Shape understanding by contour-driven retiling

M. Attene,
S. Biasotti,
M. Spagnuolo

Istituto per la Matematica Applicata e le Tecnologie
Informatiche, Consiglio Nazionale delle
Ricerche, Via De Marini 6, 16149 Genova, Italy
E-mail: {attene, silvia, spagnuolo}@ge.imati.cnr.it

Given a triangle mesh representing a closed manifold surface of arbitrary genus, a method is proposed to automatically extract the Reeb graph of the manifold with respect to the height function. The method is based on a slicing strategy that traces contours while inserting them directly in the mesh as constraints. Critical areas, which identify isolated and non-isolated critical points of the surface, are recognized and coded in the extended Reeb graph (ERG). The remeshing strategy guarantees that topological features are correctly maintained in the graph, and the tiling of ERG nodes reproduces the original shape at a minimal, but topologically correct, geometric level.

Key words: Shape analysis – Reeb graph – Remeshing – Computational topology

1 Introduction

Research in computational topology has recently attracted the attention of a broad scientific community working in diverse fields related to shape modelling, processing and analysis. Computational topology deals with solving topological problems using an algorithmic approach (e.g. computing the homology groups of a given shape) as well as with solving geometrical problems using a topological approach (e.g. mesh simplification with guaranteed topological correctness). Thinking of the many and emerging applications dealing with shape processing, it is obvious that there is a growing interest in the results computational topology (Dey et al. 1999). Knowledge about the global properties of a shape and its main features is very useful for intelligent compression and transmission over the network of dense meshes: the main features and their configuration are important for developing a simplification strategy that discards irrelevant facets without losing the overall structure (Bajaj and Schikore 1998). Computational topology tools are also extremely helpful for content-based search in object databases: tools for building abstract models which naturally combine geometry and global shape properties are very useful for defining high-level search keys and for similarity assessment (Veltkamp 2001; Hilaga et al. 2001).

1.1 Motivation and previous work

In the context of computational topology, we are mainly interested in methods to define topology-driven skeletons of triangle meshes, which bridge the topology and the geometry and allow a minimal rendering of the complete shape while preserving the object topology. While a computational theory of shape is still not formalized, we believe that topology provides the most flexible and well-founded support to devise advanced modelling approaches (Falcidieno and Spagnuolo 1998). There is clear evidence, indeed, that people use different high-level (abstract or global) and low-level (detailed or local) models for shape interpretation. This is especially relevant for the perception of complex forms, in which the ability to vary the level of descriptive abstraction is the key to recognizing and classifying highly complex shapes (Pentland 1986). Methods for extracting topological or morphological structures from low-level object models have been presented in many application areas dealing with complex models whose geometry is represented by triangular meshes.

Related work in this area has been presented by several authors, and the most classical approach to *skeletonization* is surely represented by the medial axis, firstly introduced by Blum and Nagel (1978) and extensively used in image processing. The medial axis of a two-dimensional shape is a graph-like structure whose nodes identify the points where the shape branches into protrusion-like parts and, most usefully, provides the possibility of recovering the original shape using a simple distance transform. Therefore, the medial axis integrates a correct representation of the topology with a very efficient compression of the shape. Unfortunately, in three dimensions, the structure of the medial axis contains not only linear elements but also medial surfaces, is more expensive to compute, and is also sensitive to noise or small undulations on the shape boundary.

For representing three-dimensional polyhedral objects, Lazarus and Verroust (2000) propose a shape skeleton, called a level-set diagram, given by the level sets of a function defined, for every mesh vertex, as the geodesic distance from a selected source point. The method can be applied to unorganised collections of scattered data points lying on a tubular-shaped surface. Using the geodesic distance guarantees that the resulting skeleton is invariant under rotation, translation and uniform scaling, which is a very important property. The skeleton, however, depends on the choice of the source point and on the number of level sets used to slice the surface. In their paper, a heuristics is used to select the source point that seems to work well on elongated tubular shapes such as blood vessels and bones. For this kind of shape, skeletal lines obtained with different source points are very similar. However, the choice of a single source point determines a privileged direction, which may lead to a loss of some features if the object is not strictly cylindrical.

Morse theory, and Reeb graphs in particular, have also been used to define effective topological skeletons, and they are based on the study of the critical points of a continuous function defined over the mesh. First introduced in computer graphics by Shinagawa et al. (1991), they have been used by others for different applications (Wood et al. 2000; Hilaga et al. 2001; Biasotti et al. 2001; Biasotti et al. 2000a; Shattuck and Leahy 2001). The use of Reeb graphs have initially been limited to Morse mapping functions and their construction required a priori knowledge of the object genus (Shinagawa and Kunii 1991). Degenerate critical points were generally not allowed or eliminated by local perturbations of the surface. Local perturbations, however, introduce artefacts not corresponding to any shape features, therefore leading to imprecise shape description. In Hilaga et al. (2001) the authors propose the use of a mapping function based on the geodesic distance and the resulting Reeb graph is therefore invariant under rotation, translation and uniform scaling. The graph, constructed using a multiresolution approach, is proposed as a tool for estimating the difference among shapes. In order to decrease the computational cost of the algorithm, an approximate evaluation of the geodesic distance among the mesh points is suggested. Unfortunately, such a function approximation does not guarantee the absolute independence of its values from the object orientation. Moreover, the relationship between shape and graph representation is not intuitive and does not seem useful for visualization purposes.

1.2 Approach and results

Given a shape, its Reeb graph under the height map can be efficiently extracted by cutting the mesh into parallel slices, orthogonal to the considered height direction, and by studying the properties of the resulting mesh strips delimited by two adjacent levels. Informally, areas of the sliced mesh localize surface critical points when the mesh components have only one or more than two boundary components. A similar approach in the context of CAD surfaces has recently been presented in Jun et al. (2001). The topological connectivity of critical points can be reconstructed using edge-based adjacency among contours. We first developed this method for the extraction of Reeb graphs from triangular meshes representing bivariate surfaces, also called 2.5D surfaces or height fields (Biasotti et al. 2001; Biasotti et al 2000a). Here, we present a generalization of our approach to a wider class of surfaces, in particular closed two-manifold surfaces whose critical points might also be degenerate, as is likely to happen when dealing with real objects. In particular, we describe a revised version of the algorithm presented in Attene et al. (2001) for the automatic extraction of the ERG from closed and two-manifold triangle meshes. The main difference in the presented approach is that vertices not belonging to the contours are kept during the remeshing step. The ERG construction changes accordingly, mainly for the critical-area characterization, and by using Euler's formula within each computed slice, we are guaranteed that holes are always recognized by the process.

The slicing strategy, i.e. the choice of the height function, might be questionable, as the resulting graph is obviously dependent on the choice of the slicing direction. Therefore, the ERG is invariant only under translation and uniform scaling, but not under rotation. The main motivation of our choice concerns the intuitiveness of the resulting shape description, which is based on the *natural* features used to describe a shape, i.e. peaks, pits and passes. Beside intuitiveness, since the nodes of the Reeb graph code the boundary components of critical areas, a minimal but topologically correct representation of the shape can be obtained by a simple tiling of those components (Biasotti et al. 2000b).

The remainder of the paper is organized as follows. First, the theoretical background of this work is described in Sect. 2. Then, our view of critical points and areas with their characterization for triangle meshes is presented in Sect. 3. The integrated mechanism of contour tracing and insertion is described in Sect. 4, which produces the remeshed surfaces. Given the retiled surface, the Reeb graph is easily constructed and the algorithm used is presented in Sect. 5. Results and discussions conclude the paper in Sect. 6.

2 Theoretical background

While topology allows us to classify a shape according to abstract properties of the space associated with the shape (e.g. being knotted or having holes), differential topology deals with the relations among topology and critical points of a function defined on the shape (Guillemin and Pollack 1974; Engelking and Sielucki 1992). In particular, Morse theory states that the topology of a given manifold, e.g. global properties and connectivity, can be described by analysing the critical points of a smooth function defined on the manifold itself (Milnor 1963; Guillemin and Pollack 1974). Let us consider a real-valued function h defined on a smooth manifold M, embedded in the usual three-dimensional Euclidean space. The *critical points* of h are the points of M at which the gradient is zero, i.e. $\nabla(h(P)) = 0$. The pre-image of any value assumed by h, the set $h^{-1}(a)$, defines an isocontour or level set (or simply contour when h is the height function). The height function h is the real function which associates to each point on the surface its elevation; that is, $h(P) = h((x_P, y_P, z_P)) = z_P$, for every $P \in M$. In this case, the contours correspond to the intersections of the surface with planes orthogonal to the height direction. The height function may also be considered according to other than the standard Z-axis direction.

A function h is called Morse if all of its critical points are non-degenerate – that is, if the Hessian matrix H of the second derivatives of h is non-singular at those points. In particular, non-degenerate critical points are isolated, therefore, surfaces with plateaux or volcano rims do not comply with the definition of the Morse function (Milnor 1963). Related to the Morse theory, Reeb introduced a structure, called a *Reeb graph*, which codes the evolution of the level sets of the mapping function (Reeb 1946). More precisely, the Reeb graph of a manifold M with respect to a real-valued function h is defined as follows:

Definition: Let $h : M \to \Re$ be a real-valued function on a compact manifold M. The *Reeb graph* of M with respect to h is the quotient space of $M \times \Re$ defined by the equivalence relation "$\sim$", given by:

$$(X_1, h(X_1)) \sim (X_2, h(X_2)) \Leftrightarrow h(X_1) = h(X_2)$$

and X_1 and X_2 are in the same connected component of $h^{-1}(h(X_1))$.

Intuitively, all points of a compact manifold having the same value under the real function h and whose pre-image belongs to the same connected component are collapsed into one element. Since the contour topology changes only in correspondence of critical levels of the function, the Reeb quotient space can be described as a graph (Milnor 1963; Fomenko and Kunii 1997) (see Fig. 1).

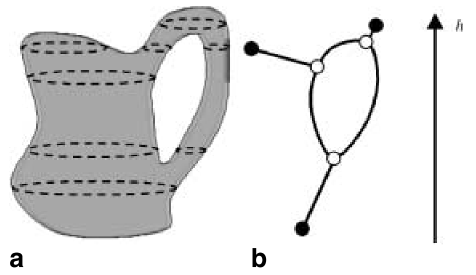


Fig. 1. a Dotted contours represent some of the Reeb equivalence classes with respect to the height function; **b** the graph representation of the quotient space

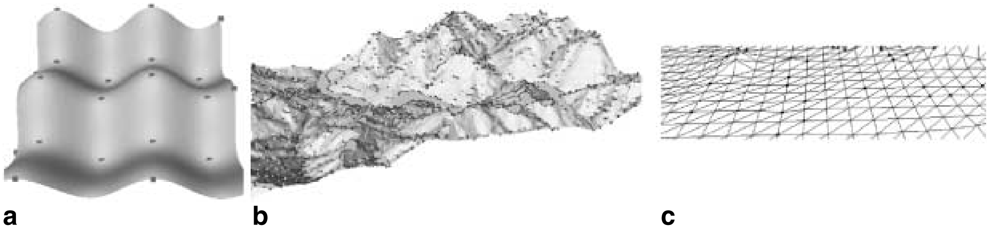


Fig. 2a–c. Characterization based on the method presented in Edelsbrunner et al. (2001) for a "quasi-smooth" triangulation **(a)** and for two real surfaces **(b,d)**. Detail of the critical point distribution is also shown **(c)**

In the Reeb graph, the nodes correspond to critical points of the function h, and the arcs represent the connection between them. Morse theory guarantees also that the topology does not change along the arcs. In applications related to polyhedral surfaces, traditional approaches to the detection of critical points are mainly based on comparing the function value at each vertex with the function value at each of its direct neighbours (Banchoff 1970). This approach is local and therefore very sensible for small changes in vertex location. For example, the Morse complex decomposition proposed in Edelsbrunner et al. (2001) should correctly analyse and simplify piecewise linear two-manifolds. Specifically, the authors extend to piecewise linear manifolds some smooth notions of the Morse theory. For instance, Morse complexes are defined using a *simulating differentiability method*. Critical points are defined as an extension of the Banchoff's criterion, which considers also degenerate saddle points, such as monkey saddles. Starting from critical points, quasi-Morse complexes are constructed by following approximate integral paths on the surface, which form a kind of network on the surface, and by introducing a measure of the importance among critical points, the structure obtained is proposed for mesh understanding and simplification with results shown for digital terrain models. This work may be viewed as an extension to discrete surfaces of the analysis described in Nackman (1984) for smooth surfaces, which provided a surface decomposition into *slope districts*, and further extended to describe *curvature districts* in Nackman and Pizer (1985). Morse complexes are also related to the topological networks called surface networks, presented in Pfaltz (1990). Takahashi et al. (1995) showed how to transform such a network into a Reeb graph.

This approach behaves well only in case of "quasi-smooth" surfaces (see Fig. 2a) but the effectiveness of the description obtained decreases in the case of real data, such as with terrain surfaces. In other words, it is hard to distinguish between global and local shape features without further analysis (see Fig. 2b,c).

By considering the surface behaviour in larger portions of surface, those bounded by the slicing planes, problems due to locality can be avoided. In other words, the slicing frequency filters away critical points, i.e. features, which correspond to small-scale details of the shape.

3 Triangle mesh characterization

Given a triangular mesh T representing a single-sided two-manifold surface without boundary, we want to compute a topological graph G with the same properties as the Reeb graph of T mapped with respect to the height function h. Similar to the smooth case, the nodes of G should correspond to the critical points of $h(T)$. Moreover, we are willing to define the Reeb graph so that h is just a continuous function without restriction to the Morse height function. This graph is the extended Reeb graph (ERG), which was fully defined for graph surfaces in Biasotti et al. (2000a). With respect to previously reported use of the Reeb graph, the extension represented by our ERG concerns the definition of critical points: we propose, in fact, to fully consider degenerate configurations which usually are forbidden or handled with

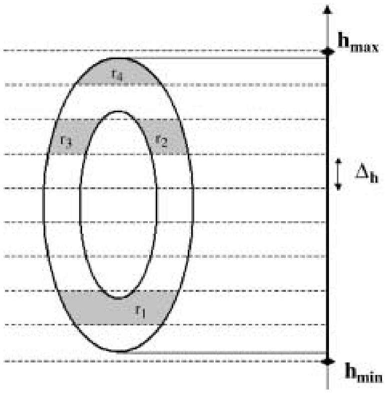


Fig. 3. Critical and regular areas of the constrained triangulation: r_1 and r_4 are critical areas, locating respectively a saddle and a maximum, while r_2 and r_3 are regular ones

local perturbations. Basically, the idea is to locate critical *areas* within which a critical point occurs and identify, starting from them, the smallest area on the mesh whose behaviour is topologically equivalent to the critical point (the *influence zone* of the critical point). Influence zones are the key to identifying the portion of the mesh contributing to the definition of Reeb graph arcs.

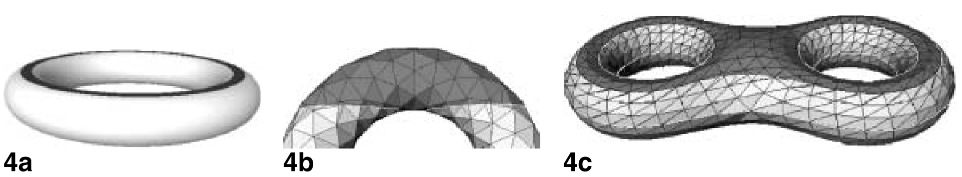
With reference to Fig. 3, the method works as follows. Let us consider a closed interval $[h_{\min}, h_{\max}] \subset R$ of the real line containing the image of T under the mapping h. After fixing the required number of slicing planes, N, we insert the contour lines $h^{-1}(h_{\min} + n * \Delta_h)$ in T, for $n = 0 \ldots N$ and $\Delta_h = (h_{\max} - h_{\min})/N$. Let us now call T^* the mesh defined as the triangulation T constrained to the computed contours; that is, the edges of T^* never cross a contour line. The insertion of the contours decomposes T^* into a set of connected regions, each located between two adjacent levels, and whose boundary edges belong to a contour. The number of connected components in the boundary of these regions and their elevation difference allows us to distinguish between *regular* regions and *critical* ones, the latter containing critical points of the function h. More precisely, if a region has only one or more than two boundary components, then it is classified as critical; otherwise, if the two boundary components have different elevations, the region is regular. In Fig. 3, examples of critical and regular regions are shown.

Critical regions with only one boundary component contain either a minimum or a maximum critical point, which can be easily decided by checking the value of h immediately outside the region – that is, by checking the ascending and descending directions on the surface across their boundary.

A region with two boundary components at the same elevation identifies either a *degenerate* critical area of the height function, or handle-like maximum or minimum critical point (see Fig. 4a,b). The distinction between the two cases is done by checking the inclusion relationship between the components. Critical regions with more than two boundary components may locate saddle points (see region r_1 in Fig. 3), or maximum or minimum areas. Again, the classification is easily done by checking the behaviour of the surface across the region boundary: if all the connected components have ascending directions then the area correspond to a minimum; if all the connected components have descending directions then the area correspond to a maximum; otherwise, it is a saddle. For maxima and minima, it is again possible to distinguish between degenerate and handle-like situations by considering the inclusion relationship of the contours in the corresponding slicing plane. In particular, only two configurations can happen: if one boundary component contains all the others, then the region represents a degenerate area (see Fig. 4c); otherwise, it is a handle-like critical area.

According to classical results of differential geometry, the influence zones of saddle points may be also regarded as the areas on T where the topological change of the surface occurs, when the surface is analysed at the chosen frequency. Note that all critical areas may contain degenerate critical points, i.e. non-isolated ones, and that influence zones may contain more than one isolated critical point at the same elevation; that is, they may locate non-simple critical points.

The adopted characterization criterion is obviously dependent on the frequency of the slicing process: if the frequency is too low, we might lose some important features, such as small holes completely contained within two adjacent slices. It is easy, however, to detect these situations and adapt the frequency to the feature size simply by using the Euler formula for each region. The number of through holes h in a region R is given by $h = (E - V - T + 2 - b)/2$ where E, V, T are the numbers of edges, vertices and triangles in R respectively, and b is



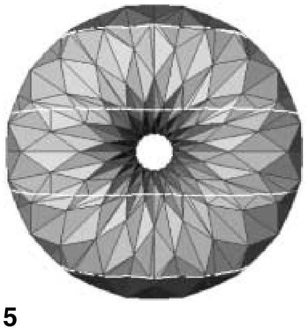


Fig. 4. a Degenerate maximum critical area. **b** Handle-like maximum with two boundary components. **c** Degenerate maximum delimited by three contours

Fig. 5. The hole is missed if too few sections are considered in the slicing process

the number of boundary components of R (Mäntylä 1988).

Handle-like critical areas also occur due to a large slicing frequency, which does not allow detecting the saddle point contained within the slice. In this case, however, there is no loss of information and the topology of the mesh is still correctly represented in the ERG. Degenerate and handle-like critical areas, indeed, are associated with macro-nodes in which the correct link to the adjacent cells is stored (see Sect. 5). In the last section, some results will be provided showing the behaviour of the characterization for different frequencies.

4 Remeshing strategy for constraint insertion

The use of the triangulation constrained to contours makes it easy to compute efficiently the ERG of the mesh. In order to extend the approach described in Biasotti et al. (2000a, 2001) to generic two-manifold triangulations without boundary, it is necessary to ensure that contours are properly inserted as constraints in the mesh. With regard to the contour extraction, plenty of algorithms can be found in the scientific literature, mainly designed for GIS applications and generally developed for 2.5D meshes, i.e. scalar fields (Bajaj et al. 1996; Livnat et al. 1996; Van Kreveld 1994).

In our context, since the contours have to be inserted as constraints, a method has been implemented which computes contours and inserts them in the mesh in a single step. To start the process, only the number N of parallel planes to be intersected with the mesh T is required. For simplicity, the contours are always considered parallel to the XY-plane of the coordinate system in which T is represented. A rotation of the whole mesh is performed for computing the slicing in any user-defined direction. At each level, the intersection of the mesh with the corresponding plane is represented by a set of closed connected components. Intersections at critical levels might produce degenerate contours, such as single points or non-simple contours. To avoid these critical intersections, in particular with superior and inferior extremes of T, the distance among planes is slightly adjusted locally. This can be done because the ERG extraction algorithm does not need the contour levels to be equidistant and is consistent with our definition of the influence zone of critical points. Therefore, inserting all the resulting connected components for each level, as explained in the following, fully solves the problem.

Since the triangulation represents a closed surface, it is possible to track and insert each contour level in a quite simple way, with basic operations involving only the adjacency relations stored in the data structure. More precisely, for each plane Π_i corresponding to a given elevation z_i, we define a list L containing the edges of T that have a non-empty intersection with Π_i. Note that an edge with one of its vertices on the plane is considered to have a non-empty intersection with it. The first element of L is chosen as the first seed for computing a contour component, and during the loop, processed edges are marked. Once the tracking of this first component is complete, the list L is scanned until an unmarked edge is found. If such an edge exists, then it is used as the starting edge to insert a new connected component. The described process is repeated until the list L contains only marked edges. Each connected component is inserted starting from the seed and tracking the contour vertex by vertex. Vertices may be either vertices of T already lying on the contour, or they have to be created every time there is an intersection between an edge and the plane. In this latter case, the mesh is locally updated with the insertion of the appropriate number of new triangles and edges, and the local adjacency relations are updated as well. From this process, a sequence $v_0, v_1, \ldots, v_k$ of vertices is obtained such that an edge of T exists for each pair (v_i, v_{i+1}), as well as for (v_k, v_0). Specifically, at each step of the contour construction an active vertex v is defined; the next vertex is searched by analysing the edges in $VT(v)$, the set of the triangles adjacent to v. Only one of these two situations may occur:

1. There exists one edge e in $VT(v)$ that entirely lies on the plane. The next vertex of the contour will be the other vertex of e, which becomes the current vertex; all the edges incident in v are marked as *visited*.
2. Otherwise, the next edge to be processed is the edge e in $VT(v)$ that intersects the plane, is not adjacent to v and is not marked as *visited*. In this case e is split at the intersection point p, where the next vertex v is created, and the local geometry and topology is updated. The new vertex and all its incident edges are marked as *visited*.

The process starts at the seed edge, which satisfies either one of the two situations described above, or it has only one vertex on the plane. In this latter case, the intersection vertex will be defining the first active vertex.

The process is recursively repeated for each new extracted edge until it is impossible to get a next edge. The algorithm is described in Fig. 6, with a simple tetrahedron as the starting mesh (a). In the first step, the new vertex is inserted at p_0, and the next edge selected is e_1 (b). e_2 could also have been selected as the next edge, depending on the order with which triangles are stored in the $VT(v)$ relation of the new vertex. In this case, the contour would have been traced in the opposite direction. Then, the next new vertex is inserted at p_1, and the next edge processed is e_2 (c). Finally, the last new vertex is inserted, and the process terminates because all the edges of the influence polygon are marked as visited (d).

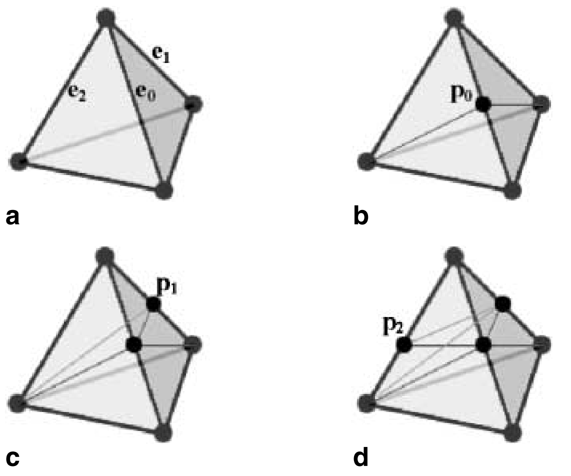
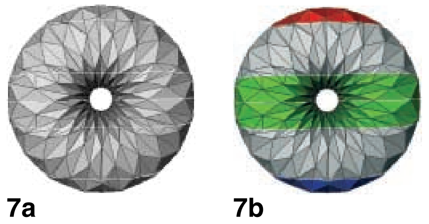


Fig. 6. Inserting a connected component onto a tetrahedron

5 ERG construction

The ERG derives from the direct application of the Reeb graph definition to T^* and allows us to consider a broader class of surfaces than the ones used in other applications of Reeb graphs. Based on these concepts, the nodes of the ERG are defined by *simple nodes*, which correspond to influence zones of simple critical points, and by *macro-nodes*, which are used to represent complex ones, i.e. degenerate and handle-like areas. Starting with the identification of the influence zones of critical points, the *arcs* of the ERG are detected with a simple criterion of topological expansion which connects them on the mesh T^*. The degree of each node is equal to the number of boundary components of the associated influence



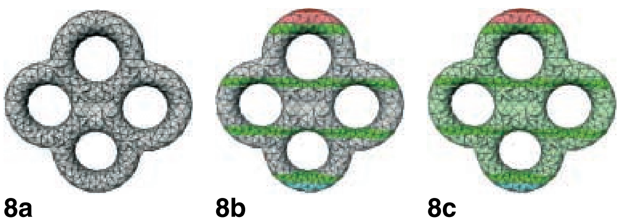


Fig. 7a,b. Adaptive contouring of the mesh in Fig. 5: **a** the hole is detected by halving the contouring step; **b** the consequent characterization

Fig. 8. a The original surface. **b** The first step of the ERG construction. **c** The final one

zone. Notice that simple nodes of type maximum and minimum will always have degree one.

First of all, the regions on T^* delimited by two adjacent levels are detected with a simple region-growing algorithm which starts from a triangle t and expands the region without crossing any constrained edge. At each step of the growing process, the current boundary is updated and possibly divided into more than one connected component if the region being constructed is multiply connected. The growing process stops when all edges of the current boundary lie on contours; that is, they are all constrained edges. At the same time, counters for computing the Euler formula are also updated so that at the end of the growing process both the number of boundary components and genus of the region are available: if the region contains a hole, then the slicing insufficiently samples the mesh and a more dense distribution is chosen (see Fig. 7); otherwise, the region classification is done according to the criteria described in Sect. 3.

With the extraction and classification of influence zones of critical points, the set of nodes of the ERG is completely defined. Each node codes the label of the critical area and its boundary components. At the first stage, we try to connect the nodes of degree one, if any, to their adjacent cells (see Fig. 8b). This is done by expanding the influence zones of simple maximum or minimum areas until the boundary of another influence zone is reached. At the second stage, the missing links between saddles and complex areas have to be determined. Again, the arcs are determined by expanding the influence zones but this time the expansion follows only the free ascending directions of its boundary – that is, directions that do not correspond to arcs already identified. When a free ascending direction is connected to another region, then one arc is defined between the starting node and the reached one (see Fig. 8c). The ERG construction ends when all influence zones and all their free ascending directions have been checked.

Some comments can be made. First of all, since we use the height function to map the surface shape, the extracted ERG depends on the orientation of the height direction. Anyway, the global topology of the shape is captured by the ERG, and as presented in Biasotti et al. (2001), the coding of critical areas in the ERG still verifies the Morse relation $\#\max - \#\text{saddle} + \#\min = \chi$. This can be proven considering that within each slice we are guaranteed that the Euler characteristic of the slice is 2, since there are no holes. Therefore, the contribution of each critical area is given by $2-b$, where b is the number of its boundary components.

The dependence of the ERG on the orientation makes it obviously unsuitable for shape classification or recognition, activities which require unique models for shape description. Nevertheless, it is important to underline that the ERG provides a topological framework for the construction of morphological skeletons, and it has been shown in Biasotti et al. (2000b) that the ERG can be effectively used to render the topology of a shape at a minimal level of detail.

Another important point concerns the density of the sweeping planes, which determines the scale at which shape features are detected. The guarantee to detect all critical points is given by choosing to slice the mesh at each value assumed by $h(v)$, for every

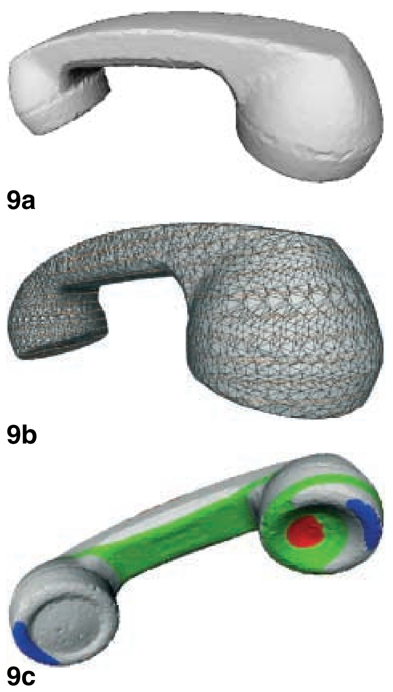


Fig. 9. The ERG extraction process applied to the phone handset

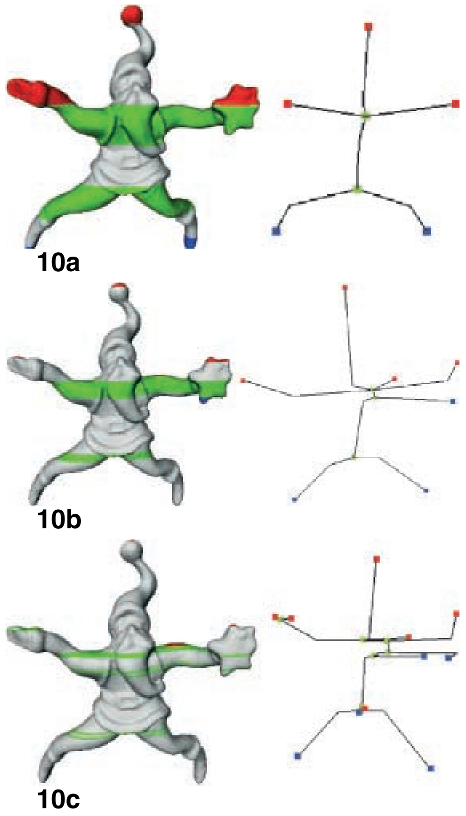


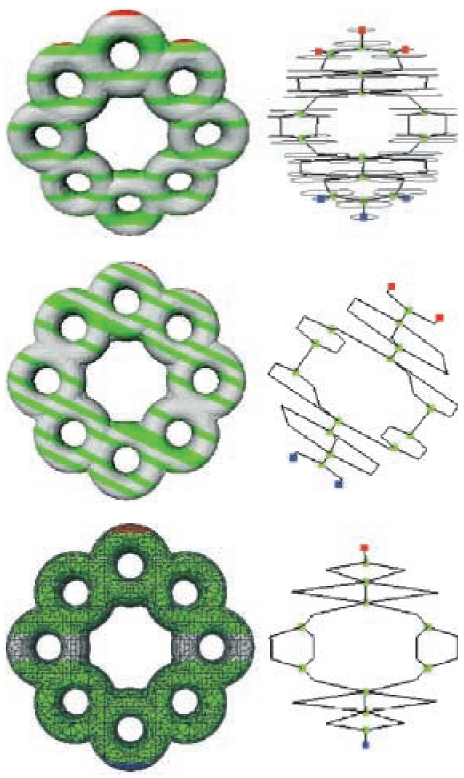
Fig. 10. Examples of extraction of the ERG from the Santa Claus model with various section densities

vertex v of T. Following this criterion, it is necessary to compute a number of sections larger than the effective features of the surface, and the characterization would converge to the one proposed by Banchoff (1970). A similar approach has been proposed in (Carr et al. 2000) for computing their contour trees. To prune away irrelevant features, Edelsbrunner et al. (2001) propose a first characterization of the triangulation and then the computation of the minimum distance among the features obtained. Our approach guarantees that holes are preserved, with their corresponding critical points, while we allow losing some maximum or minimum points whose amplitude with respect to the slicing direction is smaller that the chosen frequency. Actually, some of these small maximum or minimum points might be still recognized, if they are not completely contained within a slice. Moreover, the idea proposed in Kulkarni and Dutta (1996) of considering a non-uniform slicing of the triangulation also surely improves our algorithm.

6 Results and discussion

Based on the described techniques, a prototype system has been implemented which performs the contour computation and insertion, and the Reeb graph extraction. The original mesh can be swept along any user-defined direction and with arbitrary number of sections. The whole process is depicted in Fig. 9, where the original mesh is shown in (a) and the mesh after the contouring step is shown in (b). The critical areas are depicted in (c) with a colouring scheme which associates the blue colour to minimum areas, the red to maxima, and the green to saddles. Finally, the resulting ERG is shown in (d) with the boundary contours of the critical areas.

Figure 10 shows the ERG extraction on some examples with various sampling steps. Notice that as the section density increases smaller topological features are detected.



11

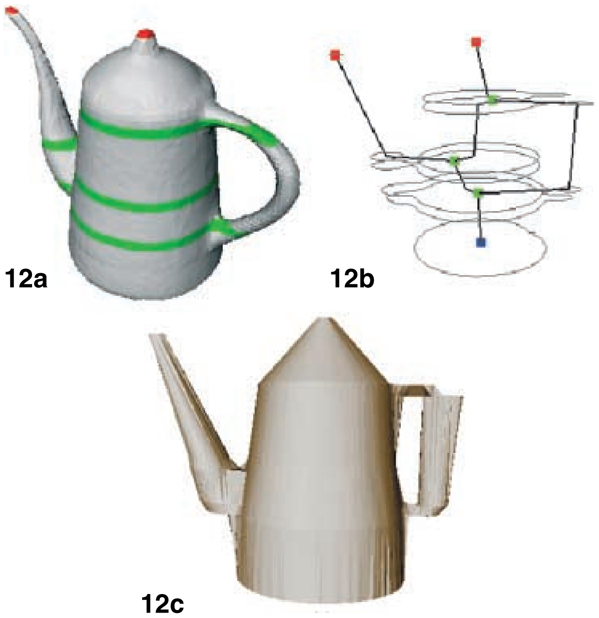


Fig. 11. Extraction of the ERG from a model with 9 through holes with various section densities and directions

Fig. 12. The ERG for shape compression: a the mesh characterization; b its ERG graph, and the mesh reconstructed only from the critical sections and their relations given by the ERG

Figure 11 shows how changing the slicing direction and density does not change the number of cycles in the computed ERG.

The global complexity of the remeshing algorithm can be given as a function of the maximum value between the number of vertices of the original triangulation, n, and the number of the constrained ones, m. Moreover, it can be seen that the number of edges and triangles are of the same order as the number of vertices. In the slicing step, the edge-ordering preprocessing requires $O(\max(m, n\log n))$ operations. Then, $O(n\log n)$ operations are needed to sort the edges, and $O(\max(m, n))$ is the number of intersection tests. Inserting the whole set of constraints requires $O(m)$ edge splits.

With regard to the computational complexity of the ERG extraction, the recognition of critical areas and the detection of influence zones require $O(t)$ operations, where t is the number of triangles. The complexity of the arc completion step is expressed by $O(t)$, so that the total computational cost of the ERG extraction is $O(m\log m)$. Therefore, the whole process, starting from a generic triangulation, requires $O(\max(m\log m, n\log n))$ operations.

As shown in Fig. 12, the ERG structure is not only an abstract topological description; the visualization of the ERG contours and their adjacency, in fact, provides a simple and effective sketch of the original shape. With reference to Fig. 12, the shape of the original surface can be restored using contour-blending techniques, especially if the ERG is augmented with more sections along the arcs that identify significant changes of the contour shape. In this sense, we are currently working on the use of the ERG as the reference structure to compress and decompress shape models (Biasotti et al. 2000b). Notice that, while the Reeb graph is dependent on the orientation of the height function, the restoring process produces the original shape with the correct topology independently of the orientation.

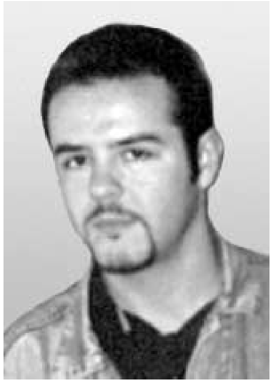
Future developments of this method mainly concern the definition of a morphological structure to be merged with the ERG, which codes also the main morphological changes among contours. From a theoretical point of view, we are also investigating a possible limit to the number of configurations of the Reeb graph of a given shape.

Acknowledgements. The authors would like to thank all the people of the Computer Graphics Group at IMATI-CNR, and especially the reviewers for their helpful comments.

References

1. Attene M, Biasotti S, Spagnuolo M (2001) Re-meshing techniques for topological analysis. In: Proceedings of Shape Modelling International 2001, Genova, Italy. IEEE Press, pp 142–152
2. Axen U, Edelsbrunner H (1998) Auditory Morse analysis of triangulated manifolds. In: Hege HC, Polthier K (eds) Mathematical Visualization. Springer, Berlin Heidelberg New York, pp 223–236
3. Bajaj C, Pascucci V, Schikore DR (1996) Fast isocontouring for improved interactivity. In: Proceedings of 1996 SPIE Symposium on Volume Visualization, San Francisco. ACM Press, pp 39–46
4. Bajaj C, Schikore DR (1998) Topology preserving data simplification with error bounds. Comput Graph 22(1):3–12
5. Banchoff T (1970) Critical points and curvature for embedded polyedral surfaces. Am Math Mon 77:475–485
6. Barequet G, Dickerson M, Eppstein D (1998) On triangulating three-dimensional polygons. Comput Geom 10:155–170
7. Biasotti S, Falcidieno B, Spagnuolo M (2001) Shape abstraction using computational topology techniques. In: Cugini U, Wozny M (eds) Geometric Modelling to Shape Modeling. Kluwer Academic Publishers, Boston
8. Biasotti S, Falcidieno B, Spagnuolo M (2000a) Extended Reeb graphs for surface understanding and description. In: Proceedings of 9th Discrete Geometry for Computer Imagery conference, LNCS 1953. Springer, Berlin Heidelberg New York, pp 185–197
9. Biasotti S, Mortara M, Spagnuolo M (2000b) Surface compression and reconstruction using Reeb graphs and shape analysis. In: Proceedings of the 16th Spring Conference on Computer Graphics, Budmerice, Slovakia. Comenius University, Bratislava, pp 174–185
10. Blum H, Nagel R (1978) Shape description using weighted symmetric axis features. Pattern Recogn 10:167–180
11. Carr H, Snoeyink J, Axen U (2000) Computing contour trees in all dimensions. In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA 2000), San Francisco. ACM Press, pp 918–926
12. Dey TK, Edelsbrunner H, Guha S (1999) Computational topology. In: Chazelle B, Goodman J E, Pollack R (eds) Advances in discrete and computational geometry. Contemporary mathematics vol 223. AMS, Providence, pp 109–143
13. Edelsbrunner H, Harer J, Zomorodian A (2001) Hierarchical Morse complexes for piecewise linear 2-manifolds. In: Proceedings of the ACM Symposium on Computational Geometry, Medford, Mass. ACM Press, New York, pp 70–79
14. Engelking R, Sielucki K (1992) Topology: a geometric approach. Sigma series in pure mathematics, vol 4. Heldermann, Berlin
15. Falcidieno B, Spagnuolo M (1998) Shape abstraction paradigm for modeling geometry and semantics. In: Proceedings of Computer Graphics International, Hannover. IEEE Computer Society, pp 646–656
16. Fomenko A, Kunii TL (1997) Topological modelling for visualization. Springer, Berlin Heidelberg New York
17. Guillemin V, Pollack A (1974) Differential topology. Prentice-Hall, Englewood Cliffs, NJ
18. Hilaga M, Shinagawa Y, Kohmura T, Kunii TL (2001) Topology matching for fully automatic similarity estimation of 3D shapes. In: ACM SIGGRAPH 2001. ACM Press, pp 203–212
19. Jun CS, Kim DS, Kim DS, Lee HC, Hwang J, Chang TC (2001) Surface slicing algorithm based on topology transition. Comput Aided Des 3:825–838
20. Kulkarni P, Dutta D (1996) An accurate slicing procedure for layered manufacturing. Comput Aided Des 28:683–697
21. Lazarus F, Verroust A (1999) Level set diagrams of polyhedral objects. In: Proceedings of the Fifth ACM/SIGGRAPH Symposium on Solid Modeling and Applications. ACM Press, pp 130–140
22. Livnat Y, Shen HW, Johnson CR (1996) A near optimal isosurface extraction algorithm using the span space. IEEE Trans Vis Comput Graph 2:73–84
23. Mäntylä, M (1988) An introduction to solid modelling. Computer Science Press, Rockville, Md.
24. Milnor J (1963) Morse theory. Princeton University Press, Princeton, N.J.
25. Nackman LR (1984) Two-dimensional critical point configuration graphs. IEEE Trans Pattern Anal Mach Intell 6(4):442–450
26. Nackman LR, Pizer SM (1985) Three-dimensional shape description using the symmetric axis transform I: theory. IEEE Trans Pattern Anal Mach Intell 7:187–202
27. Pentland AP (1986) Perceptual organization and representation of natural form. Artif Intell 28:293–331
28. Pfaltz J L (1990) Surface networks. Geograph Anal 8(1):77–93
29. Reeb G (1946) Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. C R Acad Sci Paris 222:847–849
30. Shattuck DW, Leahy RM (2001) Automated graph based analysis and correction of cortical volume topology. IEEE Trans Medical Imag 20(11):1167–1177
31. Shinagawa Y, Kunii TL, Kergosien YL (1991) Surface coding based on Morse theory. IEEE Comput Graph Appl 11(5):66–78
32. Shinagawa Y, Kunii TL (1991) Constructing a Reeb graph automatically from cross sections. IEEE Comput Graph Appl 11(6):44–51
33. Takahashi S, Ikeda T, Shinagawa Y, Kunii TL, Ueda M (1995) Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data. Comput Graph Forum 14(3):181–192

34. Van Kreveld M (1994) On quality paths on polyhedral terrains. In: Nievergelt J, Roos T, Schack HJ, Widmayer P (eds) Proceedings of IGIS'94: Geographic Information Systems. Lecture Notes in Computer Science. Springer, Berlin Heidelberg New York 884:113–122
35. Veltkamp R (2001) Shape matching: similarity measures and algorithms. In: Proceedings of Shape Modelling International 2001, Genova, Italy. IEEE Press, pp 188–197
36. Wood ZJ, Desbrun M, Schröder P, Breen D (2000) Semi-regular mesh extraction from volumes. In: IEEE Visualization Conference, Salt Lake City. IEE Computer Science Press, pp 275–282



MARCO ATTENE received a Computer Science Laurea degree from the University of Genoa in July 1998. Currently he is a researcher at the Institute for Applied Mathematics of the Italian National Research Council (CNR), where he is member of the Computer Graphics Group. He is studying for a Ph.D in Electronic and Computer Engineering and his main research interests are related to discrete geometric modeling, triangle mesh analysis and re-meshing techniques.

SILVIA BIASOTTI graduated in Mathematics at the University of Genoa in 1998. Since then, she has been on the staff of the Computer Graphics Group at the Institute for Applied Mathematics of the National Research Council in Genoa. Moreover, since 2000, she is a PhD student in Mathematics and Applications at the University of Genoa (Italy). Currently, she is a contract researcher at the Institute of Applied Mathematics and Information Technology of the National Research Council in Genoa. Her research interests include computational topology, shape abstraction and skeleton representation of polyhedral surfaces.

MICHELA SPAGNUOLO is senior researcher at IMA-CNR, where she is member of the Computer Graphics Group. She received a Laurea in Mathematics from the University of Genova, Italy, and the Doctorate degree in Computer Science Engineering from the Institute National des Sciences Appliquées de Lyon, France. Her research interests are related to shape-based approaches for modelling spatial data and geometric reasoning for the extraction of shape features from discrete surface models. Lately, she has been mainly involved in the development of geometric models for coding uncertainty in data samples (fuzzy-based modelling).