle méthode pour l'estimation de similarité partielle entre formes 3D. Nous montrons que cette approche dépasse les méthodes participant au concours international de reconnaissance de forme 2007 (SHREC 2007) par un gain de 14%.

Enfin, nous présentons deux techniques permettant de fournir une décomposition fonctionnelle d'une forme 3D, à la fois en considérant des heuristiques issues de la théorie de la perception humaine et des données 3D variant dans le temps.

Des exemples applicatifs concrets viennent illustrer l'utilité de notre approche pour chacun de ces problèmes de recherche.

Mots-clés Modélisation de forme 3D, topologie, graphe de Reeb, squelette de forme, indexation 3D par similarité partielle, segmentation.

Title Reeb graph based 3D shape modeling and applications

Abstract With the ongoing development of 3D technologies, 3D shapes are becoming an interactive media of major importance. Their commonest representation, the surface mesh, suffers however from high variability towards standard shape-preserving surface transformations. It is necessary thus to design intrinsic shape modeling techniques.

In this thesis, we explore topological modeling by studying Reeb graph based structures. In particular, we introduce a novel shape abstraction, called the enhanced topological skeleton, which enables not only the study of the topological evolution of Morse functions' level sets but also that of their geometrical evolution. We show the utility of this intrinsic shape representation in three research problems related to Computer Graphics and Computer Vision.

First, we introduce the notion of geometrical calculus on Reeb graphs for the stable and automatic computation of control skeletons for interactive shape handling.

Then, by introducing the notions of Reeb chart and Reeb pattern, we propose a new method for partial 3D shape similarity estimation. We show this approach outperforms the competing methods of the international SHape REtrieval Contest 2007 by a gain of 14%.

Finally, we present two techniques for the functional decomposition computation of a 3D shape, both from human perception based heuristics and from the analysis of time-varying 3D data.

For each of these research problems, concrete applicative examples are presented to assess the utility of our approach.

Keywords 3D shape modeling, topology, Reeb graph, shape skeleton, partial 3D shape retrieval, segmentation.