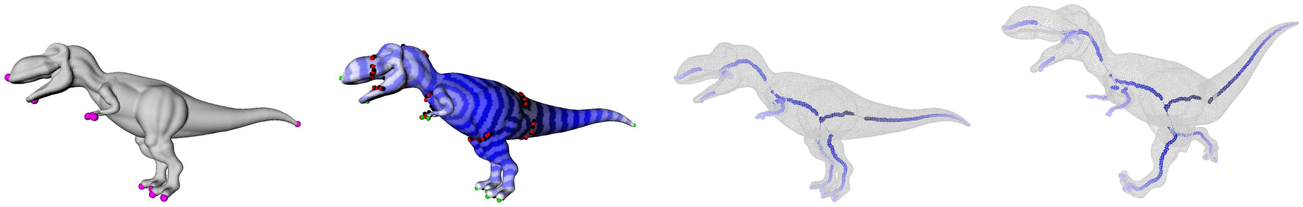
Invariant High Level Reeb Graphs of 3D Polygonal Meshes*

Julien Tierny, Jean-Philippe Vandeborre+, Mohamed Daoudi+

FOX-MIIRE Research Group – LIFL (UMR USTL/CNRS 8022)
+ GET / INT / Télécom Lille 1, France
{*tierny, vandeborre, daoudi*}@*lifl.fr*



(a) Feature points.

(b) Invariant quotient function.

(c) Skeletal representation of the high level Reeb graph.

(d) Application to mesh deformation.

Figure 1. Overview of presented method on an arbitrary mesh.

Abstract

Many applications in computer graphics need high level shape descriptions, in order to benefit from a global understanding of shapes.

Topological approaches enable pertinent surface decompositions, providing structural descriptions of 3D polygonal meshes; but in practice, their use raises several difficulties.

In this paper, we present a novel method for the construction of invariant high level Reeb graphs, topological entities that give a good overview of the shape structure. With this aim, we propose an accurate and straightforward feature point extraction algorithm for the computation of an invariant and meaningful quotient function. Moreover, we propose a new graph construction algorithm, based on an analysis of the connectivity evolutions of discrete level lines. This algorithm brings a practical solution for the suppression of non-significant critical points over piecewise continuous functions, providing meaningful Reeb graphs.

Presented method gives accurate results, with satisfactory execution times and without input parameter. The geometrical invariance of resulting graphs and their robustness to variation in model pose and mesh sampling make them good candidates for several applications, like shape deformation (experimented in this paper), recognition, compression, indexing, etc.

1 Introduction

Polygonal mesh is a widely used representation of 3D shapes. However, many applications in computer graphics need a higher level shape description as an input, like its structure for example.

To solve this problem, several approaches have been developed, like mesh segmentation [14] or skeleton extraction [5]. Topological methods, based on Reeb graphs, present the advantage to preserve the topological properties of the surface. Unfortunately, in practice, the construction of Reeb graphs raises several issues, like invariance constraint non-respect or non-significant critical point identification. This may come to results of restricted usability and of low semantic interest, as underlined in [12], that we refer as *low-level* Reeb graphs.

In this paper, we present a novel method for the construction of invariant high level Reeb graphs. First, we present theoretical backgrounds and related works. Secondly, we introduce a new feature point extraction algorithm (cf. figure 1(a)), which is used for the computation of a meaningful and invariant quotient function (figure 1(b)), and a new graph construction algorithm (figure 1(c)) that excludes non-significant critical points. Finally, we present and comment on experimental results and evoke possible applications, like mesh deformation (figure 1(d)).

*To be published in the 3^{rd} IEEE International Symposium on 3D Data Processing, Visualization and Transmission (*3DPVT'06*), June 14-16 2006, Chapel Hill, USA. Copyright IEEE Computer Society.

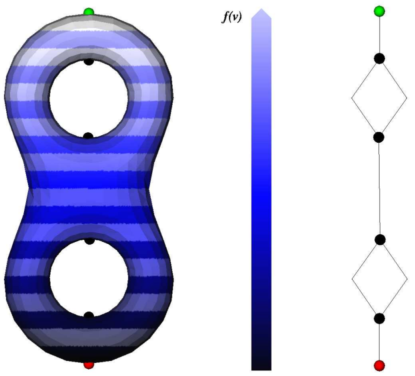


Figure 2. Evolution of the level lines of the height function on a bi-torus, its critical points and its Reeb graph.

2 Theoretical background

A Reeb graph [13] is a structure that depicts the evolutions of the level lines of a given scalar function, usually a Morse function [10], defined over objects of any dimension (*k-manifolds*). In this section, we introduce several notions, restricted to $2-$manifolds, used in our approach.

Definition 1 (Level line) *Let f be a scalar function defined on a $2-$manifold M, $f : M \rightarrow \mathbb{R}$. Let L be the set of points p_i, such as $\forall p_i \in L, f(p_i) = f_L$. L is the* level line *corresponding to the value f_L, noted $f^{-1}(f_L)$.*

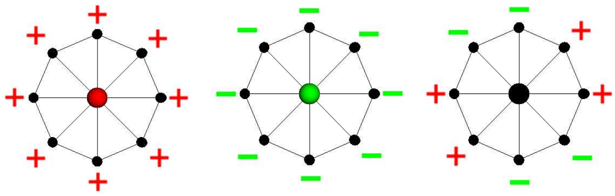
Definition 2 (Reeb graph) *Let $f : M \rightarrow \mathbb{R}$ be a scalar function defined on a $2-$manifold M. The* Reeb Graph *of f is the quotient space of f in $M \times \mathbb{R}$ by the equivalence relation $(p_1, f(p_1)) \sim (p_2, f(p_2))$, verified if and only if:*

$$\begin{cases} f(p_1) = f(p_2) \\ p_1 \text{ and } p_2 \text{ belong to the same connected} \\ \qquad \text{component of } f^{-1}(f(p_1)) \end{cases}$$

Concretely speaking, a Reeb graph is composed of nodes, which represent f critical points (see illustrations 2 and 3), and of edges, which represent the connected components linking f critical points. Most of the Reeb graph construction algorithms first identify the set of f critical points and secondly build the graph from the connectivity relations of these points.

Definition 3 (Critical point) *Let f be a scalar function defined on a $2-$manifold M, $f : M \rightarrow \mathbb{R}$. A point $p \in M$ is a* critical point *of f is the gradient of f vanished in p.*

Definition 4 (Non-degenerate critical point) *A critical point p of a scalar function f defined on a $2-$manifold is called a* non-degenerate critical point *if the matrix of second partials is non-singular in p.*



(a) L_+=1, L_-=0 (b) L_+=0, L_-=1 (c) L_+=2, L_-=2

Figure 3. Non-degenerate critical point classification.

In the discrete case, given a vertex v of a triangulated surface T, let L_+ and L_- be respectively the number of connected sets of vertices v_i and v_j on $Lk(v)$ (v direct neighbors, its *link*) such as $f(v_i) > f(v)$ and $f(v_j) < f(v)$.

Vertices corresponding to non-degenerate critical points can be classified into three categories, according to L_+ and L_-, as shown in figure 3: local minima (3(a)), local maxima (3(b)) and simple saddles (3(c)). We will use figure 3 color convention in the rest of the paper.

As mentioned by Biasotti et al. [1], continuous functions, and particularly Morse functions [10] (when all the critical points are non-degenerate), are pertinent choices for functions f. For clarity purpose, we will refer to candidate functions as *quotient* functions in the rest of the paper, to underline their use in graph construction.

Figure 2 illustrates previous definitions, presenting respectively a scalar function (the height function) computed on a $2-$manifold, its non-degenerate critical points, level lines of arbitrary range and the corresponding Reeb graph.

3 Related work

Several approaches have been explored by the computer graphics community in polygonal mesh decomposition.

Topological methods aim to produce structural descriptions of shapes where identified sub-components form connected sets. Morse and Reeb graph theories are two powerful mathematical tools that respectively enable to identify topological points of interest over the mesh and to capture their connectivity relations into a graph structure (see figure 2). Several algorithms [16, 3] have been proposed to construct Reeb-like graphs in $O(n \times log(n))$ steps (where n is the number of vertices) from piecewise linear functions computed over the mesh. Tarasov and Vyalyi [16] propose local mesh redefinition in order to purge multiple saddles and to benefit from Morse functions properties [10]. However, those papers do not address the problem of the quotient function definition. Lazarus and Verroust [9] introduces a novel notion of topological structure called *Level Set Di-*

agram, whose construction is somewhat similar to that of a Reeb graph. An interesting point of this method is the use of a quotient function, exploited for the diagram construction, based on the geodesic distance to a source vertex. Unfortunately, this scalar function suffers from a high instability [6], due to the non-determinist election of the source vertex, which excludes its use in applications where stability is fundamental, like shape retrieval for example. To overcome this problem, within the framework of shape retrieval, Hilaga et al. [6] propose to integrate this function over the whole mesh. More precisely, they define an approximation function that introduces geodesic distances to a set of *base* vertices, homogeneously spread over the mesh. Moreover, they propose a new topological structure scheme called *Multiresolutional Reeb Graph*. Basically, they proceed in a succession of mesh partitionings, according to different value-ranges of their quotient function. This initiative is of a major interest for 3D shape retrieval but does not fit other applicative contexts mainly because it will be difficult to automatically define a proper value-range for an application or another. For an interesting survey on computational topology approaches for shape modeling, we defer the reader to [1].

In short, the first key issue of topological approaches resides in the definition of the quotient function. A non pertinent function will present an important number of critical points, as underlined in [12]. Consequently the resulting Reeb graph will present many nodes and edges and will not afford a global and meaningful description of the shape (*low-level* Reeb graphs). Moreover, depending on application needs, the quotient function should present stability and invariance properties [9, 6] but those functions often generate many critical points from our experience.

The second key issue of topological approaches resides in the graph construction strategy, which should only use significant critical points and should not be conditioned by a user-parameter. In this paper, we present a novel method, whose objective is to produce *high-level*, stable, invariant and generically-exploitable Reeb graphs, solving previously mentioned issues.

4 Invariant high level Reeb graphs

Reeb graphs give a good overview of the structure of polygonal meshes. Unfortunately, constructing a Reeb graph from a non pertinent quotient function often leads to results of low semantic interest – *low-level* Reeb graphs – because of the number of identified critical points.

In this paper, we propose to challenge this key issue by proposing an invariant quotient function combined with a pertinent critical point selection algorithm, that enables the construction of *high-level* Reeb graphs over piecewise continuous functions.

4.1 Invariant quotient function

Several scalar functions have been proposed by the computer graphics community to construct topological structures. The choice of this function will directly condition the stability properties of the topological structure and therefore its usability in application fields. In this section, we detail our strategy to compute an invariant and meaningful quotient function: first, we extract mesh feature points and secondly, for each vertex in the mesh, we compute its geodesic distance to the closest feature point.

4.1.1 Metric space

Before defining the scalar function itself, a metric space needs to be defined. We will distinguish two kinds of metric spaces: on the first hand euclidean ones and on the other hand Riemannian ones.

Rotation and translation invariance properties can be obtained in both, as long as origins are taken relatively to the mesh. As an example, with an euclidean metric space, the euclidean barycenter of the mesh can be chosen.

Scaling invariance property can be obtained in both metric spaces as well, normalizing values in a regard to mesh global extrema.

In shape modeling, geodesic distances are interesting Riemannian metrics, because their evaluations are tolerant to variations in model pose. In metric geometry, a geodesic is a curve which is everywhere locally a distance minimizer. More concretely, the geodesic distance between two vertices is the length of the shortest path between them along the mesh, according to a given metric. As an example, a geodesic distance from a hand of a humanoid model to its head will be the same whether its arms are folded or not. For a formal description of Riemannian geometry, we defer the reader to [4].

In our approach, to guarantee invariance to rotation, translation, scaling and model pose, we decide to choose Riemannian metrics, and particularly geodesic distances. From an algorithmic point of view, geodesic distances can be approximated by the Moore-Dijkstra algorithm (distance minimizing in weighted graphs). In the rest of this paper, we will refer to $\delta(v_1, v_2)$ as the normalized approximation of the geodesic distance from vertex v_1 to vertex v_2.

4.1.2 Feature point extraction

Feature points are mesh vertices located on extremities of prominent components. From a perceptive point of view, the set of feature points describes in a meaningful way the global structure of a shape. From this observation, in a comparable way to [2], we propose in our method to use the set of feature points as origins for geodesic distance evaluations.

Several algorithms have been proposed for feature point extraction. They find applications in various fields, like shape metamorphosis, deformation transfer, texture mapping, etc. For example, Mortara and Pantanè [11] propose to select as features points the vertices where gaussian curvature exceeds a given threshold. Unfortunately, this method can miss feature points because of the threshold parameter and cannot resolve extraction on constant curvature areas. Katz et al. [7] developed an algorithm based on multidimensional scaling, in quadratic execution complexity.

Here, we propose a quite straightforward algorithm, based on topological tools. Most of the geodesic based scalar function local extrema appear at extremities of prominent components (see illustrations 4(a) and 4(b)), mainly because gradients vanish in those configurations (cf. definition 3). Therefore, we propose to realize a crossed analysis, using two geodesic based scalar functions – whose origins are the mesh most distant vertices – and to intersect the sets of their local extrema.

Definition 5 (Feature points) *Let f_1 and f_2 be two scalar functions defined on a connected triangulated surface T as follows:*

$$f_1(v) = \delta(v, v_{s_1}) \tag{1}$$

with:

$$v_{s_1} \in T \quad / \quad \delta(v_{s_1}, v_r) = max_{v \in T} \quad \delta(v, v_r) \tag{2}$$

with $v_r \in T$ a randomly chosen vertex, and:

$$f_2(v) = \delta(v, v_{s_2}) \tag{3}$$

with:

$$v_{s_2} \in T \quad / \quad f_1(v_{s_2}) = max_{v \in T} \quad f_1(v) \tag{4}$$

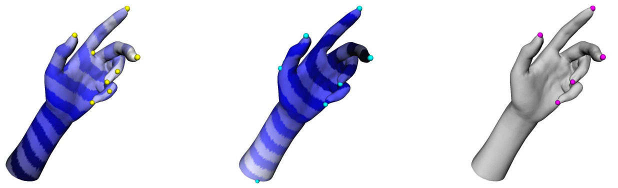
Let E_1 and E_2 be the sets of local extrema of f_1 and f_2. We define the set of feature points F of T as the intersection of E_1 and E_2 :

$$F = E_1 \cap E_2 \tag{5}$$

Concretely, we perform a crossed analysis in order to purge non-isolated extrema, as illustrated in figure 4: f_1 local extrema are displayed in yellow (figure 4(a)) and f_2 extrema in cyan (figure 4(b)). Equation 5 is well illustrated in figure 4.

In practice, f_1 and f_2 isolated local extrema do not appear exactly on the same vertices but in the same *geodesic neighborhood*. Therefore, we relax the intersection constraint as follows:

$$v \in F \Longleftrightarrow \begin{cases} \exists v_{e_1} \in E_1 \quad / \quad \delta(v, v_{e_1}) < \epsilon \\ \exists v_{e_2} \in E_2 \quad / \quad \delta(v, v_{e_2}) < \epsilon \\ \delta(v, v_{f_i}) > \epsilon \quad \forall v_{f_i} \in F \\ \epsilon \in [0, 1] \end{cases} \tag{6}$$

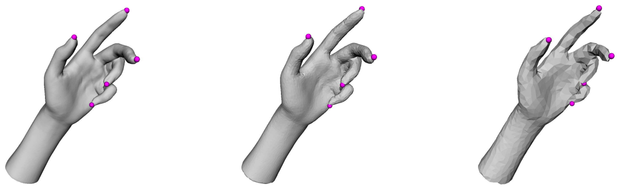


(a) E_1.

(b) E_2.

(c) $E_1 \cap E_2$.

Figure 4. Feature point extraction overview.



(a) 25 000 vertices.

(b) 5 000 vertices.

(c) 1 000 vertices.

Figure 5. Feature point extraction robustness against mesh sampling variations.

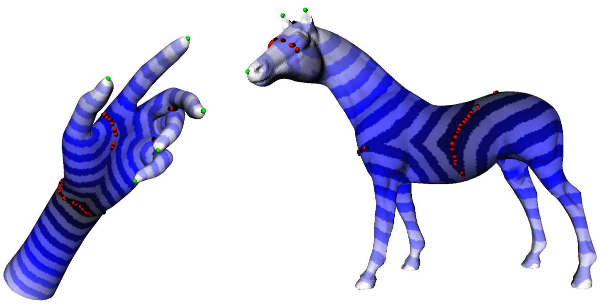
Local extrema identification is realized according to the classification presented in figure 3. Moore-Dijkstra's algorithm is an execution complexity bottleneck. f_1 and f_2 are computed each in $O(n \times log(n))$ steps, where n is the number of vertices in the mesh. E_1 and E_2 relaxed intersection is performed in $O(k \times m \times log(m))$, where $k = min_{i \in \{1,2\}} \quad |E_i|$ and m is the number of vertices in the geodesic neighborhoods.

In our experiments, setting $\epsilon = 0.05$ gives accurate results. With this configuration, m never exceeds two percent of n and k rarely exceeds 30 (depending on the model's topological complexity). Therefore E_1 and E_2 relaxed intersection algorithm's execution complexity is negligible compared to f_1 and f_2 computations.

In this section, we presented a straightforward algorithm for mesh feature point extraction. This algorithm is based on geodesic distance evaluations and therefore is stable and invariant to geometrical transformations and to model pose. Moreover, in order to select feature points, we observe geodesic gradient behaviors. Consequently, we can state that our method is robust against mesh sampling variations, as illustrated in figure 5.

4.1.3 Quotient function definition

The quotient function definition depends on what is expected to be revealed. As an example, for terrain modeling, the height function will present critical points over hills and



(a) 25 000 vertices, 6 features points.

(b) 50 000 vertices, 7 features points.

Figure 6. Evolution of the level lines of f_q and its critical points on arbitrary shapes.

valleys and will afford consequently an appropriate topological description. In our approach, we would like to identify meaningful mesh sub-components.

As underlined in paragraph 4.1.2, the set of feature points describes in a meaningful way the global structure of a shape. This set is invariant to geometrical transformations and variations in model pose and mesh sampling. From these observations, we propose to compute a meaningful quotient function from geodesic distances to the set of feature points.

In particular, an interesting objective would be to make f function level lines cut as precisely as possible the basis of prominent components, to afford a meaningful decomposition. With this aim, we use the set of feature points as origins for distance evaluations and propose the following quotient function, computed for each vertex v of an input connected triangulation T, noted f_q in the rest of the paper:

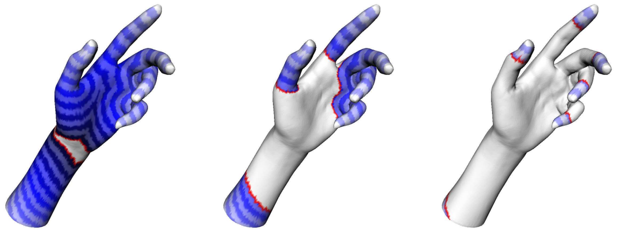
$$f_q(v) = 1 - \hat{\delta}(v, v_c) \tag{7}$$

with v_c the closest feature point from v:

$$v_c \in F \quad / \quad \hat{\delta}(v, v_c) = min_{v_{f_i} \in F} \quad \delta(v, v_{f_i}) \tag{8}$$

Figure 6 presents some computations of f_q over arbitrary shapes, as well as the number of extracted feature points.

In our experiments, we compute the set of geodesic distances towards feature points within the feature point extraction algorithm. Therefore, the number of iterations of the Moore-Dijkstra algorithm over the whole mesh is equal to the number of feature points. The execution complexity of f_q computation can be consequently approximated by $O(|F| \times n \times log(n))$, with n the number of vertices in the mesh.



(a) t = 600.

(b) t = 10 000.

(c) t = 20 000.

Figure 7. Geodesic propagation overview on a 25 000 vertex mesh (f_q function).

4.2 High level Reeb graph construction

As we can see in figure 6, f_q is a piecewise continuous function only. More precisely, discontinuity appears on areas where geodesic origins change. In figure 6, those areas can be identified with the rings of local minima and local saddles (in red and black). From a theoretical point of view, resulting Reeb graph should present as many nodes as function critical points and consequently would not afford a global, or *high level*, description of the shape.

In this section, we propose to challenge this issue with a novel critical point election algorithm. With this aim, we propose to focus on an intuitive description of Reeb graphs: *Reeb graphs are topological structures that depict the connectivity evolution of the level lines of a given scalar function defined on a* $2-$*manifold.*

Consequently, we present a method that does not propose to construct a Reeb graph from the connectivity relations of f critical points. Instead, we propose to construct *discrete* level lines around f global minimum and to observe their connectivity relations while f evolves.

Discrete level lines evolution can be modeled as a geodesic propagation within a metric space based on the quotient function values. Such a propagation can be obtained with the Moore-Dijkstra algorithm, considering for each vertex its f value as weight.

Let t be a parametric variable that denotes an iteration of the geodesic propagation algorithm. Now, we can introduce the notion of *discrete* level line:

Definition 6 (Discrete level line) *Let f be a scalar function defined on a connected triangulated surface T, $f : T \to \mathbb{R}$.*

Let $C(t)$ be the set of candidate vertices for absorption at the iteration t of the geodesic propagation algorithm.

Let $l_1(t), l_2(t), ..., l_k(t)$ be the connected subsets of vertices belonging to $C(t)$. We define each connected subset $l_i(t)$ as a discrete level line.

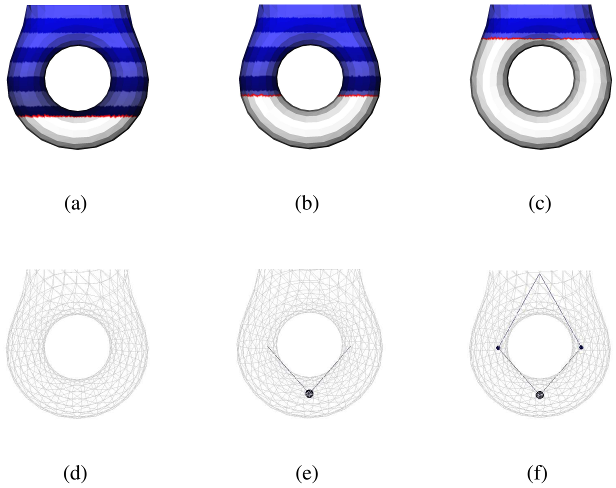


Figure 8. Bifurcation and junction contexts on a torus shape (height function).

Figure 7 gives an overview of the geodesic propagation in the metric space defined by f_q. The area depicted in white denotes the vertices visited by the Moore-Dijkstra algorithm, whereas the red areas correspond to the vertices belonging to $C(t)$. Moreover, each connected subset of $C(t)$ (red thin lines) is referred as a *discrete* level line.

This notion of *discrete* level line can be compared to the one of *topological ring*, presented by Mortara and Pantanè [11]. However, their graph construction strategy is highly dependent on their quotient function definition. Moreover, to benefit from the properties of geodesic distances, their quotient function computation needs a regular mesh refinement, which increases the overall time complexity of the approach and decreases its tolerance to mesh sampling variations.

In our algorithms, we analyze the connectivity evolutions of the *discrete* level lines at each iteration t of the geodesic propagation. Moreover, we construct simultaneously a dual Reeb graph (where connected components are represented with nodes and connectivity relations with edges), according to the following notions of topological variations: *bifurcations*, *junctions* and *terminations*.

Definition 7 (Bifurcation) *Let f be a scalar function defined on a connected triangulated surface T. Let $L(t) = \{l_1(t), l_2(t), ..., l_k(t)\}$ be the set of discrete level lines $l_i(t)$ at iteration t of f propagation and $v_t \in T$ the last visited vertex. A* bifurcation *happens in v_t iff:*

$$|L(t)| > |L(t-1)| \tag{9}$$

Figure 8 presents the bifurcation and junction contexts on a torus with the height function. In 8(a), $L(t)$ is composed of only one discrete level line, which splits in two in 8(b): a *bifurcation* is created on the graph (figure 8(e)). Similarly, we introduce the notion of *junction* as follows:

Definition 8 (Junction) *Let f be a scalar function defined on a connected triangulated surface T. Let $L(t) = \{l_1(t), l_2(t), ..., l_k(t)\}$ be the set of discrete level lines $l_i(t)$ at iteration t of f propagation and $v_t \in T$ the last visited vertex. A* junction *happens in v_t iff:*

$$\left\{ \begin{array}{l} |L(t)| < |L(t-1)| \\ \exists v_n \in Lk(v_t) \quad / \quad v_n \in C(t) \end{array} \right. \tag{10}$$

In 8(b), $L(t)$ is composed of two discrete level lines, which merge in one in 8(c): a *junction* is created on the graph (figure 8(f)). Finally, we introduce the notion of termination:

Definition 9 (Termination) *Let f be a scalar function defined on a connected triangulated surface T. Let $L(t) = \{l_1(t), l_2(t), ..., l_k(t)\}$ be the set of discrete level lines $l_i(t)$ at iteration t of f propagation and $v_t \in T$ the last visited vertex. A* termination *happens in v_t iff:*

$$\left\{ \begin{array}{l} |L(t)| < |L(t-1)| \\ v_n \notin C(t), \quad \forall v_n \in Lk(v_t) \end{array} \right. \tag{11}$$

At each step t of the geodesic propagation, we reconstruct $L(t)$ and apply needed topological variations on the graph (figures 8(e) and 8(f)), according to $|L(t)|$ evolutions (equations 9, 10 and 11), as illustrated in the next section.

5 Experimental results and comments

In this section, we present and comment on experimental results obtained with our method and discuss about its applications. Presented models are connected triangulated surfaces extracted from the Princeton Shape Benchmark database [15].

5.1 Time complexity

Let n be the number of vertices in the input mesh. As mentioned section 4.1.2, the feature point extraction is realized in $O(n \times log(n))$ steps. Moreover, f_q is computed within the feature point extraction. Consequently, the invariant quotient function computation time complexity is bounded by the Moore-Dijkstra algorithm and is realized in $O(|F| \times n \times log(n))$.

As for the graph construction algorithm, we model $C(t)$ with a binary priority heap, which means that addition and deletion of vertices are performed in $O(log(n))$. At a given iteration t of the geodesic propagation, in order to observe topological variations defined in 4.2, we re-construct each discrete level line with a recursive algorithm, in $O(n)$ steps. Consequently, an iteration t of the geodesic propagation is realized in $O(n + log(n))$ steps. Therefore, as the complete geodesic propagation takes n iterations, the overall complexity of the graph construction takes $O(n^2)$ steps.

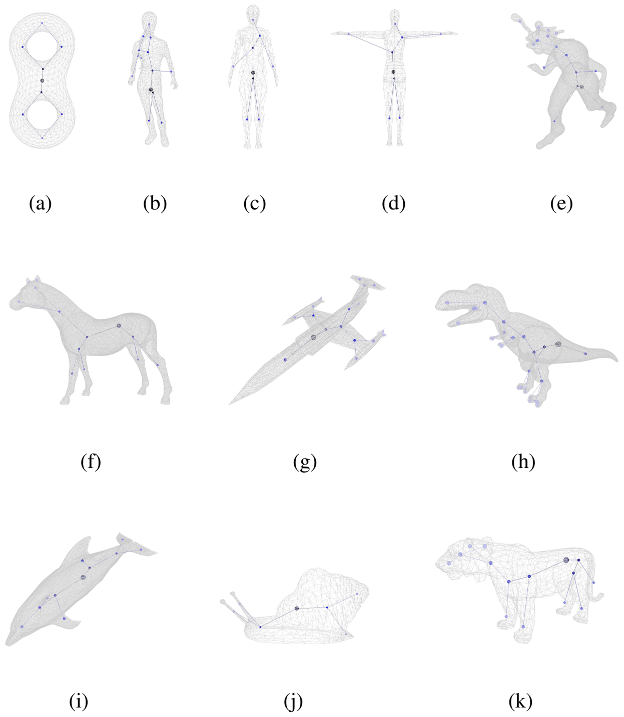


Figure 9. High level Reeb graphs of primitive and complex shapes.

Presented algorithms have been implemented in C language under GNU/Linux and experimented on a desktop PC with a 3GHz P4-CPU and 2 gigabytes of RAM. With this configuration, the computation of a high level Reeb graph takes 0.23 seconds for a 2 000 face model, 2 seconds for a 10 000 face model, 17 seconds for a 40 000 face model and 86 seconds for a 100 000 face model.

5.2 Immersion strategies

By definition, a Reeb graph is a graph representation only. Therefore, for many applications, it is mandatory to define an $\mathbb{R}^3$ immersion strategy.

In this paper, for illustration purpose, we use the following strategy: each node of the high level Reeb graph, corresponding to a whole connected component, is placed at the euclidean barycenter of its related component, as shown in figure 9. Notice that the root component, with bigger radius, represents the neighborhood of f global minimum. This area is isolated when the geodesic wavefront first collapses on itself, at the beginning of the algorithm.

Topological skeletons are a Reeb graph variant that can be obtained by placing a point at the euclidean barycenter of each discrete level line, as shown in figure 10(a). Such a skeleton is a particularly pertinent shape descriptor for automatic mesh animation. With this aim, since a point of the topological skeleton references a discrete level line, it is possible to move each vertex of the related level line with a parameterized displacement, so as to get a smooth movement of the whole connected component. In our experiments, we deformed models by applying recursively simple rotations to components, as shown in figure 10(b), but more sophisticated strategies can be used, like in [8].

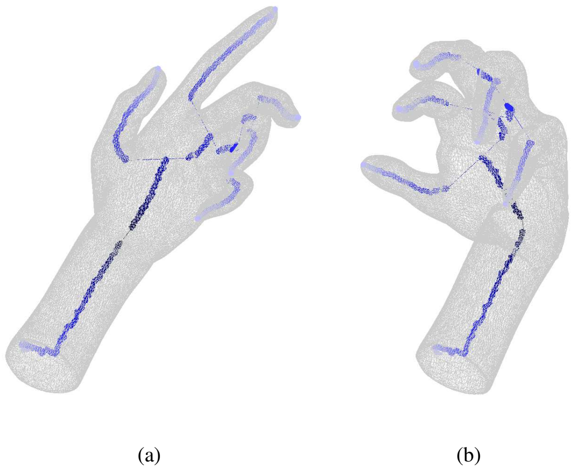


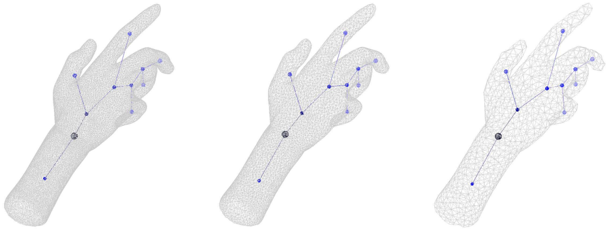
Figure 10. Skeletal representation of a high level Reeb graph (a) and an application to mesh deformation (b).

5.3 High level Reeb graph properties

Stability of the high level description First, as illustrated in figure 9, both complex and primitive shapes are well managed by our algorithm. Moreover, it can handle non-null genus surfaces correctly: in figure 9(b), the hair of the humanoid model merges with its back which forms consequently a cycle in the graph. Secondly, we can say that the high level Reeb graphs of models belonging to the same class (figure 9(b), 9(c), 9(d) and 9(e)) are quite similar, which denotes the stability of our algorithm.

Finally, our geodesic propagation within the metric space defined by the quotient function, combined with our critical point election strategy, leads to accurate results: none of the presented graphs reflects the presence of critical points on discontinuity areas. With traditional Reeb graph construction algorithms, the graph of the model 9(f) would have counted about ninety critical points (see critical points in figure 6(b) for comparison). With our approach, only meaningful topological variations are encoded.

Affine invariance Thanks to our choice of metric space (geodesic distances, with origins taken relatively to the mesh), it is quite obvious that our approach is invariant against geometrical transformations: translation, rotation and uniform scaling.



(a) 25 000 vertices.

(b) 5 000 vertices.

(c) 1 000 vertices.

Figure 11. High level Reeb graph construction algorithm robustness against mesh sampling variations.

Robustness to model pose Robustness to model pose can be observed in figures 9(c) and 9(d): the graphs are similar whether the arms of the humanoid are folded or not.

Robustness to mesh sampling Thanks to our notion of *discrete* level line, no hypothesis about mesh sampling is required. Therefore, we can state that our Reeb graph construction algorithm is tolerant against variations in mesh sampling, as shown in figure 11.

Thanks to those properties, high level Reeb graphs can be used in applications where invariance is fundamental. In shape retrieval for example, those graphs can be extended with geometrical attributes and shape comparison can be achieved with graph matching algorithms [6].

6 Conclusion and future works

In this paper, we introduced a novel method for the construction of invariant high level Reeb graphs, topological entities that afford a global understanding of shapes, with satisfactory execution times and without input parameters. This method is composed of three main steps. First, we extract feature points thanks to a robust and straightforward algorithm. Then, we use feature points as geodesic origins for the computation of an invariant quotient function f_q, used for topological analysis. Finally, we developed a new graph construction algorithm, based on a geodesic propagation in the metric space defined by f_q. It observes the connectivity evolutions of *discrete* level lines and provides meaningful Reeb graphs, which only encode significant topological variations. We illustrated the utility and the accuracy of our approach with an application to mesh deformation.

The invariance properties of presented graphs (geometrical transformations, model pose and mesh sampling) make them good candidates for various applications in computer graphics, like shape animation, retrieval, compression, etc.

In the future, we would like to refine our mesh decomposition scheme in order to propose high level shape descriptions of higher semantic interest.

References

[1] S. Biasotti, S. Marini, M. Mortara, and G. Patanè. An overview on properties and efficacy of topological skeletons in shape modelling. In *Shape Modeling International*, pages 245–254, 2003.

[2] G. J. Brostow, I. Essa, D. Steedly, and V. Kwatra. Novel skeletal representation for articulated creatures. In *European Conference on Computer Vision*, pages 66–78, 2004.

[3] H. Carr, J. Snoeylink, and U. Axen. Computing contour trees in all dimensions. In *ACM Symposium on Discrete Algorithms*, pages 918–926, 2000.

[4] A. Fomenko and T. Kunii. *Topological Modeling for Visualization*. Ed. Springer-Verlag, 1997.

[5] N. Gagvani and D. Silver. Parameter controlled volume thinning. *Graphical Models and Image Processing*, Volume 61:149–164, 1999.

[6] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, pages 203–212, 2001.

[7] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer (Pacific Graphics)*, Volume 21:865–875, 2005.

[8] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics, SIGGRAPH*, Volume 22:954–961, 2003.

[9] F. Lazarus and A. Verroust. Level set diagrams of polyhedral objects. Technical Report 3546, Institut National de Recherche en Informatique et en Automatique (INRIA), 1999.

[10] M. Morse. Relations between the critical points of a real function of n independant variables. *Transactions AM. Math. Soc.*, Volume 27:345–396, 1925.

[11] M. Mortara and G. Patanè. Affine-invariant skeleton of 3D shapes. In *Shape Modeling International*, pages 245–252, 2002.

[12] X. Ni, M. Garland, and J. Hart. Fair Morse functions for extracting the topological structure of a surface mesh. *ACM Transactions on Graphics, SIGGRAPH*, Volume 23:613–622, 2004.

[13] G. Reeb. Sur les points singuliers d'une forme de pfaff complètement intégrable ou d'une fonction numérique. *Comptes-rendus de l'Académie des Sciences*, Volume 222:847–849, 1946.

[14] A. Shamir. A formalization of boundary mesh segmentation. In *IEEE 2nd International Symposium on 3DPVT*, pages 82–89, 2004.

[15] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Shape Modeling International*, pages 167–178, 2004.

[16] S. Tarasov and M. Vyalyi. Construction of contour trees in 3D in O((n) log (n)) steps. In *Symposium on Computational Geometry*, pages 68–75, 1998.