



HAL
open science

3D mesh Reeb graph computation using commute-time and diffusion distances

Rachid El Khoury, Jean-Philippe Vandeborre, Mohamed Daoudi

► **To cite this version:**

Rachid El Khoury, Jean-Philippe Vandeborre, Mohamed Daoudi. 3D mesh Reeb graph computation using commute-time and diffusion distances. 3D Image Processing (3DIP) and Applications, Jan 2012, Burlingame, California, USA, United States. pp.8090-16. hal-00666106

HAL Id: hal-00666106

<https://hal.science/hal-00666106>

Submitted on 3 Feb 2012

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

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

3D-Mesh Reeb Graph Computation Using Commute-time and Diffusion Distances

Rachid El Khoury, Jean-Philippe Vandeborre and Mohamed Daoudi

Institut Telecom; Telecom Lille1
LIFL (UMR 8022 Lille1/CNRS) France

ABSTRACT

3D-model analysis plays an important role in numerous applications. In this paper, we present an approach for Reeb graph extraction using a novel mapping function. Our mapping function computes a real value for each vertex which provides interesting insights to describe topology structure of the 3D-model. We perform discrete contour for each vertex according to our mapping function. Topology changes can be detected by discrete contours analysis to construct the Reeb graph.

Our mapping function has some important properties. It is invariant to rigid and non rigid transformations, it is insensitive to noise, it is robust to small topology changes, and it does not depend on parameters. From the extracted graph, these properties show the significant parts of a 3D-model. We retain the evaluation criteria to the properties of the mapping function, and compared them to those used in the state of the art. In the end, we present extracted Reeb graph on various models with different positions.

Keywords: Reeb graph, topological skeleton, feature points, mapping function, diffusion geometry, Laplace-Beltrami.

1. INTRODUCTION

Nowadays, 3D data become more and more a major interactive graphical experience. A growing number of 3D graphics applications have an impact on today's society. These applications are being used in several domains ranging from digital entertainment, computer aided design, to medical applications. The evolution of this domain has created the need for 3D processing and analysing tools at the geometric, topological, and semantic levels. In the same vein, we see the benefits of topology graphs such as Reeb graph in many applications in diverse fields. It has shown to be interesting for shape description,¹ surface parametrization,² mesh segmentation,³ retrieval,⁴ and so on.

When dealing with topological structure based on Reeb graph, obviously, we have to define an appropriate mapping function. This mapping function should respect the invariance to rigid and non rigid transformations, the insensitivity to noise, the robustness to small topology changes, and the independence on parameters. The definition of such a mapping function remains an open question.¹

To answer this question, we propose in this paper an approach to define a mapping function based on the diffusion distance⁵ and the commute-time distance.⁶ Then, we analyse the topology structure and compute the Reeb graph.

This paper is organized as follows: In section 2, we introduce topological structure and function on surfaces related work. Section 3 presents mathematical background and definitions. In section 4, we discuss the construction and the properties of our mapping function. Then, section 5 is about Reeb graph computation and topological analysis of the mesh. Parameter settings of our method are defined in section 6. Before the conclusion, the experiments that prove the efficiency of our approach are explored in section 7.

Further author information: (Send correspondence to Rachid El Khoury)

Rachid El Khoury: E-mail: rachid.elkhoury@telecom-lille1.eu, Telephone: +33 (0)3 20 33 55 17

Jean-Philippe Vandeborre: E-mail: vandeborre@telecom-lille1.eu, Telephone: +33 (0)3 20 33 55 96

Mohamed Daoudi: E-mail: mohamed.daoudi@telecom-lille1.eu, Telephone: +33 (0)3 20 33 55 35

2. SCALAR FUNCTION ON SURFACES RELATED WORK

Several approaches have been explored for analysing the topological structure of a 3D-model. Traditional skeleton extraction and mesh segmentation approaches are not clearly defined to preserve topological properties of a 3D-model faced to the Reeb graph theory.⁷ Our method proposes a solution to the definition of an appropriate scalar function that presents the stability and the invariance properties. We focused on methods existing in the state-of-the-art that address this problem.

The scalar function $f : S \mapsto \mathfrak{R}$ is defined by a real value for each vertex of a polygonal mesh. Critical points are located by this function. The type and the number of these critical points are related to the topology of the mesh. In the literature, the use of a scalar function computed over triangulated surfaces is an important tool for a 3D-model analysis in different tasks: topological structure of a 3D-model,⁸ surface coding and modelling,⁹ molecular analysis,¹⁰ and so on.

The height function was firstly applied by Shinagawa and Kunii¹¹ and is a simple example to understand the Reeb graph. Most of the methods define the scalar function using many techniques based on the geodesic distance (the length of the shortest path between vertices).^{3,4,12} These functions are very sensitive to small topology changes.

Mortara and Patané¹³ extracted feature points where Gaussian curvatures are dependant on a given threshold. However, the drawback of this method appears when using constant curvature surfaces.

Gebal et al¹⁴ define an autodiffusion function depending on the time variable that leads to very different levels of details. This function computes the remaining heat on each vertex after a scale time t . For scale invariance the authors normalise the spectrum (eigenvalues) of the graph. The autodiffusion function based on the heat kernel that depends on eigenvalues and eigenvectors of Laplace-Beltrami operator. In the discrete space for his application, the authors use the cotangent weight^{15,16} that minimize the Dirichlet energy. This may creates other unexpected local extrema due to the negative values of geometry weights.

Tierny et al¹² detect the farthest two points on the mesh, then the authors compute two sets of feature points based on local maximum (a vertex such that all its direct neighbours have a lower values) from the farthest two points. The intersection of the two sets computed eliminates the unwanted feature points. The authors define the scalar function as the geodesic distance to the nearest feature point. The method's disadvantage that is sensitive to a small topology changes.

Ni et al⁸ solve the Laplace equation adding selected vertices as constraints elements, the resulting function will be harmonic. This method yields to a smooth function to eliminates local maximal point except at its boundary.

3. MATHEMATICAL BACKGROUND AND DEFINITION

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

3.1 The definition of the heat kernel

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

$$\Delta_S U(x, t) = -\frac{\delta U(x, t)}{\delta t} \quad (1)$$

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

$$\Delta_S U(x, t) = \lambda^2 U(x, t) \quad (2)$$

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

$$-\Delta_S \psi_n = \lambda_n \psi_n \quad (3)$$

Since the equation 1 is a homogeneous linear equation the initial heat distribution $U(0, x)$ can be written in the basis as the following expansion:

$$U(0, x) = \sum_{n=0}^{\infty} \langle U(0, x), \psi_n(x) \rangle \psi_n(x) \quad (4)$$

where the inner product is defined as the surface integral $\langle U(0, x), \psi_n(x) \rangle = \int_S (U(0, y) \psi_n(y)) dy$.

The solution of the problem can be written as:

$$U(t, x) = \sum_{n=0}^{\infty} \langle U(0, x), \psi_n(x) \rangle \psi_n(x) e^{-\lambda_n t} \quad (5)$$

Finally

$$U(t, x) = \int_S \underbrace{\sum_{n=0}^{\infty} e^{-\lambda_n t} \psi_n(y) \psi_n(x)}_{K(t, x, y)} U(0, y) dy \quad (6)$$

$K(t, x, y): C^\infty((0, \infty) \times S \times S) \rightarrow \mathfrak{R}$ is the fundamental solution of the heat equation corresponding to the initial condition of an initial point source of heat at a known position called the heat kernel. For more details see Grigorian.¹⁷

3.2 Diffusion and Commute-Time distances

The diffusion distance is a metric⁵ defined using the heat kernels as

$$d_S(t, x, y) = \|K(t, x, z) - K(t, y, z)\|_{L_2(S)} \quad (7)$$

In spectral expression

$$d_S(t, x, y)^2 = \sum_{i=1}^{\infty} e^{-2\lambda_i t} (\psi_i(x) - \psi_i(y))^2 \quad (8)$$

The diffusion distance $d_S(t, x, y)$ between a pair of nodes (x, y) on the surface S measures the connectivity with paths of length time variable t that connect (x, y) . The connectivity is deduced from a normalized probability of paths that connect (x, y) . The diffusion distance increases in value if there is a few paths connecting a pair of nodes. For small t the diffusion distance depends on local information since the diffusion did not propagate significantly.¹⁸

The commute-time distance is defined similarly to the diffusion distance. But the commute-time distance take into consideration of all paths connecting a pair of nodes in the graph, not only paths with length t ¹⁹

$$dc_S(x, y)^2 = \sum_{i=1}^{\infty} \frac{1}{\lambda_i} (\psi_i(x) - \psi_i(y))^2 \quad (9)$$

3.3 Reeb Graph

A Reeb graph⁷ is an interesting graph to describe topology structure that encodes the connectivity of its level sets based on the critical points of a scalar function. In continuous space, Reeb graph is defined as follow: Let S be a compact manifold and $f : S \rightarrow \mathbb{R}$ be a real valued function on S . The Reeb graph of f is the quotient space of f in $S \times \mathbb{R}$ by the equivalence relation " \sim " defined as $(v_i, f(v_i)) \sim (v_j, f(v_j))$ if and only if $f(v_i) = f(v_j)$ and v_i, v_j belong to the same connected component of $f^{-1}(f(v_i))$. In an other way, all nodes having the same value under f are represented by one element in the Reeb graph. Fig 1 illustrates the Reeb graph on a human body and a 3D-model with genus 3 using the height function $f(v_i) = z_i$ the height coordinate of the vertex v_i .

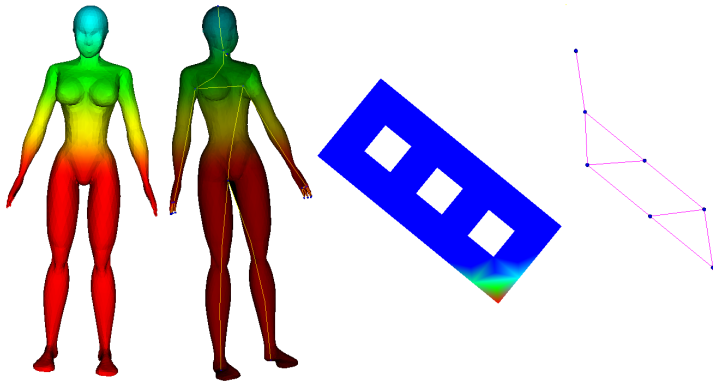


Figure 1. The Reeb graphs of a human body and of a 3D-model with genus 3, using the height function.

4. CONSTRUCTION AND PROPERTIES OF THE MAPPING FUNCTION

To compute the Reeb graph of 3D-model, we have to define an appropriate mapping function. This function must have properties of stability and invariance in order to highlight the most meaningful parts of a 3D-model. The construction of our mapping function consists of three steps. First, we extract the feature points. Then, the mapping function is computed based on these feature points. At last, the mapping function is transformed to a piecewise linear Morse function.

4.1 Feature point extraction

We define vertices located on the extremities of the important elements of the model as feature points. These feature points will be used as origins of our mapping function. In the literature, many techniques were proposed to detect feature points.^{12, 14, 20} The autodiffusion function proposed by Gebal et al¹⁴ detects feature point automatically since the quantity of heat that remains will be bigger on features or close to them. But to our experiments, this function generates undesirable feature points or misplaces them. Tierny et al¹² proposed a crossed analysis method using two geodesic function. The author's algorithm produces well-localized feature points but very sensitive to topology changes. Our method is inspired from Tierny et al.¹² The method starts by feature points extraction using the diffusion distance. In a large time variable t , global properties are detected and the farthest two feature points (where the diffusion distance is minimum) are computed. Let V_1 and V_2 be the farthest feature points. These two feature points are used as origins for two scalar function defined as

$$f_{V_1}(v, t) = d_S(t, v, V_1)^2 = \sum_{i=1}^{\infty} e^{-2\lambda_i t} (\psi_i(v) - \psi_i(V_1))^2 \quad (10)$$

$$f_{V_2}(v, t) = d_S(t, v, V_2)^2 = \sum_{i=1}^{\infty} e^{-2\lambda_i t} (\psi_i(v) - \psi_i(V_2))^2 \quad (11)$$

In a small variable time t , we compute local minimum diffusion distance (vertex that all its level-one neighbours have a higher value) to detect local properties. Let S_1 and S_2 be the set of local minima of f_{V_1} and

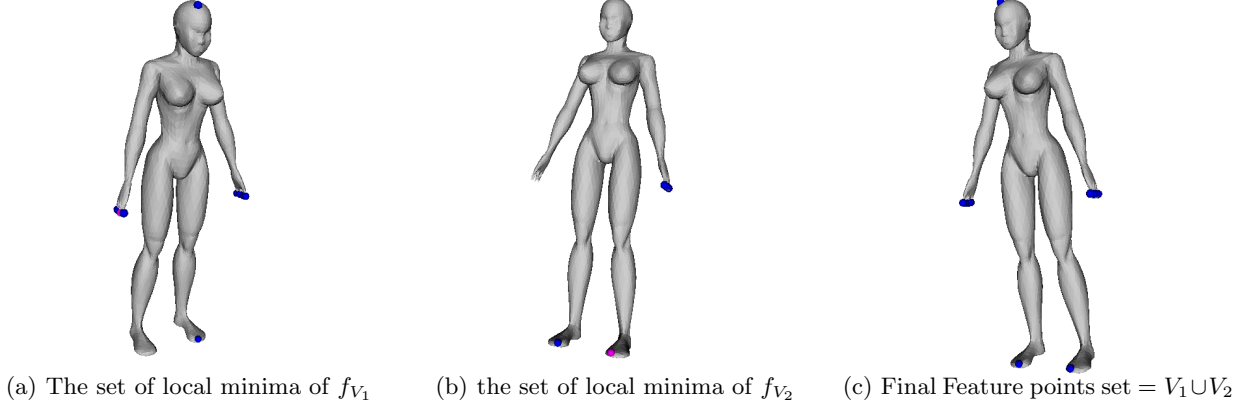


Figure 2. Approach for feature points detection. Fig 2(a) and Fig 2(b) show the set of local minima of f_{V_1} and f_{V_2} respectively and Fig 2(c) shows the final set of feature points.

f_{V_2} respectively (see Fig 2(a) and Fig 2(b)). The set of feature points is the union of S_1 and S_2 presented in Fig 2(c). In the discrete space, feature point will appear in the same neighbourhood of the vertex not exactly on the sameones this is why we put a limit dependant on a one level neighbourhood. This combination of local and global properties to extract feature points shows good results on different 3D-models under different rigid and non-rigid transformations (see Fig 3).

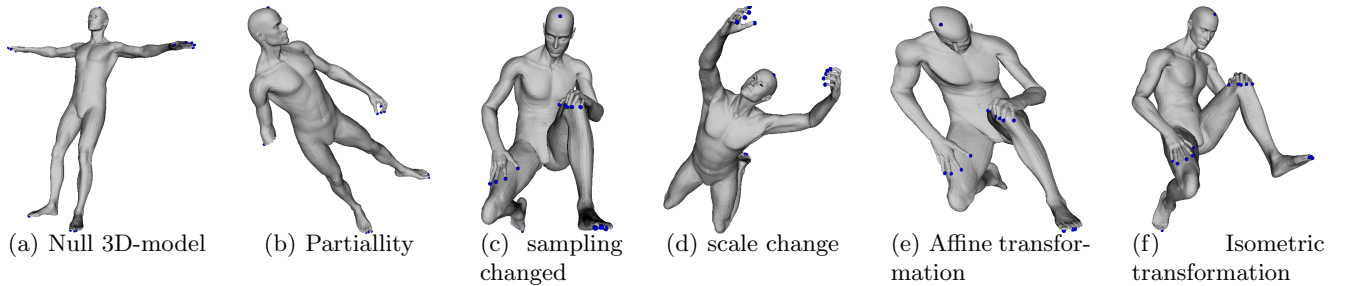


Figure 3. Feature points extracted from different poses of a 3D-model taken from Shrec 2011 database.

4.2 Mapping function computation and properties

The concept of the mapping function is to reveal the most meaningful parts of the model. Tierny et al¹² define a mapping function that computes for each vertex v the geodesic distance to the nearest feature point. This function does not handle small topology changes and is very sensitive to noise. Since the commute-time distance takes into consideration of all paths connecting a pair of nodes in the graph, a small topology changes do not affect enormously the results. Based on this distance we define our mapping function F_m that computes for each vertex v the commute-time distance to the nearest feature point.

$$F_m(v) = \max(dc_S(v, V_i), i = 1..nbV_i) \quad (12)$$

where V_i is the i^{th} feature point and nbV_i is the number of feature points. The mapping function is defined by the commute-time distance. It is computed using the using the eigenfunction and eigenvalue of the Laplace-Beltrami operator. It can be seen as global smooth scalar function and handle noisy data. For uniform scaling, we normalized the spectrum (eigenvalues) of the mesh. The mapping function is defined to be dependent only on the structure of the mesh. This makes it robust to isometric invariance.

4.3 Piecewise linear Morse function

In the discrete space, Reeb graph needs to be defined relatively to a piecewise linear Morse function. To transform the mapping function defined in section 4.2 to a Morse function, we used a perturbation strategy such that the mapping function takes distinct values in every vertex of the mesh. That will guarantee distinct values of the function on critical points and transform degenerate critical points into non degenerate ones (see Fig 4). Let $V_s = v_0, v_1, \dots, v_i, \dots, v_n$ the set of vertices, where v_i is the i^{th} vertex. We sorted V_s with regard to the $F_m(v)$, and added a constraint by considering the neighbours of a vertex. Then, we introduced the perturbation strategy induced by $f : S \rightarrow \mathfrak{R}$ as follows:

$$f(v_i) = \frac{i}{n} \tag{13}$$

with n number of vertices in V_s .

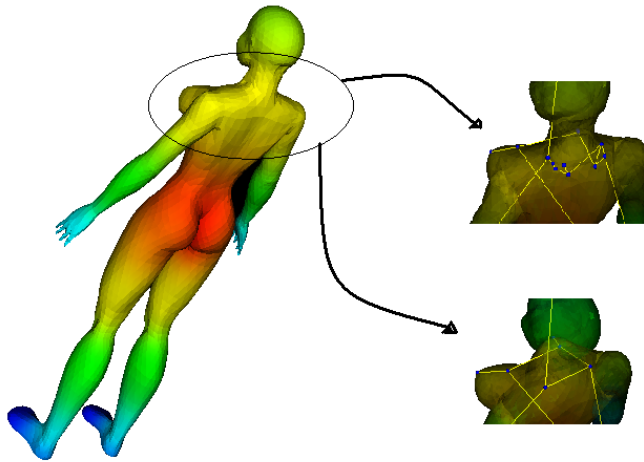


Figure 4. Top left is the mapping function. Bottom left is the Morse function after the perturbation strategy of the mapping function (to transform critical vertices into non degenerate ones).

5. REEB GRAPH COMPUTATION AND TOPOLOGICAL ANALYSIS OF THE MESH

After the computation of the Morse function in section 4.3 on the mesh, we use the defined function to generate level sets.²¹

These level sets are defined by the equivalence relation in section 3.3. We analyse them to define 4 types of topological changes: the start point is located where the value of the Morse function is the minimum (center of the 3D-model). Bifurcation or junction are detected where the level sets split or merge respectively, and the last types is the termination point. The critical points are created in each iteration where we detect a topological changes. These critical points and the connection between them correspond to the nodes and the edges of the Reeb graph.

6. PARAMETER SETTINGS OF OUR METHOD AND IMPLEMENTATION

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

In practice, we numerically computed the eigenfunctions and the eigenvalues using the discretization proposed by Meyer et al.²² We solve the generalized eigenvalue problem using the Implicitly Restarted Arnoldi Method implemented in MATLAB.

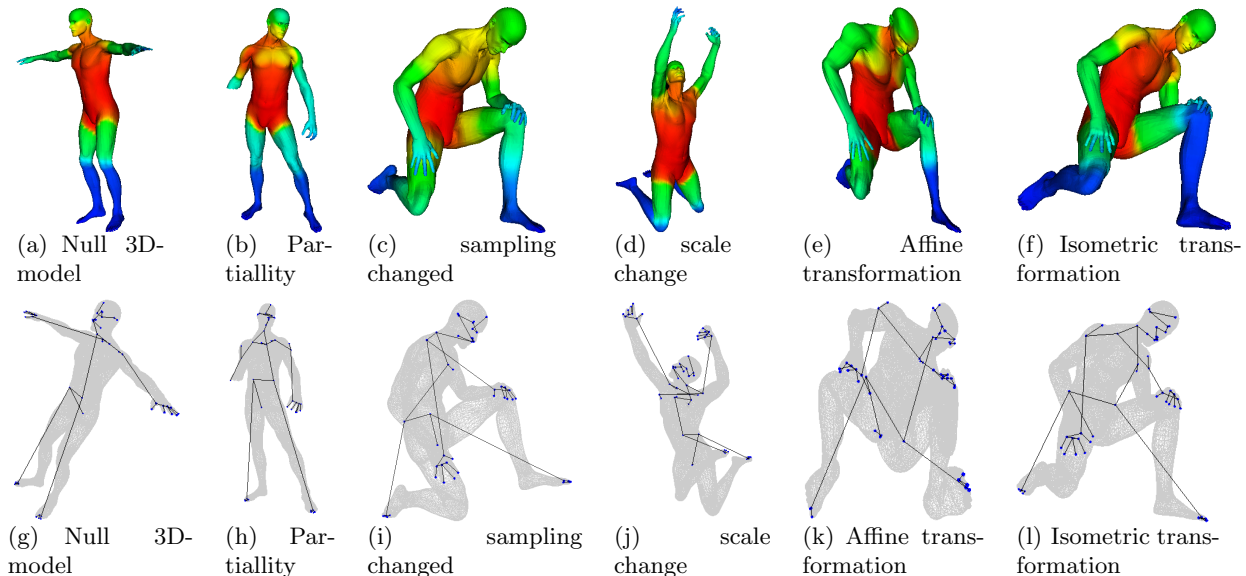


Figure 5. Fig(a) to fig(f) show the robustness of the mapping function against variations in 3D-model pose, red to blue colors express the increasing values of the mapping function. Fig(g) to Fig(l) show the corresponding Reeb graph of the mapping function.

3D-models with boundary are solved by imposing the Newman boundary condition.²³ We force the outward normal derivative to be zero on the boundary vertex.

We define the basis by 50 eigenfunctions related to the 50 smallest eigenvalues. The first 50 eigenvalues are fairly enough to detect the most important details and smooth enough to eliminate the noise. To extract feature points described in section 4.1, the diffusion distance is estimated in a small and in a large variable time t . For small t ranging in $[1, 2]$, the diffusion is propagated significantly to detect local properties. For a large $t > 15$, the diffusion distance remains almost unchanged; so we fix it to 20. Experiments and results of Reeb graphs on various 3D-models are generated using the Visualization Toolkit (VTK). The Visualization Toolkit implements the most robust Reeb graph computation algorithms.²¹

7. EXPERIMENTS AND RESULTS

In this section, we evaluate our method on 3D-models presented as connected triangulated surfaces and we discuss three points: experiments on our mapping function, examples on different types of 3D-models including partial and combined ones, and finally we prove robustness to topology changes.

7.1 Experiments on our mapping function

Our mapping function expresses the robustness against variations in 3D-model pose. In Fig 6, the same colors have been affected to the same corresponding regions. Variation from red to blue colors expresses the increasing values of the mapping function. Our mapping function does not depend on a parameter to handle the level of details. It describes the local details and also describes the global ones as shown in Fig 7(j). As an example the small fingers and the back of the camel are detected. The autodiffusion function defined by Gebal et al¹⁴ depends on a parameter to handle the level of details. For a large variable time t , the function describes the global details of the 3D-model and misses the small ones. For a small variable time t , the autodiffusion function detects all details and moreover, it manages an important number considered as details that will lead to create unwanted branches in the Reeb graph.

Two reasons, as drawbacks of our method, lead to perturb the Reeb graph. First, the scalar value based on the mapping function on each vertex and its neighbours are very close or almost similar. These values depend on a variation around 10^{-20} and affect a bifurcation to change direction. In Fig 6(d) the connection from the

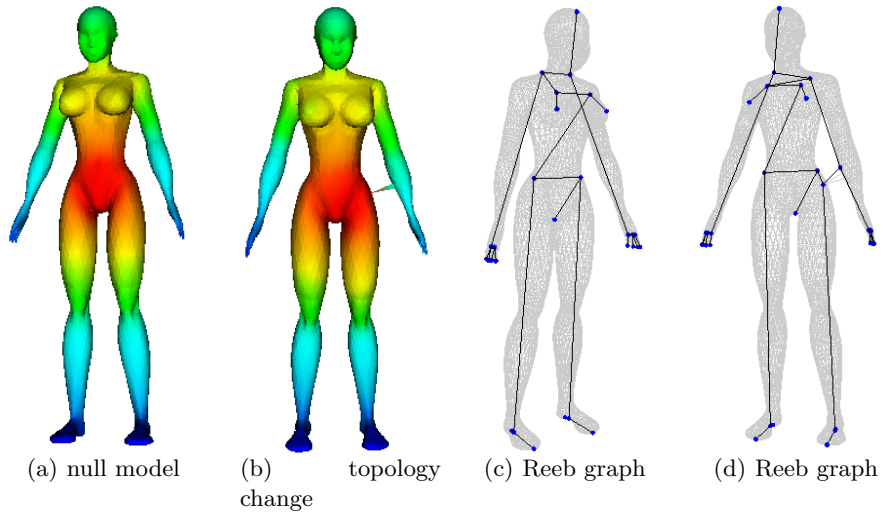


Figure 6. Robustness against topology changes

right breast to right the shoulder changed to the left shoulder. After the perturbation strategy, the Morse function takes into consideration the number of vertices to sort. This leads to detect bifurcation or junction in the construction of the Reeb graph, such as the chest of the human model in Fig 5(j).

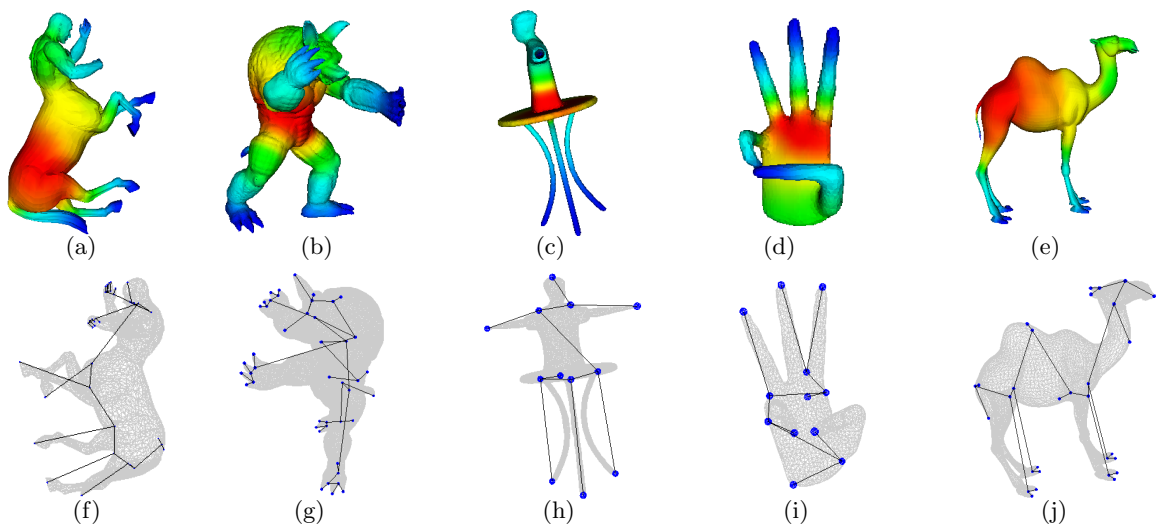


Figure 7. Reeb graph for different types of 3D-models.

7.2 Examples on different types of 3D-models

We tested our method on different types of 3D-models including 3D-models with missing parts or combined with others. Fig 5(a) and Fig 5(b) show the robustness of our mapping function that leads to a stable Reeb graph. The Reeb graph of a partial 3D-model (Fig 5(h)) includes the Reeb graph of its null 3D-model (Fig 5(g)). Fig 7 shows examples of our method applied on 3D-models with rigid and non rigid transformations.

7.3 Robustness to topology changes

Fig 7 shows the robustness of the mapping function toward a small topology change. The colors remain the same after adding a small topology change linking the hump and the hand in Fig 6(b). This proves the invariance of

our mapping function toward a topology change, due to the commute-time distance which took into consideration all paths connecting the vertices. If we apply a function based on the geodesic distance like the one defined by Tierny et al,²⁴ the Reeb graph will globally be changed. But using our mapping function, the Reeb graph undergoes the local change to describe the genus created toward the topology change.

8. CONCLUSION

To conclude, we presented in this paper a robust invariant mapping function for Reeb graph computation. This mapping function has good properties as the invariance to rigid and non rigid transformations, the insensitivity to noise and the robustness to small topology changes. We used this function to compute the Reeb graph which encodes the topological information of the 3D-model. Our approach includes a method to extract stable feature points defined on the extremities of a 3D-model. Also, the paper includes a perturbation strategy to transform the mapping function to a Morse function. To show the effectiveness of our approach, we applied our method to different types of 3D-models including partial and combined ones and we prove the robustness toward a topology change.

REFERENCES

- [1] S. Biasotti, S. Marini, M. Mortara, and G. Patane, “An overview on properties and efficacy of topological skeletons in shape modeling,” in *Shape Modeling International*, pp. 245 – 254, May 2003.
- [2] D. Steiner and A. Fischer, “Cutting 3d freeform objects with genus-n into single boundary surfaces using topological graphs.,” in *Symposium on Solid Modeling and Applications*, pp. 336–343, 2002.
- [3] A. Shamir, “A formulation of boundary mesh segmentation,” in *Symposium on 3D Data Processing, Visualization and Transmission. Proceedings*, pp. 82 – 89, September 2004.
- [4] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, “Topology matching for fully automatic similarity estimation of 3d shapes,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques, SIGGRAPH '01*, pp. 203–212, ACM, 2001.
- [5] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, “Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps,” *Proceedings of the National Academy of Sciences of the United States of America* **102**(21), pp. 7426–7431, 2005.
- [6] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, “Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation,” *IEEE Transactions on Knowledge and Data Engineering* **19**, 2006.
- [7] G. Reeb, “Sur les points singuliers d’une forme de Pfaff complètement intégrable ou d’une fonction numérique,” *Comptes Rendus Acad. Sciences* **222**, pp. 847–849, 1946.
- [8] X. Ni, M. Garland, and J. C. Hart, “Fair morse functions for extracting the topological structure of a surface mesh,” *ACM Trans. Graph.* **23**, pp. 613–622, August 2004.
- [9] Y. Shinagawa, T. Kunii, and Y. Kergosien, “Surface coding based on morse theory,” *IEEE Computer Graphics and Applications* **11**, pp. 66–78, September 1991.
- [10] P.-T. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci, “A topological hierarchy for functions on triangulated surfaces,” *IEEE Transactions on Visualization and Computer Graphics* **10**, 2004.
- [11] Y. Shinagawa and T. Kunii, “Constructing a reeb graph automatically from cross sections,” *Computer Graphics and Applications, IEEE* **11**, pp. 44 –51, nov 1991.
- [12] J. Tierny, J. P. Vandeborre, and M. Daoudi, “3d Mesh Skeleton Extraction Using Topological and Geometrical Analyses,” in *14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006)*, pp. 85–94, (Taipei, Taiwan), 2006.
- [13] M. Mortara and G. Patane, “Affine-invariant skeleton of 3d shapes,” in *Shape Modeling International.*, pp. 245 –252, 2002.
- [14] K. Gebal, J. A. Brentzen, H. Aans, and R. Larsen, “Shape analysis using the auto diffusion function,” *Computer Graphics Forum* , pp. 1405–1413, 2009.
- [15] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov, “Shape google: Geometric words and expressions for invariant shape retrieval,” *ACM Trans. Graph.* **30**, Feb. 2011.

- [16] U. Pinkall, S. D. Juni, and K. Polthier, “Computing discrete minimal surfaces and their conjugates,” *Experimental Mathematics* **2**, pp. 15–36, 1993.
- [17] A. Grigor’yan, “Estimates of heat kernels on riemannian manifolds,” *Spectral Theory and Geometry. ICMS Instructional Conference, Edinburgh, 1998* **273**, pp. 140–225, 1999.
- [18] “An introduction to diffusion maps,” *Proceedings Nineteenth Annuals Symposium of Pattern Recognition Society of South Africa*, .
- [19] H. Qiu and E. Hancock, “Clustering and embedding using commute times,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**, pp. 1873 –1890, November 2007.
- [20] S. Katz, G. Leifman, and A. Tal, “Mesh segmentation using feature point and core extraction,” *The Visual Computer* **21**(8-10), pp. 649–658, 2005.
- [21] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas, “Robust on-line computation of reeb graphs: simplicity and speed,” *ACM Trans. Graph.* , 2007.
- [22] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, “Discrete Differential-Geometry Operators for Triangulated 2-Manifolds,” in *VisMath '02*, (Berlin, Germany), 2002.
- [23] B. Vallet and B. Lévy, “Manifold harmonics,” tech. rep., INRIA - ALICE Project Team, 2007.
- [24] J. Tierny, J.-P. Vandeborre, and M. Daoudi, “Fast and precise kinematic skeleton extraction of 3d dynamic meshes,” in *19th International Conference on Pattern Recognition*, pp. 1 –4, December 2008.