The Visual Computer manuscript No.
(will be inserted by the editor)

Julien Tierny · Jean-Philippe Vandeborre · Mohamed Daoudi

Enhancing 3D Mesh Topological Skeletons with Discrete Contour Constrictions

Abstract This paper describes a unified and fully automatic algorithm for Reeb graph construction and simplification as well as constriction approximation on triangulated surfaces.

The key idea of the algorithm is that *discrete contours* – curves carried by the edges of the mesh and approximating the continuous contours of a mapping function – encode both topological and geometrical shape characteristics. Therefore, a new concise shape representation, *enhanced topological skeletons*, is proposed, encoding contours' topological and geometrical evolution.

Firstly, mesh feature points are computed. Then they are used as geodesic origins for the computation of an invariant mapping function that reveals the shape most significant features. Secondly, for each vertex in the mesh, its *discrete contour* is computed. As the set of *discrete contours* recovers the whole surface, each of them can be analyzed, both to detect topological changes and constrictions. Constriction approximations enable Reeb graphs refinement into more visually meaningful skeletons, that we refer as *enhanced topological skeletons*.

Extensive experiments showed that, without preprocessing stage, proposed algorithms are fast in practice, affine-invariant and robust to a variety of surface degradations (surface noise, mesh sampling and model pose variations). These properties make *enhanced topological skeletons* interesting shape abstractions for many computer graphics applications.

Keywords Shape abstraction · Topological skeletons · Feature points · Constrictions · Topology driven segmentation

J. Tierny
LIFL (UMR USTL/CNRS 8022), University of Lille, France
E-mail: tierny@lifl.fr

J.P. Vandeborre · M. Daoudi
GET / INT / TELECOM Lille 1
LIFL (UMR USTL/CNRS 8022), University of Lille, France
E-mail: {vandeborre, daoudi}@lifl.fr

1 Introduction

Polygonal mesh is a widely used representation of 3D shapes, mainly for exchange and display purposes. However, many applications in computer graphics need higher level shape descriptions as an input. Topological skeletons have shown to be interesting shape descriptions [4]. They benefit diverse fields like shape metamorphosis [25], deformation [8], retrieval [16], texture mapping [35], etc.

Many topological approaches study the properties of real valued functions computed over triangulated surfaces. Most of the time, those functions are provided by the application context, such as scientific data analysis [7]. When dealing with topological skeletons, it is necessary to define an invariant and visually interesting mapping function, which remains an open issue [4].

Moreover, traditional topological graph construction algorithms assume that all the information brought by the mapping function is pertinent, while in practice, this can lead to large graphs [24,9], encoding noisy details.

Finally, topological approaches cannot discriminate visually interesting sub-parts of identified connected components, like the phalanxes of a finger. This is detrimental to certain applications, such as mesh deformation.

In this paper, an original and unified framework is proposed to address the above issues. Given a closed connected triangulated surface T, feature points are firstly extracted (fig. 1(a)) in order to compute an invariant mapping function, noted f_m (fig. 1(b)), which reveals the shape most significant parts. Secondly, for each vertex in the mesh, we compute its *discrete contour*, a connected curve traversing it and locally minimizing f_m gradient. We show that a topological analysis of those *discrete contours* enables a pertinent Reeb graph construction and simplification (fig. 1(c)), without any input parameter. Finally, we show that a geometrical analysis of *discrete contours* can approximate constrictions on prominent components (fig. 1(d)), enabling the refinement of Reeb graphs into enhanced topological skeletons (fig. 1(e)).

This paper, which extends authors' previous work [32], is structured as follows. Firstly, we introduce topological skeleton related work. Secondly, we define our mapping func-

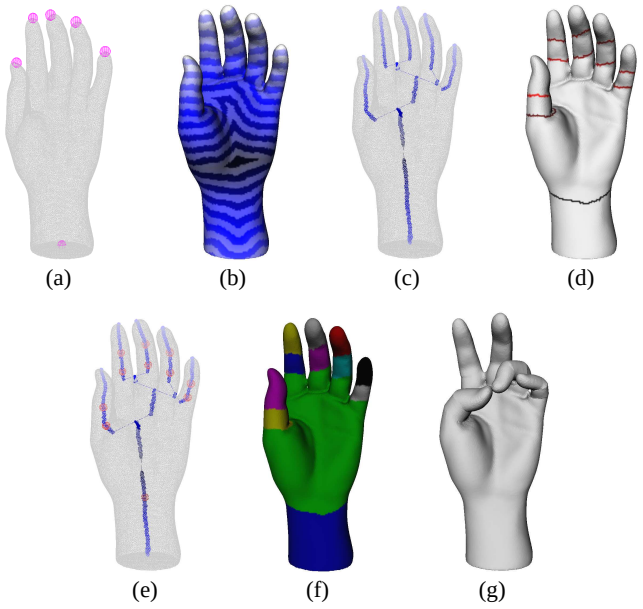


Fig. 1 Main steps of the framework. First, feature points are extracted (a). Then a mapping function (b) is contoured both to construct a topological skeleton (c) and to enhance it (e) with constrictions (d). Topology driven segmentation (f) and skeleton driven deformation (g) are two examples of application of the framework.

tion f_m. Thirdly, we present our algorithm for *discrete contour* computation, which is used both for the Reeb graph construction and simplification as well as the constriction approximation.

In particular, a special emphasis is given on the discrete contour constriction detection algorithm, which provides more robust and more accurate results than previous work [32]. Moreover, an extensive discussion on experimental results and framework evaluation is proposed. Finally, to demonstrate the usability of the presented method, two applications are discussed: topology driven mesh segmentation (fig. 1(f)) and skeleton driven mesh deformation (fig. 1(g)).

2 Related work

Several approaches have been explored for the decomposition of polygonal meshes into meaningful sub-components, to extract skeletal representations of shapes. They can be roughly classified into four categories: semantic-oriented segmentations, medial axis transforms, generalized potential fields and topological skeletons. Each one of them has its own advantages and drawbacks. For example, medial axis transform based methods [6,26] are commonly attributed the drawback of high sensitiveness to small surface perturbations, which is detrimental to many applications. On the contrary, generalized potential field based methods [33] provide more robust results at the cost of high computational times. In comparison to mesh segmentation based methods [18,17,2] and traditional skeleton extraction [33], topological approaches, based on Morse and Reeb graph theories [22,28], present the advantage to preserve the topological properties of the shape [4] (number of loops, number and relations between components, etc.). However, with regard to shape skeletons, we identify three main drawbacks in topological approaches, successively addressed in this paper.

Firstly, it is difficult to define an invariant and visually interesting mapping function. Secondly, constructing and transforming a topological graph into a manageable skeleton is not a trivial problem. Finally, topological approaches decompose a surface into connected sub-components only. This means that visually interesting sub-parts of identified connected components will not be discriminated: for example, a finger of a hand model will not be decomposed into phalanxes.

2.1 Mapping functions

Differential topology based approaches study the properties of real valued functions, that we refer as *mapping functions*, defined on input surfaces, either to construct Reeb graphs [30,10], contour trees [9], level set diagrams [20] or Morse complexes [24,7]. Those functions are often brought by the application context: terrain modeling [30], MRI analysis [9], molecular analysis [7], etc.

When dealing with topological skeletons, it is necessary to define a scalar function which satisfies invariance and stability constraints, and which also provides a topological description that highlights visually significant surface sub-components.

Lazarus and Verroust [20] introduced such a function, defined by the *geodesic distance* (the length of the shortest path between vertices) from a source vertex to any other vertex in the mesh. It leads to visually interesting results for natural objects because it is invariant to geometrical transformations and it is robust against variations in model pose [17]. Due to a lack of stability, within the framework of shape retrieval, Hilaga et al. [16] proposed to integrate this function all over the mesh. Unfortunately, from our experience, that function generates an important amount of critical points, configurations where the gradient of the function vanishes, which makes the construction of visually meaningful graphs more complex.

In our method, to reveal the shape most significant features, we focus on feature points. Feature points are mesh vertices located on extremities of prominent components [17]. Mortara and Patanè [23] proposed to select as feature points the vertices where Gaussian curvature exceeds a given threshold, but this cannot resolve extraction on constant curvature areas. Katz et al. [17] developed an algorithm based on multi-dimensional scaling in quadratic execution complexity. In this paper, we propose a robust and straightforward algorithm for feature point extraction (fig. 1(a)). Moreover, we use them as geodesic origins for the definition of our mapping function (fig. 1(b)). Such a function well reveals the most visually significant parts of the mesh, generating manageable critical point sets.

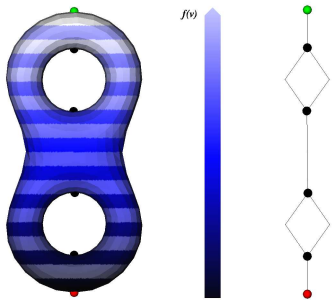


Fig. 2 Evolution of the level lines of the height function on a bi-torus, its critical points and its Reeb graph.

2.2 Graph construction and simplification

A Reeb graph [28] is a topological structure that encodes the connectivity relations of the critical points of a scalar function defined on an input surface. More formally, Reeb graphs are defined as follows:

Definition 1 (Reeb graph) *Let $f : M \rightarrow \mathbb{R}$ be a Morse function defined on a compact manifold M. The Reeb graph of f is the quotient space of f in $M \times \mathbb{R}$ by the equivalence relation $(p_1, f(p_1)) \sim (p_2, f(p_2))$, which holds iff:*

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

Figure 2 gives an example of a Reeb graph computed on a bi-torus with regard to the height function and well illustrates the fact that Reeb graphs can be used as skeletons.

Constructing a Reeb graph from a scalar function f computed on a triangulated surface first requires to identify the set of vertices corresponding to critical points. With this aim, several formulations have been proposed [11,31] to identify local maxima, minima and saddles, observing for each vertex the evolution of f at its direct neighbors. Several algorithms have been developed to construct Reeb graphs from the connectivity relations of these critical points [10,9], most of them in $O(n \times log(n))$ steps, with n the number of vertices in the mesh. However, they assume that all the information brought by the scalar function f is relevant [24,9]. Consequently, they assume that all the identified critical points are meaningful, while in practice, this hypothesis can lead to unmanageably large Reeb graphs. To overcome this issue, Ni et al. [24] developed a user-controlled simplification algorithm. Bremer et al. [7] proposed an interesting critical point cancellation technique based on a *persistence* threshold. Attene et al. [1] proposed a seducing approach, unifying the graph construction and simplification, but it is conditioned by a *slicing* parameter.

In this paper, we propose a discrete formulation of contours, connected subsets of level lines, which enables, without any input parameter, the construction of visually meaningful Reeb graphs (fig. 1(c)).

2.3 Constriction computation

Psychological research works [5] claim that the human visual system tend to segment complex objects along the narrowest and the most concave regions. Basing on this hypothesis, in order to improve topological skeletons' description quality, we focus on constrictions on surfaces.

Hétroy and Attali [15] define constrictions as simple closed curves, whose length is locally minimal. Recently, Hétroy [14] showed that constriction detection could be achieved by analyzing surface curvature.

In this paper, we propose to analyze the geometrical characteristics of discrete contours, and particularly their curvature, to approximate constrictions (fig. 1(d)), in order to decompose previously identified components into more visually interesting parts (fig. 1(e)).

3 Framework overview

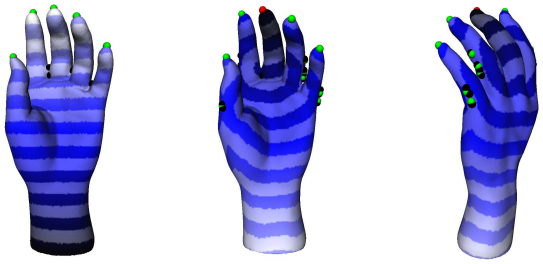
Given a closed connected triangulated surface T, we propose in this paper a unified method to decompose T into visually meaningful sub-parts, considering the topological and geometrical characteristics of *discrete contours*.

The algorithm proceeds in three stages. Firstly, mesh feature points are extracted (fig. 1(a)) in order to compute an invariant and visually interesting mapping function (fig. 1(b)), denoted f_m in the rest of the paper. Secondly, for each vertex in the mesh, we compute its *discrete contour*, a curve traversing it and approximating f_m continuous contour. Finally, as the set of *discrete contours* recovers the entire mesh, it is possible to analyze each contour characteristics, either to detect topological changes (fig. 1(c)) or to detect curvature transitions (fig. 1(d)).

Our scientific contribution resides in three points. *(i)* We propose a robust and straightforward algorithm for feature point extraction. *(ii)* We show that a *discrete contour* formulation enables, without re-meshing and without any input parameter, a pertinent Reeb graph construction, providing visually meaningful graphs, affine-invariant and robust to surface degradations. *(iii)* We show that the geometrical information brought by *discrete contours* enables the approximation of constrictions on prominent components and consequently Reeb graph refinement.

4 Feature point extraction

To compute visually meaningful topological skeletons, we first have to define a mapping function that will highlight the most significant parts of the mesh. To achieve this, we first focus on feature points, vertices located on the extremities of prominent components, because they provide a good overview of the shape structure [23,17].



(a) (b) (c)

Fig. 3 Euclidean (a) and geodesic (b, c) based mapping function critical points.

4.1 Algorithm overview

To extract feature points, we propose a quite straightforward algorithm, using differential topology tools. It is based on the key idea that feature points can be extracted analyzing some mapping function local extrema. In figure 3(a), the height function has been computed on a hand model. As the fingers are directed upwards, local maxima (in green) are identified at the extremities of related components and can be consequently referred as feature points. However, it is straightforward that the height function is not a relevant choice in the general case because it is dependent on the object orientation.

The function that maps a vertex to its geodesic distance to a source vertex seems much more appropriated because geodesic distances are affine-invariant and robust to variations in model pose. In figures 3(b) and 3(c), the geodesic distance function has been computed on the hand model, using the extremity of the middle finger (in red) as source vertex. The set of local extrema (minima and maxima, in red and green) actually contains the set of feature points (extremities of the fingers and the wrist).

However, figure 3(c) shows some local maxima in configurations which do not correspond to feature points (green points on the side of the little and the ring fingers). To discriminate local extrema that correspond to feature points from those which do not, we propose to realize a crossed analysis, using two geodesic based mapping functions – whose origins are the mesh most distant vertices – and to intersect the sets of their local extrema, as illustrated in figure 5.

4.2 Algorithm formulation

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 $\delta(v_i, v_j)$ as the normalized approximation of the geodesic distance from vertex v_i to v_j, normalized with regard to mesh global extrema.

Let v_{s_1} and v_{s_2} be the most geodesic distant vertices of a closed connected triangulated surface T, computed with the Tree Diameter algorithm [20]. In figure 4, v_{s_1} is located at the extremity of the wrist (fig. 4(a)) while v_{s_2} is located at the extremity of the middle finger (fig. 4(b)).

(a) v_{s_1}. (b) v_{s_2}.

Fig. 4 Most geodesic distant vertices of the hand model.

(a) E_1. (b) E_2. (c) $E_1 \cap E_2$.

Fig. 5 Feature point extraction overview.

Let f_{g_1} and f_{g_2} be two scalar functions defined on each vertex v of T, as follows:

$$f_{g_1}(v) = \delta(v, v_{s_1}) \tag{1}$$

$$f_{g_2}(v) = \delta(v, v_{s_2}) \tag{2}$$

Basing on the critical point classification proposed in [10], a *local minimum* is defined as a vertex such that all its direct neighbors have an upper function value. Reciprocally, we define a *local maximum* as a vertex such that all its direct neighbors have a lower function value.

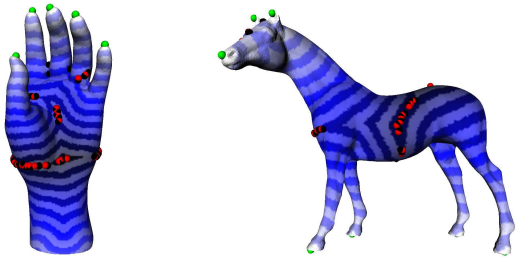
Let E_1 be the set of local extrema (minima and maxima) of f_{g_1} (in yellow in figure 5(a)) and E_2 be the set of local extrema of f_{g_2} (in cyan in figure 5(b)). Extremities of prominent components are configurations where f_{g_1} and f_{g_2} tend to an extremum (figs. 5(a) and 5(b)). Consequently, the set of feature points is both included in E_1 and E_2. Therefore, we define the set of feature points F of T (fig. 5(c)) as follows:

$$F = E_1 \cap E_2 \tag{3}$$

In practice, f_{g_1} and f_{g_2} local extrema which correspond to feature points do not appear exactly on the same vertices but in the same *geodesic neighborhood*. Therefore, the intersection constraint is relaxed as follows, with $\epsilon \in [0, 1]$ the radius of the *geodesic neighborhood* (geodesic distances are normalized):

$$v \in F \Longleftrightarrow \begin{cases} \exists v_{e_1} \in E_1 \quad / \quad \delta(v, v_{e_1}) < \epsilon \\ \exists v_{e_2} \in E_2 \quad / \quad \delta(v, v_{e_2}) < \epsilon \\ \delta(v, v_{f_i}) > \epsilon \quad \forall v_{f_i} \in F \\ \epsilon \in [0, 1] \end{cases} \tag{4}$$

From our experience, using only two geodesic mapping functions (f_{g_1} and f_{g_2}) and setting $\epsilon = 0.05$ give accurate results. The choice of such settings as well as the efficiency and properties of the presented algorithm will be discussed in the experiment dedicated section.



(a) $|F| = 6$, $|C| = 100$. (b) $|F| = 7$, $|C| = 92$.

Fig. 6 f_m mapping function computed on two standard models. The important number of identified critical points prevents standard algorithms from producing concise shape representations.

5 Mapping function definition

Mapping function definition depends on what is expected to be revealed. For example, for terrain modeling [30], the *height function* will present critical points over hills and valleys, providing consequently an appropriate topological description. When dealing with topological skeletons, it is necessary to define an invariant and visually interesting mapping function, which highlights the global structure of the object. Moreover, the mapping function should not generate an unmanageable set of critical points, in order to make the graph simplification easier. From our experience, this is not the case of the function presented in [16].

Firstly, to guarantee invariance to geometrical transformations and robustness against variations in model pose, geodesic distances are used. Secondly, to define a visually interesting mapping function, feature points are taken as origins for geodesic distance evaluations, because they provide a good overview of the object global structure [23].

Therefore, we propose the following mapping function, noted f_m in the rest of the paper, which computes in each vertex v of T the geodesic distance to the closest feature point:

$$f_m(v) = \frac{f_c(v) - min_{v \in T} f_c(v)}{max_{v \in T} f_c(v) - min_{v \in T} f_c(v)} \quad (5)$$

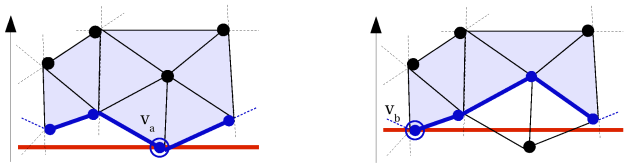
f_m is a normalized version of the function f_c, defined as follows ($f_c(v) \geq 0, \forall v \in T$):

$$f_c(v) = 1 - \delta'(v, v_c) \quad (6)$$

with v_c the closest feature point from v:

$$v_c \in F \quad / \quad \delta'(v, v_c) = min_{v_{f_i} \in F} \quad \delta(v, v_{f_i}) \quad (7)$$

Notice that f_m is invariant to uniform scaling (thanks to the normalization), rotation and translation (thanks to the use of geodesic distances). Figure 6 presents some computations of f_m over arbitrary shapes, the number of extracted feature points ($|F|$) and the number of critical points ($|C|$, identified according to the classification proposed in [10]). f_m has been defined so as it tends to maxima (in green) at feature points and it tends to minima (in red) at the center of the object.



(a) $\Gamma(v_a)$. (b) $\Gamma(v_b)$.

Fig. 7 Example of continuous (red) and discrete (blue) level lines (height function). The upper valued triangulation T^+ is colored in light blue.

As shown in figure 6, f_m generates an important number of critical points. Consequently, standard Reeb graph construction algorithms would create large graphs, counting as many nodes as critical points, which is a major issue for the extraction of meaningful topological skeletons. In the next section, we present a formulation of *discrete contours*, which enables a unified graph construction and simplification process.

6 Discrete contour computation

Defining contours of a real function f computed on a triangulated surface T is not a simple problem. In the continuous case, a level line $f^{-1}(f(p))$ is the set of points p_i such that $f(p_i) = f(p)$. Moreover, two points p_1 and p_2 belong to the same *contour* if they belong to the same connected component of $f^{-1}(f(p_1))$.

In the discrete case, for a given vertex $v \in T$, depending on T sampling, the set of vertices v_i such that $f(v_i) = f(v)$ is often reduced to the vertex v itself. With regard to definition 1, a correct Reeb graph could not be constructed from this formulation of discrete contours, because the conditions of the equivalence relation would rarely be satisfied.

To preserve contour topological properties in the discrete case, we define the *discrete level line* $\Gamma(v)$ associated to the vertex v as a curve computed along the edges of T which approximates by upper value the continuous level line $f^{-1}(f(v))$. More formally, it can be defined as follows:

Definition 2 (Discrete level line) *Let T^+ be the subset of a closed connected triangulated surface T such that $\forall v^+ \in T^+, f(v^+) \geq f(v)$. The discrete level line associated to the vertex v and noted $\Gamma(v)$ is the set of edges (and related vertices) belonging to T^+ such that each edge of $\Gamma(v)$ is adjacent to only one face of T^+.*

The *discrete level line* $\Gamma(v)$ is the boundary between the upper valued triangulation T^+ and the rest of T. Figure 7 shows *discrete level lines* traversing an arbitrary triangulation, with regard to the height function. Moreover, each connected subset of a *discrete level line* is referred as a *discrete contour*.

In particular, we define the *discrete contour* $\gamma(v)$ associated to the vertex v as the connected subset of $\Gamma(v)$ containing v. Notice that the more T will be dense, the more *discrete contours* will tend to continuous contours.

(a) (b) (c)

Fig. 8 Examples of discrete level lines (in red) on a 26 000 vertex mesh (f_m function). $\Gamma(v_{2\ 000})(a)$, $\Gamma(v_{10\ 000})(b)$ and $\Gamma(v_{20\ 000})(c)$ are respectively composed of 2, 3 and 6 *discrete contours*.

Algorithm 1 Discrete contour computation.

$Vt = \emptyset$
$Cd \leftarrow \{argmin_{v' \in T} f(v')\}$
while $Cd \neq \emptyset$ **do**
  $v \leftarrow argmin_{v' \in Cd} f(v')$
  $\Gamma(v) \leftarrow Cd$
  $\gamma(v) \leftarrow$ connected subset of Cd, containing v
  Remove v from Cd
  $Cd \leftarrow Cd \cup \{v$ neighbors, which are not in $Vt\}$
  Add v to Vt
end while

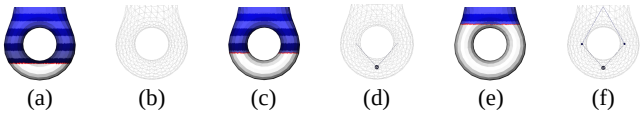
Discrete contours can be computed for the whole mesh using a step by step gradient ascent process, described in algorithm 1. This algorithm describes a propagation from f global minimum to f local maxima. It handles two heaps, respectively the set of visited vertices Vt and the set of candidate vertices for visit Cd. At each step, Cd surrounds Vt by upper value and corresponds to the discrete level line $\Gamma(v)$, with $v = argmin_{v' \in Cd} f(v')$.

In figure 8, examples of discrete level lines $\Gamma(v_i)$ are shown, at different iterations i of the algorithm. Vt vertex set is displayed in white and $\Gamma(v)$ is displayed in red. Visiting in a recursive fashion each vertex of $\Gamma(v)$ enables the identification of each of its connected subsets, and particularly $\gamma(v)$. In the next sections, discrete contours will be analyzed, both to detect topological changes and constrictions.

7 Topological analysis of discrete contours

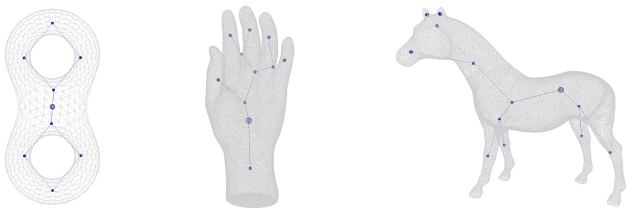
Standard Reeb graph construction algorithms need simplification in order to remove noisy details. In this section, we propose a unified algorithm for graph construction and simplification, based on the topological analysis of discrete contours. Following the definition 1 of a Reeb graph in the continuous case, we can state an analog equivalence relation in the discrete case between two vertices $v_1, v_2 \in T$, based on our notion of discrete contour:

$$(v_1, f(v)) \sim (v_2, f(v)) \Longleftrightarrow \begin{cases} v_1, v_2 \in \Gamma(v) \\ v_1, v_2 \in \gamma(v) \end{cases} \quad (8)$$



(a) (b) (c) (d) (e) (f)

Fig. 9 Bifurcation and junction contexts on a torus shape (height function).



(a) (b) (c)

Fig. 10 Dual Reeb graphs of primitive and complex shapes (f_m function).

v_1 and v_2 belong to the same connected component if they satisfy the above conditions. Therefore, at each iteration of the contour computation algorithm, each individual connected component of T, traversed by $\Gamma(v)$, can be identified.

Thus, topological changes can be detected observing the number $N_{\Gamma(v)}$ of connected subset of $\Gamma(v)$, as f evolves. We define three types of topological changes:

1. *bifurcations*: when $N_{\Gamma(v)}$ increases from iteration t to iteration $t+1$ ($\Gamma(v)$ splits in two contours from 9(a) to 9(c)),
2. *junctions*: when $N_{\Gamma(v)}$ decreases from t to $t+1$ and when several discrete contours merge (two contours merge in one from 9(c) to 9(e)),
3. *terminations*: when $N_{\Gamma(v)}$ decreases from t to $t+1$, without discrete contour merge.

Figure 10 shows several dual Reeb graphs obtained with this strategy, with regard to f_m (graph embedding in space will be detailed section 9).

The main contribution of our algorithm is that graph construction and simplification are performed at the same time, without input parameter or preprocessing.

If we compare figures 10 and 6, we notice that the dual Reeb graphs do not reflect the presence of noisy critical points (points in red in figure 6), because discrete level lines do not disconnect in those configurations.

Standard Reeb graph algorithms would have generated graphs counting as many nodes as critical points: 100 nodes for the hand model and 92 nodes for the horse model. In our approach, as contours do not disconnect in f_m non-smooth parts, only meaningful topological variations are encoded in the graph.

8 Geometrical analysis of discrete contours

Constriction approximations enable the subdivision of the branches of topological skeletons into more visually in-

(a) (b) (c)

Fig. 11 Several curvature estimations on the hand model: mean curvature (a), Gaussian curvature (b) and curvature index (c) [19]. The curvature index generates a more important contrast between convex and concave areas.

teresting parts, detecting potential articulations for example. For each *discrete contour* identified in the previous stage, its weighted curvature index is computed and local minima are identified as constrictions.

8.1 Topological constraint

Since constrictions are defined as closed curves, the analysis has to be restricted on *closed* discrete contours only. Considering each contour $\gamma(v)$ as a connected and non directed planar graph G, $\gamma(v)$ is a cycle, and consequently a closed curve, if the degree of all its vertices equals two. Therefore for each discrete contour of T reduced to a planar graph G, the degree of each of its vertex is computed and we only consider in the rest of our algorithms contours that satisfy the above property.

8.2 Contour curvature estimation

Many curvature computations have been proposed in the past [19,21]. In this paper, contour curvature is evaluated using the curvature index [19]. Firstly it is invariant to uniform scaling, which contributes to the stability of the algorithm. Secondly, experiments showed it provided more relevant results than Gaussian [32] or mean curvatures. The curvature index is computed for each vertex $v \in T$ as follows:

$$I_c(v) = \begin{cases} \frac{2}{\pi}\, arctan\, \frac{k_1+k_2}{k_1-k_2} & \text{if} \quad k_1 \neq k_2 \\ 0 & \text{if} \quad k_1 = k_2 \end{cases} \tag{9}$$

with k_1 and k_2 the principal curvatures in v, computed with the algorithm described in [21]. Figure 11 shows several curvature estimations on the hand model (the mean and the Gaussian curvatures have been displayed with logarithmic scales). Notice that the curvature index generates a more important contrast between convex (in light red) and concave (in dark red) areas, which benefits concave contour detection.

In order to estimate the curvature $\zeta(\gamma(v))$ associated to the discrete contour $\gamma(v)$, a weighted average of curvature indexes is computed as follows:

$$\zeta(\gamma(v)) = \frac{\sum_{\forall v_i \in \gamma(v)} \; I_c(v_i) \times (\mathcal{L}_{e_1}(v_i) + \mathcal{L}_{e_2}(v_i))}{2 \times \mathcal{P}(\gamma(v))} \tag{10}$$

where $\mathcal{P}(\gamma(v))$ stands for the perimeter of the contour $\gamma(v)$ and where $\mathcal{L}_{e_1}(v_i)$ and $\mathcal{L}_{e_2}(v_i)$ stand for the lengths of the edges adjacent to v_i on the contour. Vertex curvatures are weighted so as not to give too much importance to the densely sampled parts of the contour.

8.3 Component curvature curves

During the topological analysis of discrete contours, each contour is sequentially linked to its related node in the dual Reeb graph. Thus, each node of the graph is equipped with a sorted collection of discrete contours, sorted with regard to f_m. Computing $\zeta(\gamma(v))$ for each element of this sorted collection gives a *curvature curve*, an overview of $\gamma(v)$ curvature evolution as f_m evolves.

Curves shown in figure 12 (left column) give examples of such evolutions on some connected components of the hand model. f_m is reported on the X-axis while $\zeta(\gamma(v))$ is reported on the Y-Axis. Left values correspond to contour curvature at the basis of components while right values correspond to contour curvature at the extremity of components.

Curvature is a well-known noise sensitive entity. Consequently, to compute nice-looking discrete contour constrictions, we have to reduce high frequency noise in curvature curves. Reducing noise on a one-dimensional data set is a trivial signal-processing problem. This can be achieved by applying an ideal low-pass filter of cutoff frequency f_τ, defined by the following transfer function:

$$H(f_{\gamma(v)}) = \begin{cases} 1 \text{ if} & f_{\gamma(v)} \leq f_\tau \\ 0 \text{ if} & f_{\gamma(v)} > f_\tau \end{cases} \tag{11}$$

A filtered version of $\zeta(\gamma(v))$ is given by the following expression, where FT stands for the Fourier Transform:

$$\widehat{\zeta}(\gamma(v)) = FT^{-1}(H(f_{\gamma(v)}) \times FT(\zeta(\gamma(v)))) \tag{12}$$

As shown in figure 12, low-pass filtering smooths curvature curves and enables a clear discrimination of local minima and maxima. The f_τ parameter has been set experimentally to 8, as discussed in the experiment dedicated section.

8.4 Constriction selection

By definition, if $\widehat{\zeta}(\gamma(v))$ is positive, $\gamma(v)$ neighborhood is globally convex, otherwise it is concave. Constrictions appear on the narrowest, or the most concave, parts of a surface. Consequently, local negative minima are identified as contour constrictions. In figure 12(b), one negative minimum is identified and is marked as a constriction in figure

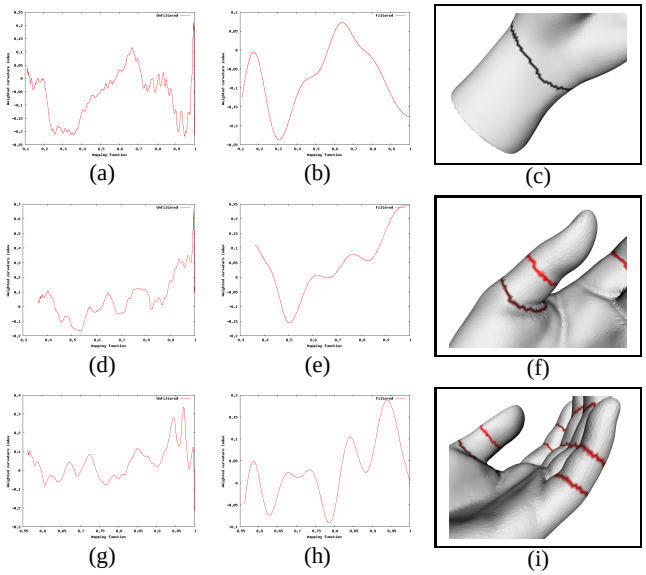


Fig. 12 Unfiltered (left column) and filtered (center column, $f_\tau = 8$) curvature curves and their related components. Local negative minima have been marked with gradations of red on the hand model views.

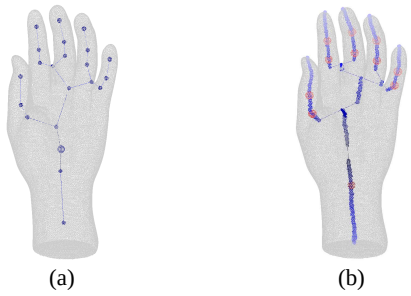


Fig. 13 Enhanced topological skeleton embeddings: graph embedding (a) and medial axis embedding (b).

12(c). Similarly, two negative minima are identified in figure 12(e) and are displayed in red in figure 12(f).

Constriction detection is motivated by a need to segment skeleton components into more meaningful subparts. Nevertheless, to keep the high-level interest of such a refinement, *over*-segmentation must be avoided. Roughly, if two contours that are negative minima of $\widehat{\zeta}(\gamma(v))$ are too close from each other (if the difference between their average value of f_m is lower than a given threshold $\Delta = 0.1$), only the most concave one is identified as a *meaningful constriction*.

9 Topological skeleton enhancement

In previous stages of the framework, a dual Reeb graph has been provided by a topological analysis of discrete contours. This graph is composed of nodes and edges which respectively represent identified connected components and adjacency relations between them. Each node of the graph references a collection of vertices as well as the collection of corresponding discrete contours. Moreover, constriction detection has been provided by a geometrical analysis of discrete contours. Consequently, for each node of the dual Reeb graph, its contour and vertex collections can be sub-divided using contour constriction as boundaries. Thus, each identified connected component of the mesh can be sub-divided into more visually meaningful sub-parts.

Basing on these decomposition processes, several embeddings of enhanced topological skeletons can be provided. For display purpose, a graph embedding (figure 13(a)) can be computed by placing a node at the euclidean barycenter of the related mesh patch.

For some applications, it is important to constraint the skeleton position strictly inside the object. Moreover, particularly for mesh deformation, a link between vertices and skeleton branches will be additionally needed. Most of the time, this association is performed computing for each vertex its closest point on the skeleton. Thanks to their theoretical foundations, enhanced topological skeletons are appropriate candidates. Firstly, a medial axis embedding (figure 13(b)) can be computed, placing a node at the euclidean barycenter of each discrete contour. Secondly, constriction approximation can be reported with a special node (in red in figure 13(b)), denoting a potential articulation in the object. As discrete contours have been defined relatively to a vertex, our algorithm provides a natural equivalence between the vertices of the mesh and the points of the skeleton.

10 Experiments and results

In this section, we present and comment on experimental results obtained with our method. The discussion will be laid out as follows. Firstly, experiments for parameter setting are detailed. Secondly, we focus on framework evaluation. To our knowledge, no ground-truth evaluation process has been proposed in the past for evaluating skeleton extraction accuracy. Nevertheless, we retained four evaluation criteria: time execution complexity, intrinsic properties (enumerated by Wu et al. [33]), robustness to surface degradation and applicative usability (section 11). Thirdly, framework limitations will be detailed.

Models presented in this discussion are closed connected triangulated surfaces extracted from the *Princeton Shape Benchmark* [29] and the I.N.R.I.A. Gamma Research Group [13] repositories.

10.1 Parameter setting

Feature point extraction is the first stage of the pipeline that requires a threshold value setting. This parameter (ϵ) stands for the radius of geodesic neighborhoods during the geodesic based mapping function extrema intersection process (extrema representing a same feature point might not appear on the same vertex but in the same neighborhood). If ϵ is too small, extrema will have to be very close to be merged; consequently, some feature points might be missed.

Model	ϵ = 0.01	ϵ = 0.02	ϵ = **0.05**	ϵ = 0.1	ϵ = 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 1 Number of feature points with different ϵ parameter values.

Model	N_G = **2**	N_G = 3	N_G = 4	N_G = 5	N_G = 10
Sphere	**2**	2	2	2	2
Bottle	**2**	2	2	2	2
Bird	**7**	6	5	5	4
Dinosaur	**7**	5	5	4	2
Horse	**7**	6	2	2	2
Hand	**6**	1	0	0	0

Table 2 Number of feature points when the number N_G of geodesic mapping functions for intersection increases. Intersecting more and more sets reduces the number of common elements.

Model	f_τ = 1	f_τ = 5	f_τ = **8**	f_τ = 10	f_τ = 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 3 Number of discrete contour constrictions with different f_τ parameter values. Most visually conforming results have been obtained with $f_\tau = 8$.

Model	Δ = 0	Δ = 0.05	Δ = **0.1**	Δ = 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 4 Number of discrete contour constrictions with different Δ parameter values. Setting $\Delta = 0.1$ avoids over segmentation on concerned objects (dinosaur and horse models) without affecting constriction selection on others.

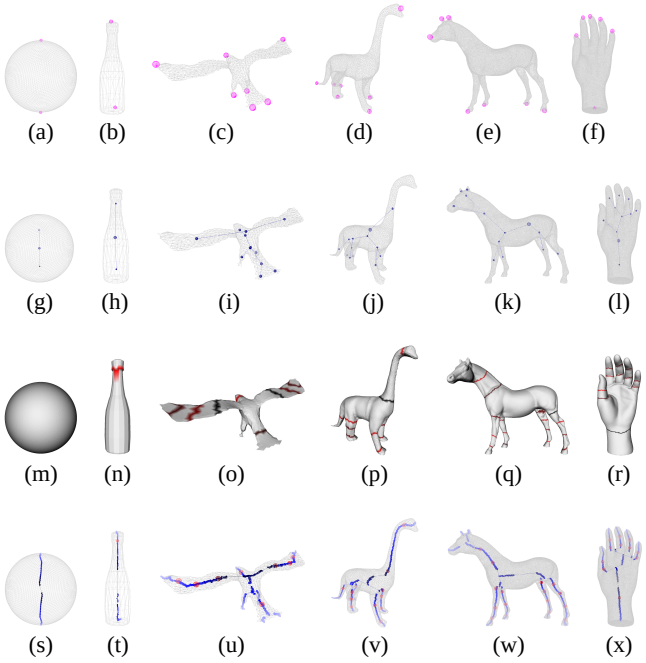


Fig. 14 Feature points, dual Reeb graph, constriction approximations and enhanced topological skeleton of simple and complex objects.

On the contrary, if it is too high, distant extrema will tend be merged into the same feature point; thus some feature points might represent several protrusions. Table 1 shows that the number of extracted feature points is quite stable when ϵ varies. The most visual perception conforming results have been obtained setting $\epsilon = 0.05$ (bold column, see figure 14 for graphical results).

The number of geodesic based mapping functions used for intersection (noted N_G) can also be discussed as a parameter of the approach. Table 2 shows experiments with a variant of proposed algorithm, using a varying number of mapping functions. It shows that the more mapping functions are intersected, the less feature points are identified, as a consequence of the intersection process of multiple sets.

Discrete contour constriction detection is conditioned by the cutoff frequency f_τ of the low-pass filter applied on curvature curves. This filter's objective is to decrease curvature high frequency noise influence. Setting $f_\tau = 8$ has shown to result in nice-looking constrictions (see figure 14 and table 3). Moreover, constriction selection is also conditioned by a threshold parameter (Δ) which denotes the minimal acceptable distance between two consecutive constrictions. Table 4 shows it is not a critical parameter though it helps avoiding over-segmentation on two models.

As a conclusion, tables show that most of those parameters are not critical for the stability of the overall method and that proposed values (used for figures 14 and 15) give visual perception conforming results.

10.2 Time complexity

Given an input closed connected triangulated surface T, let n be the number of vertices in T. Feature point extraction is performed in $O(n \times log(n))$ steps. f_m is computed in $O(|F| \times n \times log(n))$ steps with $|F|$ the number of identified feature points. Notice that f_m has a lower computational cost than the function proposed in [16] ($|F|$ rarely exceeds 20). Each discrete contour computation takes $O(log(n)+n)$. Therefore, as contours are computed for each vertex in T, the overall discrete contour computation takes $O(n^2)$ steps. Topological and geometrical analyses are more straightfor-

Model	Faces	Feature pts	Constrictions	Time (s.)
Sphere	8 000	2	0	6.342
Bottle	520	2	1	0.045
Bird	2 000	7	8	0.425
Dinosaur	10 000	7	13	3.490
Horse	40 000	7	16	33.793
Hand	52 000	6	11	109.194

Table 5 Computation times.

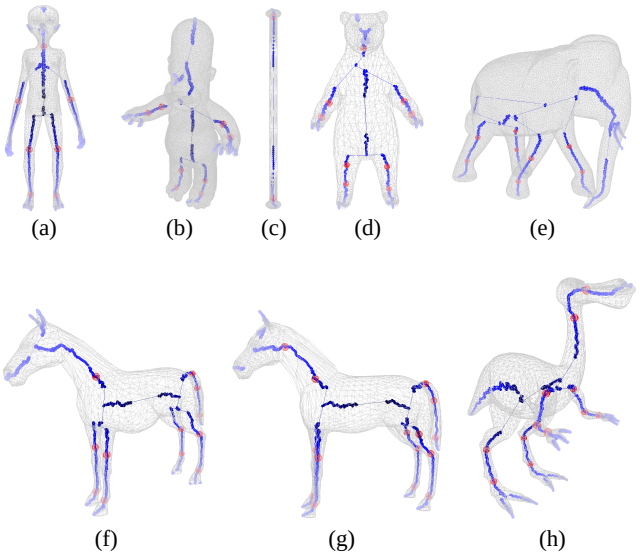


Fig. 15 Enhanced topological skeleton of various objects.

ward. Topological analysis is performed in $O(n)$ steps. Curvature curves are computed in $O(n)$ steps. Their smoothing is realized in $O(n \times log(n))$, using the Fast Fourier Transform algorithm. Consequently, we can state that the overall complexity of our method is bounded by the discrete contour computation, which takes $O(n^2)$ steps in the worst case.

Presented algorithms have been implemented in C language under GNU/Linux and experimented on a desktop PC with a 3 GHz P4-CPU and 2 gigabytes of RAM. Table 5 shows the computation times corresponding to the models presented in figure 14.

For comparison, latest constriction detection algorithms [14] run in 80 seconds for a $10\ 000$ faced model. Domain connected graph [33] are computed in more than $1\ 900$ seconds for such models. 3D mesh segmentation based on feature points and core extraction [17] takes 28 seconds for $4\ 000$ faced models. Notice that our overall method has a significantly lower running time than these state-of-the-art methods for equivalently sampled meshes.

10.3 Discussion

Intermediary results and topological skeletons are shown in figures 14 and 15. On these illustrations, the reader can notice that enhanced topological skeletons concisely encode

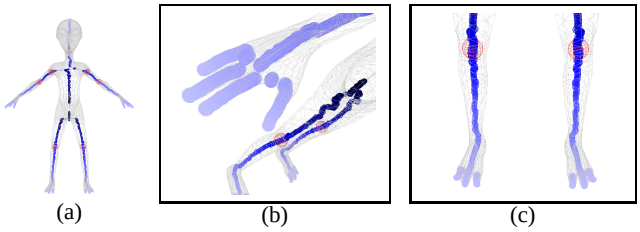
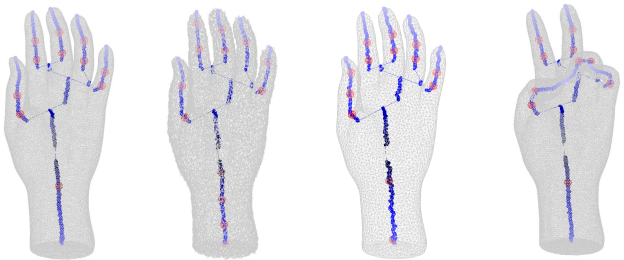


Fig. 16 Zooming in the alien model, we can see that feature points have been extracted on each finger and toe, thus creating corresponding branches in the skeleton.

both shape topology (Reeb graph, in blue) and geometry (constrictions, in red). On the one hand, the topological analysis results in the identification of the most significant topological components only, preserving the topological properties of the shape: model in figure 15(e) is a 1-genus surface and its skeleton consequently contains a cycle. On the other hand, the geometrical analysis enables the subdivision of these topological components into more visually meaningful subparts, increasing the description quality of traditional topological skeletons (subdividing humanoid models' limbs along potential articulations).

According to Wu et al. [33], a shape skeleton is expected to respect certain intrinsic properties: simplicity, stability, meaningfulness, neutrality and hierarchy. Firstly, as illustrated in figure 13(a), enhanced topological skeletons can be collapsed into very concise representations. Secondly, the stability of the presented method can be observed in figures 15(f) and 15(g), where two different horse models have nearly identical skeletons. More generally, the overall robustness of the framework will be addressed later on this discussion. Thirdly, reader can notice that no noisy details are encoded in the skeletons, which shows the meaningfulness of the framework. Fourthly, the neutrality of the skeleton is provided by the medial axis embedding we proposed in section 9. Fifthly, a natural skeleton hierarchy is provided by the mapping function f_m: skeleton subparts at the center of the object, at the root of the hierarchy, have a low function value (denoted in dark blue in figures 14 and 15) whereas extremity parts have a high function value (denoted in light blue in figures 14 and 15). Consequently, we can state that enhanced topological skeletons respect the intrinsic properties of shape skeletons [33], and consequently provides a good description of 3D meshes.

Feature points have been extracted in the past. Contrary to the algorithm presented by Mortara et al. [23], the presented method resolves extraction on constant curvature areas, such as a sphere where two poles are extracted (figure 14(a)). Moreover, the algorithm presented by Katz et al. [17] runs in quadratic execution complexity while our algorithm needs $O(n \times log(n))$ steps, with n the number of vertices in the mesh. At last, we can state that our feature point extraction algorithm leads to visual perception conforming results, as shown in figures 16(b) and 16(c), where feature points have been extracted on the extremity of each finger and toe.



(a) (b) (c) (d)

Fig. 17 Algorithm robustness against surface degradations: original (a), surface noise (b), under sampling (c) and user deformations (d).

As for the topological skeleton construction algorithm, reader can notice that no noisy details have been encoded in the skeletons, because discrete contours do not disconnect in f_m non-smooth configurations (denoted with red and black critical points in figure 6). Contrary to traditional Reeb graph simplification algorithms, no noise-corrective threshold is required.

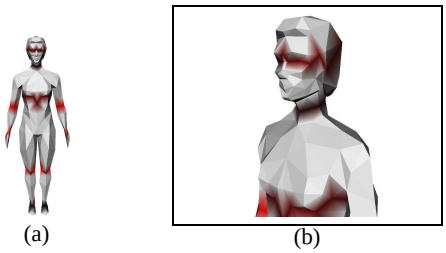
Discrete contour constriction detection also leads to visual perception conforming results, identifying the most significant surface bottlenecks (such as humanoid models' elbows and knees for example). As shown in figures 14(n) and 14(o) proposed algorithms leads to acceptable results even with coarsely designed objects, even on strongly tubular components. Notice that it also behaves correctly with primitive shapes, such as the sphere (fig. 14(m)), where no constriction is identified. Moreover, contrary to previous works [15, 14], only *meaningful* constrictions are extracted.

10.4 Robustness against surface degradation

As f_m is based on normalized geodesic distance evaluation, the algorithm is invariant to geometrical transformations (translation, rotation and uniform scaling). The following experiments try to estimate our framework's stability against more complex transformations.

The first experiment deals with surface noise (fig. 17(b)), denoting scanning acquisition noise for example. Each vertex of the input triangulation has been moved randomly in a box whose volume corresponds to 1% of the bounding box of the overall object. Reader can notice that the topological description (in blue) remains unchanged while the geometrical description (in red) slightly varies. Notice that such a noise dramatically affects the surface and is known to be critical for Medial Axis Transform or segmentation based methods.

In comparison to [1], no re-meshing pre-process is required in our framework. To evaluate our method dependence on the triangulation of the surface, we propose in the second experiment to divide by 5 the number of vertices in the mesh, as shown in figure 17(c). In a similar way to previous experiment, surface under-sampling slightly affects the constriction detection process only.



(a) (b)

Fig. 18 False positive constriction on the head of a coarsely designed humanoid model. As the eye areas are deeply concave, associated discrete contour is incorrectly identified as a constriction.

The third experiment purpose is to evaluate enhanced topological skeletons robustness against variations in model pose. This property benefits applications such as surface comparison. Geodesic metrics are commonly attributed the property to be invariant to variations in model pose though it is rarely experimented. In figure 17(d), the hand model has been deformed using rigid rotations (as detailed in section 11). Then, the enhanced topological skeleton of the deformed surface has been computed. Notice not only the topological but also the geometrical analyses behaved nearly identically: only one constriction is missing in the little finger and the other ones kept precisely their original position. Thus, we can conclude from this experiment that the presented method is also robust against variations in model pose.

10.5 Framework limitations

In figure 15(d), the bear model seems symmetrical while it is not rigorously. This results in a skeleton whose structure is not symmetrical. Generally speaking, topological skeletons (because of their theoretical foundations) tend to amplify models' asymmetry. We believe that for future work, capturing the notion of *perceived symmetry* could increase the descriptive quality of the framework, inserting in an artificial way some symmetry in the skeleton if needed. In this sense, recent results in geometry processing for symmetry detection [27] worth being investigated. This property could benefit applications such as skeleton-based surface similarity estimation, which often uses asymmetry sensitive graph-matching algorithms.

Moreover, proposed method assumes that constrictions appear along extracted discrete contours. Even if it is verified in practice for most of tested objects (see figures 14 and 15), this is a strong hypothesis. An undesired consequence of this hypothesis is presented in figure 18, where a contour passing through a deeply concave area is incorrectly identified as a constriction. Reciprocally, expected constrictions that are not located along discrete contours will not be identified. Graph-cut based optimization algorithms could bring a solution to this issue, moving constriction approximations to their exact location, helping the constriction selection.

11 Examples of application

In previous section, our framework has been evaluated according to its time execution complexity, its descriptive properties and its robustness against surface degradation. In this section, we focus on its applicative usability as its last evaluation criteria and propose two applications: topology driven mesh segmentation and skeleton driven mesh deformation.

11.1 Topology driven mesh segmentation

Mesh segmentation roughly consists in cutting up a surface into meaningful sub-parts [2]. Similarly to skeleton extraction, it benefits a large spectrum of applications. Recently, the need for *compatible segmentation* – which means segmenting identically two meshes representing the same class of object – has been expressed [17]. Contrary to geometrical low-level based approaches [34], a solution to this issue could reside in driving the segmentation using high-level shape information, such as symmetry or topology. In this sense, Berreti et al. [3] proposed to use the topological description provided in [16] to compute simple, but similar segmentations. In this paragraph, we propose to exploit this idea and to take advantage of the higher descriptive quality of our skeletons (in comparison to traditional topological skeletons) to improve topology driven mesh segmentation.

Proposed algorithm is summed up in figure 19. First, for each node of the graph, its related surface patch is displayed with a distinctive color. This results in a raw over-segmentation (fig. 19(b)). Then, the graph is simplified (fig. 19(c)) and the final segmentation (fig. 19(d)) is computed using the following two node-merging heuristics:

1. **Delete thin patches:** if the degree of processed node is greater than 2, then merge it with its adjacent node of biggest area, except if a constriction separates them;
2. **Select the most concave boundaries:** if the degree of processed node is equal to 2 and it is the first node of a branch and its last contour has a $\widehat{\zeta}(\gamma(v))$ value lower than a given threshold $\widehat{\zeta}_{min}$ (fixed to -0.1), then merge it with its parent node (its adjacent node of lowest average f_m value).

More results are shown in figure 20. On these illustrations, objects are segmented into core and limbs and limbs are segmented along constrictions. Notice that this algorithm benefits our skeleton extraction framework properties, and particularly its robustness against surface degradation.

11.2 Skeleton driven mesh deformation

User defined 3D model deformations often need tricky mesh editing operations. Shape skeletons have shown to be user-friendly shape abstraction for surface deformation [8, 18, 33]. Basically, the user grabs the branches of the skeleton and move them to produce the desired deformation. Contrary to conventional topological skeletons, our skeletons are enhanced with constrictions which mostly corresponds to articulations. Consequently, the user can employ these special points of the skeleton (in red in figure 21) as natural rotation reference points. Once a branch of the skeleton has been rotated, the corresponding rotation matrix is constructed. Then, for each vertex of the related surface patch, its new position is computed multiplying its original position vector by the rotation matrix. Figure 21 shows that nice-looking deformations can be obtained with only a few user interactions, thanks to the high-level description provided by enhanced topological skeletons.

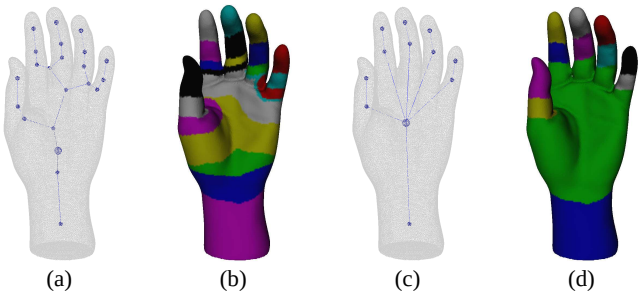


Fig. 19 Topology driven segmentation overview: a raw segmentation (b) is obtained from the enhanced topological skeleton (a), then a node merging algorithm (c) completes the process (d).

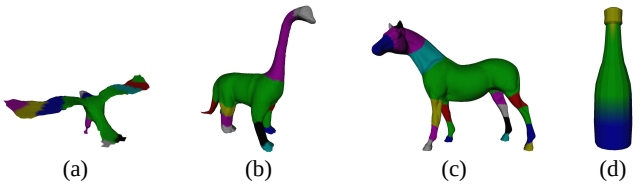


Fig. 20 Topology driven segmentation results for some test objects. Proposed algorithm identifies core and limbs, subdividing limbs along deeply concave areas.

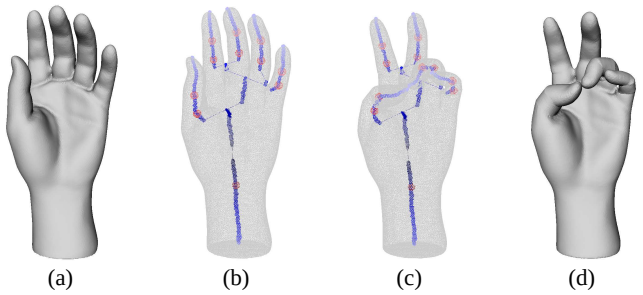


Fig. 21 Application to mesh deformation. The user directly grabs the branches of the skeleton (b) and rotates them using constrictions (in red) as articulations (c). Thanks to the equivalence relation between the vertices of the mesh and the points of the skeleton, deformations are directly reflected on the object (d).

12 Conclusion and future work

In this paper, we presented a new concise shape abstraction, *enhanced topological skeleton*, that encodes both topological and geometrical evolutions of some mapping function contours. We also proposed a unified and fully automatic algorithm for its extraction. First, it computes a Reeb graph. Then it refines it using discrete contour constriction as boundaries. To the authors' knowledge, this is the first approach that unifies Reeb graph construction and constriction computations.

Three scientific contributions are given. Firstly, we proposed a robust and straightforward algorithm for feature point extraction. It enables the computation of an invariant mapping function which reveals well the shape most significant features. Secondly, we presented an algorithm for *discrete contours* computation. We showed that a topological analysis of these *discrete contours* enables a unified Reeb graph construction and simplification process. Finally, we showed that a geometrical analysis of the *discrete contours* provides visual perception conforming constriction approximations, enabling the refinement of the traditional Reeb graphs into more visually meaningful skeletons.

Moreover, we presented an extensive evaluation discussion of experimental results, according to four criteria. Firstly, we have shown that *enhanced topological skeletons* respect shape skeleton intrinsic descriptive properties [33], improving traditional topological skeletons description quality. Secondly, we have shown that the presented algorithm runs significantly faster than state-of-the-art methods. Thirdly, we have shown that *enhanced topological skeletons* are affine-invariant and robust to a variety of surface degradations, such as surface noise, mesh sampling variation and surface deformation. At last, to evaluate the usability of the presented framework, we developed two applications related to topology driven mesh segmentation and skeleton driven mesh deformation.

Some limitations have been enumerated along the discussion. In future work, we would like to capture the notion of *perceived symmetry* in order to extract symmetrical skeletons if the objects seem roughly symmetrical. We believe this could benefit many applications, like surface similarity estimation. As a long term perspective, we would like to use *enhanced topological skeletons* as a shape abstraction for mesh editing operations within the framework of *modeling by example* systems [12].

Acknowledgements The authors would like to thank Pacific Graphics anonymous reviewers for their relevant comments, which helped them improving the framework. This work is partially supported by the European Network of Excellence *Delos* No. 507618 – http://www.delos.info.

References

1. Attene, M., Biasotti, S., Spagnuolo, M.: Shape understanding by contour-driven retiling. The Visual Computer **19**, 127–138 (2003)
2. Attene, M., Katz, S., Mortara, M., Patané, G., Spagnuolo, M., Tal, A.: Mesh segmentation: A comparative study. In: Shape Modeling International, pp. 14–25 (2006)
3. Berreti, S., Del Bimbo, A., Pala, P.: Partitioning of 3D meshes using Reeb graphs. In: IEEE ICPR, pp. 19–22 (2006)
4. Biasotti, S., Marini, S., Mortara, M., Patanè, G.: An overview on properties and efficacy of topological skeletons in shape modelling. In: Shape Modeling International, pp. 245–254 (2003)
5. Biederman, I.: Recognition-by-components: A theory of human image understanding. Psychological Review **94**, 115–147 (1987)
6. Blum, H., Nagel, R.N.: Shape description using weighted symmetric axis features. Pattern Recognition **10**, 167–180 (1978)
7. Bremer, P.T., Edelsbrunner, H., Hamann, B., Pascucci, V.: Topological hierarchy for functions on triangulated surfaces. IEEE Transactions on Visualization and Computer Graphics **10**, 385–396 (2004)
8. Capell, S., Green Seth abd Curless, B., Duchamp, T., Popović, Z.: Interactive skeleton-driven dynamic deformations. ACM Transactions on Graphics **21**, 586–593 (2002)
9. Carr, H., Snoeyink, J., de Panne, M.V.: Simplifying flexible isosurfaces using local geometric measures. In: IEEE Visualization, pp. 497–504 (2004)
10. Cole-McLaughlin, K., Edelsbrunner, H., Harer, J., Natarajan, V., Pascucci, V.: Loops in Reeb graphs of 2-manifolds. In: Symposium on Computational Geometry, pp. 344–350 (2003)
11. Edelsbrunner, H., Mücke, E.P.: Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. ACM Transactions on Graphics **9**, 66–104 (1990)
12. Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., Dobkin, D.: Modeling by example. ACM Transactions on Graphics **23**, 652–663 (2004)
13. Gamma Research Group Repository, I.N.R.I.A.: URL http://www-c.inria.fr/gamma/disclaimer.php
14. Hétroy, F.: Constriction computation using surface curvature. In: Eurographics, pp. 1–4 (2005)
15. Hétroy, F., Attali, D.: From a closed piecewise geodesic to a constriction on a closed triangulated surface. In: Pacific Graphics, pp. 394–398 (2003)
16. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.: Topology matching for fully automatic similarity estimation of 3D shapes. In: SIGGRAPH, pp. 203–212 (2001)
17. Katz, S., Leifman, G., Tal, A.: Mesh segmentation using feature point and core extraction. The Visual Computer **21**, 865–875 (2005)
18. Katz, S., Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Transactions on Graphics **22**, 954–961 (2003)
19. Koenderink, J.J., van Doorn, A.J.: Surface shape and curvature scales. Image and Vision Computing **10**, 557–565 (1992)
20. Lazarus, F., Verroust, A.: Level set diagrams of polyhedral objects. Tech. Rep. 3546, Institut National de Recherche en Informatique et en Automatique (INRIA) (1999)
21. Meyer, M., Desbrun, M., Schrder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and Mathematics, pp. 33–57 (2002)
22. Morse, M.: Relations between the critical points of a real function of n independant variables. Transactions AM. Math. Soc. **27**, 345–396 (1925)
23. Mortara, M., Patanè, G.: Affine-invariant skeleton of 3D shapes. In: Shape Modeling International, pp. 245–252 (2002)
24. Ni, X., Garland, M., Hart, J.: Fair Morse functions for extracting the topological structure of a surface mesh. ACM Transactions on Graphics **23**, 613–622 (2004)
25. Nieda, T., Pasko, A., Kunii, T.L.: Detection and classification of topological evolution for linear metamorphosis. The Visual Computer **22**, 346–356 (2006)
26. Ogniewicz, R., Ilg, M.: Voronoi skeletons: Theory and applications. In: IEEE Computer Vision and Pattern Recognition, pp. 63–69 (1992)
27. Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., Funkhouser, T.: A planar-reflective symmetry transform for 3D shapes. ACM Transactions on Graphics **25**, 549–559 (2006)

28. Reeb, G.: Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. Comptes-rendus de l'Académie des Sciences **222**, 847–849 (1946)
29. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton shape benchmark. In: Shape Modeling International, pp. 167–178 (2004)
30. Shinagawa, Y., Kunii, T.L., Kergosien, Y.L.: Surface coding based on morse theory. IEEE Computer Graphics and Applications **11**, 66–78 (1991)
31. Takahashi, S., Ikeda, T., Shinagawa, Y., Kunii, T.L., Ueda, M.: Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data. Computer Graphics Forum **14**, 181–192 (1995)
32. Tierny, J., Vandeborre, J.P., Daoudi, M.: 3D mesh skeleton extraction using topological and geometrical analyses. In: Pacific Graphics, pp. 85–94 (2006)
33. 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**, 117–135 (2006)
34. Yamauchi, H., Gumhold, S., Zayer, R., Seidel, H.P.: Mesh segmentation driven by gaussian curvature. The Visual Computer **21**, 649–658 (2005)
35. Zhang, E., Mischaikow, K., Turk, G.: Feature-based surface parametrization and texture mapping. ACM Transactions on Graphics **24**, 1–27 (2005)

Julien Tierny received the M.Sc. degree (summa cum laude) in Computer Science from the University of Lille (USTL) along with an engineer degree from TELECOM Lille 1 in 2005. He is currently a Ph.D. candidate within the Computer Science laboratory of the University of Lille (UMR USTL/CNRS 8022). He is also a teaching assistant in the Computer Science Department of the University of Lille. His research interests include shape modeling, shape similarity estimation, geometry processing and their applications.

Jean-Philippe Vandeborre received the M.S. dergee in 1997 and the Ph.D. degree in Computer Science in 2002, both from the University of Lille (UTSL), France. Currently, he is an Associate Professor at TELECOM Lille 1 (a graduate engineering school) in the Computer Science and Network Department. He is also a member of the Computer Science laboratory of the University of Lille (UMR USTL/CNRS 8022). His current research interests are mainly focused on three-dimensional model analysis, and include multimedia indexing and retrieval from content and their applications.

Mohamed Daoudi received the Ph.D. degree in Computer Science from the Lille's Science and Technology University (USTL), France, in 1993 and Habiltation à Dirger des Recherches (HDR) from the University of Littoral, France, in 2000. He is currently Professor at the INT/TELECOM Lille 1, Department of Computer Science. He was the head of the MIIRE research group of LIFL (UMR USTL-CNRS 8022), a research laboratory in the Computer Science of the USTL. His research interests include pattern recognition, image processing, invariant representation of images and shapes, three-dimensional analysis and retrieval and more recently 3D face recognition. He has contributed to more than 70 articles in journals and conference proceedings. Dr. Daoudi has served as a Program Committee member for the International Conference on Pattern Recognition (ICPR) in 2004 and the International Conference on Multimedia and Expo (ICME) in 2004 and 2005. He is a frequent reviewer for IEEE Transactions on Pattern Analysis and Machine Intelligence and for Pattern Recognition Letters.