3D Mesh Decomposition using Reeb Graphs

Stefano Berretti, Alberto Del Bimbo and Pietro Pala
Dipartimento di Sistemi e Informatica
University of Firenze
via S. Marta 3, 50139 Firenze, Italy
{berretti, delbimbo, pala}@dsi.unifi.it
January 22, 2009

Abstract

Decomposition of complex 3D objects into simpler sub-parts is a challenging research subject with relevant outcomes for several application contexts. In this paper, an approach is proposed for decomposition of 3D objects based on Reeb-graphs. The approach is motivated by perceptual principles and supports identification of salient object protrusions. Experimental results are presented to demonstrate the effectiveness of the proposed approach with respect to different solutions appeared in the literature, and with reference to ground-truth data obtained by manual decomposition of 3D objects.

Keywords: 3D mesh decomposition; Reeb-graph; geodesic distance

1 Introduction

In recent years, a growing number of solutions have been proposed to generate 3D digital models of real objects. These solutions differ in terms of costs, resolution and type of acquired information (the un-textured external surface of the object, the textured external surface or even the interior composition of the object) and include CAD, tomography, magnetic resonance, 3D laser scanners, structured light systems and photogrammetry.

As a result, 3D models are nowadays used in many application domains, including manufacturing industries (to support design and rapid prototyping), medicine (to support disease study and diagnosis), video production (to boost construction of special effects by reusing parts of available high quality 3D models), and cultural heritage (to enable accurate detection of geometric deformation of art-pieces due to inadequate preservation conditions). Several of these applications can benefit from the possibility to cut a 3D object model into simpler parts, a process that in the literature is referred to as 3D object decomposition. This can be useful in
areas as diverse as modeling [1], metamorphosis [2, 3], compression [4], remeshing and simplification [5, 6], 3D shape retrieval [7, 8], collision detection [9], texture mapping [10] and skeleton extraction [11].

Psychological studies have evidenced that human perception of 3D shapes is partially based on objects decomposition and that any complex object can be regarded as an arrangement of simple primitives or components [12]. Depending on the application context, 3D object decomposition can target different objectives and thus yield the identification of different parts. Accordingly, solutions proposed so far are application specific, differing in terms of optimization criteria and partitioning objectives. For instance, in applications dealing with object compression or simplification, object decomposition may target the identification of parts with some uniform geometric property so as to maximize the effectiveness of the compression/simplification algorithm on each individual part. Differently, in applications dealing with object modeling and retrieval, object decomposition may target the identification of parts with a clear semantics or parts that are perceptually meaningful, so as to enable reuse of object components in the development of new models or support searching by parts in content based retrieval of 3D objects.

Following the previous considerations, approaches for 3D object decomposition can be distinguished into geometry based and semantic oriented [13, 14]. In the former, the goal is to identify parts of the object that are homogeneous with respect to a criterion based on geometric properties (e.g., local curvature or distance to a reference point). In the latter, the goal is to identify parts of the object that are meaningful under some semantic or perceptive criteria (e.g., protrusions or articulated parts). An up to-date review of the state of the art on 3D mesh decomposition can be derived from the recent survey of Shamir [13], and the comparative study of some representative techniques given by Attene et al. [14].

1.1 Previous Work

Geometry based approaches aim at partitioning the object surface into patches of homogeneous geometric properties and are typically used as pre-process for other geometry processing tasks.

Curvature information is exploited by many of the works in this category. Developing on surface curvature, in [15] a watershed approach to 3D mesh decomposition is presented that generalizes the morphological watersheds originally proposed for image segmentation. The curvature of the surface defined at each vertex is used as an indication of region boundaries.
which partition the mesh into patches of relatively homogeneous curvature. One problem of this algorithm is its dependency on the exact triangulation of the model. In addition, even planar components can be undesirably partitioned. Developing on this approach, a fast marching watershed solution is reported in [16]. While results in [15] are negatively affected by the use of pseudo-curvatures, the actual principal curvatures and directions (computed according to [17]) are used in [16] to improve the partitioning results. Principal curvature information for mesh decomposition is also exploited in [18]. In this case, decomposition of 3D objects is achieved by way of a stochastic relaxation process determined by the mutual synchronization of a network of pulse-and-fire oscillators. Curvatures at mesh vertices are used as inputs to drive synchronization of the oscillator network. Although grounding on perceptually motivated principles, decomposition results are biased by typical drawbacks of relaxation processes including dependence on variable initialization and premature truncation of the relaxation period.

Several geometry based methods identify homogeneous patches by using clustering techniques such as region growing [19], iterative clustering [6, 20] or hierarchical clustering [5].

In [19], mesh decomposition is accomplished by extracting values of principal curvatures $k_{\min}$ and $k_{\max}$, and clustering them using standard $k$-means. Once clustering is complete, each mesh vertex is labeled according to the cluster it belongs to. Analysis of connected components and region growing are performed to remove small regions and identify relevant object parts. However, there is no evidence that perceptually salient regions correspond to regions of uniform curvature. A further drawback is that the number of regions ultimately depends on the final number of clusters, which is assumed to be known a priori.

In [6], a variant of the $k$-means algorithm is presented for the creation of planar shape proxies. For any given partition of a surface in $k$ regions, a set of shape proxies, represented by their average point and surface normal, is associated that approximates the whole geometry. In [20], an algorithm is proposed to partition a mesh into developable parts. First each segment is approximated by a developable surface element. During this step an iterative region growing approach, using a variant of the $k$-means algorithm is applied. Segments are then modified so as to guarantee that neighboring surface elements intersect at their boundaries, thus prohibiting stretching. Finally, the algorithm extracts the analytical boundaries between the surface elements, making the boundaries intuitive and easy to cut and glue.

Clustering has been also used according to a hierarchical approach that initializes each
face with its own separate cluster. During clustering, each pair of clusters is assigned a cost for merging them into one cluster and the lowest cost pair is merged. In [5], hierarchical face clustering uses $L_2$ distance and orientation norms from representative planes as a measure of planarity, but formulates them using quadratic error metric for efficient computation. This algorithm also uses a bias term to create circular compact clusters by using the perimeter and the area of the cluster to measure its compactness.

The spectral graph theory, which states the relationship between the combinatorial characteristics of a graph and the algebraic properties of its Laplacian [21], is also used by some geometry based solutions. For example, in [22] the spectral analysis is exploited by constructing the symmetric matrix of squared geodesic distances between surface vertices, and then using the IsoMap dimensionality reduction method. Given a set of high-dimensional points, IsoMap computes the geodesic distances along a manifold as sequences of hops between neighboring points. It then applies Multi Dimensional Scaling (MDS) to these geodesic distances to find a set of points embedded in low-dimensional space with similar pairwise distances. This results in the identification of a set of surface patches on the original 3D model. Boundaries of patches are optimized so as to satisfy two objectives: they should cut through areas of high curvature without being too jaggy; and they should minimize the embedding distortions of the charts they border.

The recent and increasing interest in mesh decomposition approaches in the semantic oriented category is motivated by their capability in supporting parametrization or re-meshing schemes, metamorphosis, 3D shape retrieval, skeleton extraction as well as the modeling by a composition paradigm that is based on natural shape decompositions.

Early studies in this category developed on the minima rule, a theory of human perception of 3D shapes which states that part boundaries occur along lines of negative minima curvature [12]. For example, in [23] an object is viewed as a charged perfect conductor, and the physical property which tends to accumulate charges on the surface of a conductor at points of sharp convexity and vanish at points of sharp concavity is used to guide the decomposition process: boundaries of object parts are detected by locating surface points exhibiting minima of the local charge density. This approach leads to meaningful decompositions, although applicable to a limited range of object types: genus-one topology, no dents on the surface of the object and presence of at least one part boundary. The minima rule is also exploited in [24], where first loci of minimum curvature are identified on the object surface, then the most relevant
ones are used to partition the object by finding candidate contours. However, the approach is complicated by the difficulty in finding stable relevant contours and in closing them in order to determine salient partitioning profiles. Eventually, a part saliency test is used to discard irrelevant parts. Though a completely automatic scissoring is possible, manual intervention is required to select relevant contours in difficult situations.

Developing on the idea that, in addition to curvature, also proximity information is relevant in determining salient object parts, in [25] clustering has been applied to a function accounting for both the proximity and angular distance between surface elements. Though meaningful components of 3D objects are found with this approach, boundaries between regions are often jagged and not accurately identified. More recently, in [11] this idea has been extended through a hierarchical mesh decomposition algorithm based on fuzzy clustering. The basic idea of this approach is first to find the meaningful components of the mesh and only afterwards focus on generating the exact boundaries between components. This is obtained by defining pairwise geodesic distances between mesh faces, and then iteratively refining a set of $k$ representative faces through a fuzzy clustering algorithm. In this process, each face is assigned the probability of belonging to each of the representative sets so that faces are clustered into patches. The decomposition is iteratively applied to each part so as to yield a hierarchy of decomposed regions. A recursion stop condition is considered to identify the level of the hierarchy, but it is difficult to relate it with a perceptually salient object decomposition.

Many of the semantic oriented approaches introduce a bias which is assumed to help the partitioning algorithm to provide significant decompositions. Examples of bias are feature points [26, 27], symmetry transformations [28], shape diameter [29], skeletons [30], and Reeb-graph defined with respect to the integral geodesic function [31].

In [26], the outer contour of the 2D spectral embedding of the mesh (or sub-mesh) is used as feature to guide the partitioning. The main idea lies in utilizing appropriately defined projection operators to retain and enhance structural and geometric information in a 1D or 2D space. In the low-dimensional space, a segmentability analysis, sampling and cut extraction are performed more efficiently and effectively. When the object being segmented is flexible or dynamic, such as a human or animal models, it can maintain various poses. In such cases it is important that the objects partitioning remains consistent despite the pose changes. To construct a pose-invariant decomposition, in [27] multidimensional scaling (MDS) to 3D is used on a coarse approximation of the mesh. MDS finds an embedding of higher dimensional distances.
(geodesic distances from each point to all other points) into a lower dimension Euclidean space, where Euclidean distances approximate well the higher dimensional distances. Often this maps different poses of the same object to similar poses. Later, feature points are extracted, namely points that reside on tips of prominent components of a given model. Each prominent component (or segment) of the object is defined by one or more of these feature points. Using spherical mirroring, the core of the object is extracted and then the other segments. A final refinement stage uses graph cut to finalize the boundaries of the segments.

In [28], the symmetry analysis is used to segment a mesh into components. First the $m$ major symmetry planes of an object are found based on sampling. Next, for each face and for every symmetry plane, a score is computed so as to capture the degree to which the face contributes to the symmetry with respect to that plane. Each face is thus described using the $m$ score values and object decomposition is obtained by clustering faces based on their score values.

The shape diameter function (SDF) is applied in [29] to measure the local diameter of the object at points on its boundary instead of the local radius (distance to the medial axis). The function values on a point laying on the mesh surface are averaged from sampling the length of rays sent from the point inward to the other side of the objects mesh. The SDF gives a good distinction between thick and thin parts of the object and thus has been used for part-type partitioning. Another advantage of the SDF is its pose-oblivious nature: the SDF values of points on the mesh remain largely invariant under pose changes of the object. Hence, it has been used to consistently partition meshes representing dynamically moving objects. Other methods which use the connection from the mesh to the volume of the object include [32], in which geometric primitives are fitted to the mesh to partition the object, and [33] in which the partitioning is based on blowing a spherical bubble at each vertex and studying how the intersection of that bubble with the surface evolves.

Some semantic oriented approaches are based on the notion of object skeleton. This is a lower dimensional object that captures the shape of the original object. Since the skeleton is simpler than the original object, many operations, e.g., shape recognition and deformation, can be performed more efficiently on the skeleton than on the full object. An interesting approach based on skeletonization is proposed in [9]. In this solution, a skeleton is extracted from the original mesh and critical points (i.e., points capturing changes in geometry, topology or both) are identified by sweeping a plane perpendicular to mesh branches. Each critical
skeleton point is used to define cuts using the sweep planes which segment the mesh into different parts. Appealing results are obtained with this algorithm, though the smoothing operation used for skeleton extraction is associated with potential loss of mesh details. In [30], a framework that simultaneously generates shape decompositions and skeletons is proposed. It extracts the skeleton from the components in a decomposition and evaluates the skeleton by comparing it to the components. The process of simultaneous shape decomposition and skeletonization iterates until the quality of the skeleton becomes satisfactory.

Recently, based upon ideas from Morse theory [34] and Reeb-graphs [35], topological analysis of smooth functions defined on the mesh surface has been used to perform mesh decomposition. In so doing, the Reeb-graph is the bias used to drive the decomposition algorithm. The possibility to define global functions that smooth the effect of noise and local perturbations, makes these solutions appealing especially in applications that require the identification of relevant protrusions of 3D objects. In [31], the local extrema of the average geodesic distance (AGD) function [36] define the critical points. Rather than computing geodesic distances from all the surface points, distances are computed from a small number of evenly spaced points on the surface. A Reeb-graph is constructed by performing region growing in increasing order of AGD and tracking the topological changes in the wavefront. However, the embedded Reeb-graph can contain an excessive number of local maxima and saddle points, thus requiring a filtering scheme to weed out extra local maxima and split saddle points. In [37] a hierarchical partitioning approach is proposed that uses enhanced topological skeletons [38]. First, mesh feature points (i.e., mesh vertices located on extremities of prominent components) are extracted by using two geodesic based mapping functions—whose origins are the mesh most distant vertices—and intersecting the sets of their local extrema. Then, for each vertex in the mesh a mapping function is defined as the geodesic distance to the closest feature point. The analysis of the number of connected subsets as the function evolves, enables the construction of a Reeb-graph. Each connected component of the Reeb-graph is modeled with an ordered collection of closed curves and a curvature estimation is computed for each of its curves. Curves that locally minimize the curvature estimation are identified as constrictions. In particular, the alignment of the mapping function level lines with surface bottlenecks is obtained by integrating surface curvature into the geodesic distance computation. The effect is to increase the distance between two vertices when a concave region separate them.
1.2 Overview and Paper Organization

In this paper, we propose an approach for perceptually consistent decomposition of 3D objects based on Reeb-graph. Our solution falls in the *semantic oriented* category and is motivated by the need to overcome limitations of geometry based solutions which mainly rely on the sole curvature information to perform mesh decomposition. In particular, we propose the use of Reeb-graph to extract structural and topological information of a mesh surface and to drive the decomposition process. Curvature information is used to refine boundaries between object parts in accordance to the minima rule. Thus, object decomposition is achieved by a two steps approach accounting for Reeb-graph *construction* and *refinement*. In the construction step, topological as well as metric properties of the object surface are used to build the Reeb-graph. Due to the metric properties of the object that are considered for building the Reeb-graph (i.e., the AGD is used), the structure of this graph captures the object protrusions. In the refinement step, the Reeb-graph is subject to an editing process by which deep concavity and adjacency are used to support fine localization of part boundaries. In doing so, the main goal of our contribution is to provide and experiment a model to support perceptually consistent decomposition of 3D objects to enable reuse and retrieval of parts of 3D models archived in large model repositories. To demonstrate the effectiveness of the proposed solution, a comparison with two different approaches to object decomposition is reported. Furthermore, in order to estimate the extent to which an object decomposition matches perceptually salient object parts, a test is carried out using a ground-truth archive of manually partitioned objects.

This work has some relations with our previous studies on 3D object decomposition using Reeb-graphs. In particular, preliminary ideas and results appeared in [39] and [40]. Other works used Reeb-graphs for the purpose of 3D mesh partitioning. Among these, the more related to our proposal are [31] and [37]. With respect to the work of Zhang et al. [31], our solution differs in some aspects. First, in our approach the AGD undergoes to an equalization process that emphasizes differences in the AGD distribution. In addition, we propose a new and original approach for the adjustment of region boundaries based on curvature information that improves the capability of the approach to produce significant mesh decomposition. Differently from the work of Tierny et al. [37], where geodesic distances are computed with respect to *features points* of a mesh, the AGD is used in our approach as the function that induces the Reeb-graph. Furthermore, in [37] geodesic distances from the feature points are computed by weighting the length of mesh edges with a term which depends on the curvature index [41],
and Gaussian curvature is computed for each curve delimiting a node of the Reeb-graph. In our solution, a different approach is proposed for the refinement of region boundaries which is performed independently from the geodesic distance computation using principal curvatures.

A distinctive aspect of the proposed solution is the non-uniform quantization of the AGD that induces the Reeb-graph (see Sect. 3.1). This aims to adaptively derive topological variations of Reeb-graph—and consequently of the 3D object—without any discrete quantization that implicitly determines the resolution at which the object is analyzed. Doing so, our approach is directly related to the work of Attene et al. [42]. In particular, in [42] the idea to densely quantize values of the function that induces the Reeb-graph so as to maximize the resolution at which a 3D shape is analyzed was proposed. However, Reeb-graphs reported in their work are induced by the height function using a set of slicing planes. The density of the sweeping planes is conditioned by a slicing parameter and ultimately determines the scale at which the shape features are detected. Detection of all critical points is obtained by slicing the mesh at each value assumed by the height function, for every vertex. The solution we propose develops on this idea, but is not constrained to uniform quantization, by using the non-uniform quantization induced by the values of AGD on mesh vertices.

The remainder of the paper is organized as follows: In Sect. 2, the theoretical foundation of Reeb-graph and its relationship to surface topology are discussed. In Sect. 3, the proposed approach to mesh decomposition using Reeb-graph is expounded. In particular, we first describe the process of Reeb-graph construction based on surface topology, then we discuss how to refine the Reeb-graph based on surface curvature analysis. The effectiveness of the proposed approach is experimented in Sect. 4, both presenting decomposition results in comparison to different approaches (Sect. 4.1), and measuring the perceptual saliency of object decompositions (Sect. 4.2). Properties of the approach are also discussed with respect to several criteria that are commonly used in the analysis of approaches to 3D object decomposition (Sect. 4.3). Finally, conclusions and future research directions are drawn in Sect. 5.

2 Surface Topology and Reeb-graphs

Topology is the study of the properties of a shape that do not change under deformation. In this context, the deformation of a shape $A$ into a shape $B$ should be intended as a transformation obeying four basic rules: (i) onto, i.e., every point of $A$ is used and every point of $B$ is used; (ii) one-to-one, i.e., each point of $A$ maps to a single point on $B$; (iii) no overlap, i.e., two
distinct points on $A$ cannot be mapped onto the same point on $B$; (iv) continuous, i.e., the transformation cannot tear $A$, join sections of $A$ together, poke holes into $A$, or seal up holes in $A$. If all these rules hold, $A$ is said to be homeomorphic to $B$, which means that $A$ and $B$ are topologically equivalent.

Morse theory [34] draws a connection between the differential properties of a certain class of smooth functions (the Morse functions) defined on a manifold and the topological properties of that manifold. Given an object surface $S$, and a smooth real valued function $f$ defined on it, Morse theory provides the relationship between the critical points of $f$ and the topology of $S$. In particular, the critical points of $f$ are defined as points on the surface where the partial derivatives of $f$ are zero (i.e., the gradient no longer has a specific direction).

As an example, consider the eight-shaped object shown in Fig. 1 and assume that the function $f$ is the elevation (for a generic point $(x, y, z)$ on the object surface $f = z$). Dark lines shown on the surface of the object highlight some level sets induced by $f$. Level sets correspond to loci of points $(x, y, z)$ on the surface with the same values of $f$. In the example of Fig. 1, the magnitude of the derivative of $f$ at a generic point $(x, y, z)$ on the object surface is the slope of the surface at $(x, y, z)$. The direction of the derivative of $f$ at $(x, y, z)$ is the direction of steepest ascent.

In Fig. 1, arrows highlight six critical points on the object surface where there is no single direction of steepest ascent. The local properties of the surface at these critical points can be described through an index $\chi = \chi(x, y, z)$ representing the number of negative eigenvalues of the Hessian (i.e., the number of independent directions in which $f$ decreases):
\[ H(f)|_{(x,y,z)} = \begin{pmatrix} \frac{\partial^2 f}{\partial \alpha^2} \bigg|_{(x,y,z)} & \frac{\partial^2 f}{\partial \alpha \partial \beta} \bigg|_{(x,y,z)} & \frac{\partial^2 f}{\partial \beta^2} \bigg|_{(x,y,z)} \\ \frac{\partial^2 f}{\partial \beta \partial \alpha} \bigg|_{(x,y,z)} & \frac{\partial^2 f}{\partial \beta^2} \bigg|_{(x,y,z)} & \frac{\partial^2 f}{\partial \alpha \partial \beta} \bigg|_{(x,y,z)} \end{pmatrix} \]

being \( \alpha \) and \( \beta \) variables of a local parametrization of the object surface at \((x, y, z)\).

Value of the index is zero for minima points (point ‘a’ in Fig. 1), one for saddle points (points ‘b’, ‘c’, ‘d’ and ‘e’ in Fig. 1) and two for maxima points (point ‘f’ in Fig. 1).

It can be noticed that critical points identify structural changes of the topology of level sets. For instance, level sets identified by values of \( f \) slightly above the elevation of point ‘d’ in Fig. 1 are homeomorphic (topologically equivalent) to pairs of disjoint circles. Differently, level sets identified by values of \( f \) slightly below the elevation of point ‘d’ are homeomorphic to a single circle.

A Reeb-graph is a schematic way to encode the behavior of a Morse function \( f \) on a surface. Nodes of the graph represent connected components of the level sets of \( f \), contracted to points. Formally, given a surface \( S \), the Reeb-graph is the quotient space of \( f \) in \( S \times \mathbb{R} \) formed by the equivalence relation \((v_1, f(v_1)) \sim (v_2, f(v_2))\), being \( v_1 \) and \( v_2 \) two points on \( S \). According to this, given two points \( v_1 \) and \( v_2 \) on the model surface, the equivalence relation ‘\( \sim \)’ holds if and only if both the following conditions are fulfilled:

1. \( f(v_1) = f(v_2) \),
2. \( v_1 \) and \( v_2 \) are in the same connected component of \( f^{-1}(f(v_1)) \).

For each set of equivalent surface points, one node in the Reeb-graph is defined. Arcs of the graph are used to represent adjacent sets of equivalent surface points. An example of Reeb-graph construction is shown in Fig. 2. Values of function \( f \) represent points elevation and are quantized into 4 and 8 levels to yield the level sets shown in Fig. 2(a) and (c), respectively (different level sets are highlighted with different gray levels). Each connected component of a level set identifies a set of equivalent surface points and is represented with a node in the graph. The gray level of a graph node is the same as the gray level of the corresponding level set. Two disconnected components of the same level set are represented by two distinct nodes in the graph, both nodes with the same gray level. It can be noticed that a different quantization of values of \( f \) affects both the number of graph nodes as well as the topology of the graph. Generally, the number of quantization levels identifies the scale at which the topology of the surface is analyzed.
3 Mesh Decomposition

Choice of the $f$ function is a key element determining the topology of level sets, the structure of the Reeb-graph and its properties with respect to the original object. These properties include invariance or robustness of the graph structure with respect to model rotation, noise or slight irregularities of the model surface. Examples of different choices of the function $f$ range from the simple elevation of model points (this choice is particularly common for terrain modeling), to the distance measured from each point of the model to other reference points of the model, such as the model center of mass, curvature extrema or generic fiducial points [36, 37].

The proposed model for mesh decomposition develops on the assumption that distinguishing perceptual features of a 3D object relate to its protrusions. For a generic surface, a protrusion can be defined as a portion of the surface that sticks out of the remaining surface. Accordingly, a distinctive feature of points belonging to a protrusion is their distance to the other surface points: the higher the average distance of a point $v$ to all the other points of the surface, the higher the probability for $v$ to belong to a protrusion of the surface. To capture the above idea, the value of function $f$ on a point $v$ of the model surface is defined as the average geodesic distance (AGD) of $v$ to all the other points on the model surface [36].

In particular, assuming that the surface $S$ is approximated through a discrete mesh $M$ of $n$ vertices $v_1, \ldots, v_n$, the value of $f$ for a generic vertex $v_i$ of the mesh is computed as:

$$f(v_i) = \frac{1}{n} \sum_{j=1}^{n} \mu(v_i, v_j),$$

being $\mu(v_i, v_j)$ an estimate of the geodesic distance between vertices $v_i$ and $v_j$ of the mesh $M$. 

Figure 2: Construction of the Reeb-graph encoding topological features of a surface with respect to the elevation of surface points. (a) Level sets induced by quantization of elevation values into 4 levels. (b) The Reeb-graph of (a). (c) Level sets induced by quantization of elevation values into 8 levels. (d) The Reeb-graph of (c).
In order to provide for invariance of \( f \) values with respect to mesh scale, values of \( f \) are normalized so as to span in the range \([0,1]\):

\[
f_N(v_i) = \begin{cases} 
\frac{f(v_i) - f_{\min}}{f_{\max} - f_{\min}} & \text{if } f_{\max} \neq f_{\min} \\
0 & \text{otherwise}
\end{cases}
\]  

(3)

being \( f_{\min} \) and \( f_{\max} \) the minimum and maximum values of \( f \) on mesh vertices, respectively.

To simplify the notation, in the following we use \( f \) to refer to \( f_N \).

Computation of the geodesic distance on the piecewise linear decomposition of the manifold mesh \( M \) is accomplished through the Dijkstra’s algorithm and approximates the actual geodesic distance with the length of the shortest piecewise linear path on mesh vertices. Formally, given two vertices \( v_i \) and \( v_j \) on the mesh, a path \( R(v_i, v_j) \) from \( v_i \) to \( v_j \) is defined as an ordered sequence of vertices \((v_{r(1)}, \ldots, v_{r(k)})\) obeying the following constraints:

\[
\begin{align*}
&v_{r(1)} \equiv v_i \\
&v_{r(k)} \equiv v_j \\
&v_{r(h)} \text{ is adjacent to } v_{r(h+1)}, \forall h \in \{1, \ldots, k-1\}.
\end{align*}
\]  

(4)

For a generic path \( R(v_i, v_j) \), its length \( L(R(v_i, v_j)) \) is measured as the sum of the Euclidean distances \( d_E \) between adjacent vertex pairs:

\[
L(R(v_i, v_j)) = \sum_{h=1}^{k-1} d_E(v_{r(h)}, v_{r(h+1)}).
\]  

(5)

Based on the above assumptions, the geodesic distance between two vertices is approximated by the following expression:

\[
\mu(v_i, v_j) = \min_{R} L(R(v_i, v_j)).
\]  

(6)

Accordingly, for each vertex of the mesh the value of the AGD to all the other mesh vertices is computed.

Values of the AGD undergo to equalization so as to maximize discrimination among distance values. In image processing and analysis equalization is a normalization operation applied to image histograms to improve the local contrast of the image [43]. This is accomplished by effectively spreading out the most frequent intensity values, and can be generally applied to any histogram based representation. In our particular case, equalization enhances the difference between values of the AGD, so that the distribution of the equalized values is approximately uniform (at least ideally). For a mesh with \( n \) vertices, function \( f \) can assume up to \( n \) different values each corresponding to a different vertex. In the following, we indicate with \( p_i \) the \( m \leq n \)

13
distinct values of $f$ sorted in ascending order (i.e., $p_0 = \min_i f(v_i)$, $p_{m-1} = \max_i f(v_i)$, and $\forall i < j \Rightarrow p_i < p_j$), and with $q_i$ their equalized counterparts, that is $q_i = \tau(p_i)$, for $i = 1, \ldots, m$. In practice, the number of vertices with $p_i$ value is often reduced to just one or a few, so we actually consider $[p_i, p_i + \epsilon]$ as approximation of $p_i$, with $\epsilon$ equal to 0.01.

Since the equalization function $\tau(p_i)$ must be monotonic, for any $k$ the sum of the number of vertices with AGD between $p_0$ and $p_k$ must be equal to the sum of the number of vertices with equalized AGD between $\tau(p_0)$ and $\tau(p_k)$. If $G(q_i)$ and $H(p_i)$ are the histograms over $q_i$ and $p_i$, respectively, since $G(q_i)$ must approximate an uniform distribution it should have constant value $G(\tau(p_0)) = \frac{m}{\tau(p_{m-1}) - \tau(p_0))}$. According to this, it is easy to show that the equalized value $q_k$ of a generic $p_k$ is given by:

$$q_k = \tau(p_k) = \frac{\tau(p_{m-1}) - \tau(p_0)}{m} \sum_{i=p_0}^{p_k} H(p_i) + \tau(p_0),$$

(7)

where the maximum and minimum of the equalized values are $\tau(p_{m-1}) = 1$ and $\tau(p_0) = 0$, respectively.

As an example, Fig. 3 shows a model where vertices are colored based on the value of their AGD (original as well as equalized) to all the other vertices of the model. Lowest values of the distance are colored with red, highest values with blue. As expected, prominent protruding parts, corresponding to hands and legs, are colored with cyan-blue hue. Histograms of AGD values are also shown to highlight the effect of equalization.

It is worth noting that computation of geodesic distances may depend on the regularity of the mesh. In particular, since the Dijkstra’s algorithm approximates the actual geodesic distance with a sum of Euclidean distances between adjacent vertex pairs, in presence of non-triangular meshes (i.e., meshes with generic polygonal faces) or non-regular meshes (i.e., meshes with triangles of different sizes) approximation of the actual geodesic distance may be inaccurate. To avoid such drawback, meshes undergo to a pre-processing step accounting for triangularization and regularization before being processed for decomposition. Further, exact computation of the AGD would be quite costly, resulting in a time complexity of $O(n^2 \log n)$. To quickly compute a satisfactory approximation of AGD, we follow the algorithm of Hilaga et al. [36]. Accordingly, the geodesic distances are not calculated from all the surface vertices, but rather from a small number of evenly spaced vertices (base vertices) of the surface. The number of base vertices varies depending on the mesh area (i.e., base vertices are separated by a distance $2 \cdot \sqrt{0.005 \cdot S}$, being $S$ the area of the mesh).
Finally, it should be noticed that, from a theoretical point of view, the AGD function is not guaranteed to be a Morse function. A function is called Morse if all of its critical points are non-degenerate (i.e., if the Hessian matrix $H$ of the second derivative of the function is non-singular at those points). In particular, to be non-degenerate critical points should be isolated. For instance, the AGD is a constant function on a sphere, in which case every point is a degenerate critical point. However, in [44] it is shown that a function can be perturbed into a Morse function with surface wave traversal, provided that the mesh is properly subdivided.

In the remaining parts of this section, we present the construction and editing of the Reeb-graph induced by the normalized and equalized AGD (Sect. 3.1), and the refinement of the borders of the decomposed parts determined by a curvature analysis (Sect. 3.2). We point out that, to keep the discussion general, in the following we refer to the values of function $f$, though in our case we actually work on its equalized values $q$.

### 3.1 Reeb-graph Construction by Non-uniform $f$ Quantization

Many of the previous approaches to the computation of Reeb-graphs for mesh analysis rely on uniform quantization of $f$ values (see [36, 39]). However, the uniform quantization scheme may prove inadequate to capture distinguishing topological properties of a mesh. In fact, this is equivalent to performing the analysis of the mesh at a single scale. As it can be noticed in Fig. 2, depending on the quantization level, surface vertices belonging to perceptually different
parts are melted together into the same connected level set component (e.g., the lowest part of the vase handle). This effect could be reduced by progressively increasing the number of quantization levels. However, if increasing the number of quantization levels supports more accurate identification of part boundaries, it may also produce an over-decomposition of most of the object surface. Actually, a different solution is required to quantize $f$ values in order to preserve the accuracy of part identification without incurring in over-decomposition. For this purpose, a novel approach is proposed, that exploits non-uniform quantization of $f$ values and information about curvature of level sets to yield accurate identification of parts as well as of their boundaries.

Instead of back-projecting on the object surface an uniform quantization of $f$ values, we analyze values of the $f$ function so as to identify its critical points, that is, those values that correspond to a discontinuity of the number of connected components of the level sets. For this purpose, value of the $f$ function is computed at every mesh vertex and each $f$ value is associated with a measure $\mathcal{N}(f) : \mathbb{R} \mapsto \mathbb{N}$, representing the number of connected components of the corresponding level set on the object surface. $\mathcal{N}(f)$ is not continuous and values of $f$ which determine discontinuities of $\mathcal{N}(f)$ correspond to critical points of $f$, and detect topologically salient region boundaries on the object surface.

Detection of discontinuities of function $\mathcal{N}(f)$ is accomplished through an iterative merging approach based on adjacency and cardinality of connected components of level sets. At the first iteration step, values of $f(v_i)$ and $\mathcal{N}(f(v_i))$ are computed for each mesh vertex $v_i$. Each $f(v_i)$ value defines a degenerate interval $I_k = [f(v_i), f(v_i)]$ in $\mathbb{R}$, i.e., an interval for which the left and right extremity values coincide. Intervals $I_k$ defined at every vertex are ordered based on increasing $f$ values and for each interval the corresponding $\mathcal{N}(f)$ value is considered. In this way, each interval is associated with a data structure $D(I_k) \equiv (f(v_i), f(v_i), \mathcal{N}(f(v_i)))$ storing its bounds (both bounds are $f(v_i)$ at the beginning of the iterative merging process) and the number of its connected components. At the next iteration steps, contiguous intervals with the same number of connected components are merged.

Formally, given an ordered collection of $n$ disjoint intervals $\{I_k = [a_k, b_k]\}_{k=1}^n$ with $b_k \leq a_{k+1}$ $\forall k$, two intervals $I_l$ and $I_m$ are contiguous iff $\exists k$ $(b_l \leq a_k \leq a_m) \lor (b_l \leq b_k \leq a_m)$. Fusion of two contiguous intervals $I_l = [a_l, b_l]$ and $I_m = [a_m, b_m]$ associated with the same number of connected components $\mathcal{N}(a_l) = \mathcal{N}(b_l) = \mathcal{N}(a_m) = \mathcal{N}(b_m)$ yields a new interval with new bounds $[a_l, b_m]$. The number of connected components of the new interval $[a_l, b_m]$
is recomputed as it generally depends on the topology of the surface. Operatively, since the back-projection of \([f(v_i), f(v_i)]\) is often reduced to \(v_i\) itself or just a few vertices, the value \([f(v_i), f(v_i) + \epsilon]\) should be actually estimated. As reported in Sect. 3, in our solution this is performed during the equalization process.

This procedure is iterated until all contiguous intervals are associated with different values of the number of connected components. When the iterative process is complete, each vertex of the mesh is labeled based on the interval it maps to. However, sets of vertices with the same label are not necessarily connected, in that vertices located in several different parts of the surface can map into the same interval. As a consequence, connected components (regions) are identified by running a flood-fill algorithm adapted to the 3D case, which labels in the same way sets of adjacent vertices whose values of \(f\) fall in the same interval \(I_k\). These regions \(R_i\) are approximations of the equivalence classes of the Reeb-graph.

Each region is associated with an adjacency index, representing the number of its adjacent regions. Regions are merged based on their adjacency index: pairs of adjacent regions with the same adjacency index are iteratively merged.

According to this, the Reeb-graph of the model is derived by assigning nodes to regions, while edges of the graph connect nodes corresponding to adjacent regions. An editing operation is finally performed on the graph in order to remove small regions which can result at the end of the iterative merging process [45]. These regions usually correspond to parts that are not relevant for the object identification. In particular, a region on the object surface is considered a small one if the ratio of its area, computed by summing the areas of triangles in the region, and the area of the largest region of the object is less than a predefined threshold (set to 0.05 in our experiments).

Fig. 4 shows some representative steps of the Reeb-graph construction process. Fig. 4(a)-(b) shows the status of vertex labeling at two steps of the iterative merging process based on adjacency and cardinality of level sets. Fig. 4(c) shows the final status of vertex labeling based on connectedness analysis. It can be observed that disconnected regions (e.g., the two legs) associated with the same interval of \(f\) values are marked with different labels. Note that protrusions of small area, like the fingers, are removed in the processing step which edits the Reeb-graph by merging small regions to their neighbors.
Figure 4: Construction of the Reeb-graph for the model shown in Fig. 3(c): (a), (b) Dynamics of vertex labeling for two steps of the iterative merging process; (c) final vertex labeling based on connectedness analysis.

3.2 Curvature based Adjustment of Region Boundaries

Due to the discrete nature of the mesh, values of $f$ are sampled on a lattice thus preventing, in the general case, accurate identification of critical point values and consequently of part boundaries. Therefore, for accurate identification of part boundaries a different clue is exploited, relying on the evidence of the role played by deep surface concavities (i.e., loci of negative minima of each principal curvature) in perceptual surface decomposition [12]. In particular, accurate identification of object parts requires a process of refinement of region boundaries that should follow as much as possible deep concavities of the object surface.

As described in the previous section, each region boundary is a level set and is thus associated with a value $f^*$ of the $f$ function that corresponds to a critical point $f_c$. A perturbation of the $f^*$ value yields a new configuration of the boundary that moves to a new level set (see Fig. 5). Thus, profiles of region boundaries can be adjusted by changing the value of $f^*$ they are associated with. In this adjustment step, two independent and sometimes opposite requirements should be targeted: *locality* and *concavity*. The locality requirement demands that in order to comply to the topological characteristics of the object surface the optimal value $f^{opt}$ should be close to $f_c$ (the closest critical point of $f$). Differently, the concavity requirement demands that in order to identify a perceptually salient decomposition, the optimal value $f^{opt}$
should determine a boundary that matches a deep concavity profile on the object surface.

In the proposed solution, boundary adjustment is modeled as an optimization process by which the optimal value $f^{opt}$ that determines the boundary profile is found as the solution to a minimization problem. The objective function $C(f^*, f_c)$ includes two different terms accounting for locality and concavity requirements:

$$C(f^*, f_c) = (1 - \alpha)E_{local}(f^*, f_c) + \alpha E_{concave}(f^*),$$

being $E_{local}(f^*, f_c) = ||f^* - f_c||$, $\alpha$ a weighting coefficient (in our tests, it has been set experimentally to 0.5), and $E_{concave}(f^*)$ a measure of the concavity of the region boundary determined by $f^*$. Minimization of $C(f^*, f_c)$ is obtained using a plain gradient descent algorithm.

For the $E_{concave}(f^*)$ term, the sequence of $k_{\min}$ curvature values along the boundary profile is considered (principal curvatures have been computed according to the method proposed in [17]). In particular, let $c_{f^*,R}(t)$ be the curve—parameterized with respect to the normalized arc-length $t$—representing the portion of the $f$-valued level set corresponding to the boundary of region $R$. $E_{concave}(f^*)$ is defined as follows:

$$E_{concave}(f^*) = \min_{f^*}(k_{\min}(c_{f^*,R}(t)) \otimes G_\sigma(t)),$$

being $k_{\min}(\cdot)$ a function returning the $k$-min curvature value at surface points, $\otimes$ the convolution operator and $G_\sigma(\cdot)$ the normalized Gaussian smoothing kernel of variance $\sigma^2$. In this equation, convolution with a Gaussian kernel permits to smooth the values of the $k$-min principal curvature along the boundaries of the region, thus making more stable and reliable the identification of the minimum.
4 Experimental Results

In the following, we present and discuss experimental results that show the potential of the proposed solution for decomposition of 3D objects. First, we evaluate the effectiveness of decomposition considering two different points of view: subjective and objective. For both the evaluations we compare results of the proposed approach against results obtained using two state of the art solutions for 3D objects decomposition. Detailed descriptions of the subjective and objective evaluations are reported in Sect. 4.1 and Sect. 4.2, respectively. Then, in Sect. 4.3 we discuss our approach with respect to the seven criteria for evaluating mesh decomposition algorithms enumerated in [14], and provide further decomposition examples for specific object classes.

The subjective evaluation is accomplished by reporting decomposition results for a set of reference 3D models covering a variety of objects with different structural complexity and resolution. This evaluation is referred to as subjective because we do not provide objective performance measures of the decomposition approaches under analysis. Rather, performance of each approach is evaluated subjectively by comparing decomposition results on the set of reference 3D models. This approach to the evaluation of decomposition results conforms to the approach followed in other works [11, 19, 24, 26, 27].

Differently, the objective evaluation is accomplished by defining a measure of the perceptual saliency of decompositions. For this purpose, a ground-truth set of annotated 3D objects has been constructed through a user based experiment. In the experiment, some people were asked to identify relevant parts of 3D models representing real objects of different structural complexity. Answers were used to assess the degree by which decomposition provided by the approaches under comparison matched relevant parts identified by users.

In particular, we compared the proposed solution based on Reeb Graph (RG), to a geometric based solution, the curvature clustering (CC) [19], and to the semantic oriented method based on spectral embedding and contour analysis (SE) [26]. The use of these approaches is motivated by the fact that methods in the geometric based category exploit different characteristics of the object surface than RG (basically CC uses the sole curvature information to produce object decomposition), while the SE method belongs to the semantic oriented category, as RG, thus permitting a comparison between methods that pursue the same objectives and exploit similar characteristics of 3D models. The two approaches are shortly summarized in the following.

Mesh decomposition based on CC is performed by extracting values of principal curvatures
\(k\)-min and \(k\)-max [46], and clustering them using standard \(k\)-means (the number of clusters is set manually based on the structural complexity of 3D objects). Once clustering is complete, each mesh vertex is labeled according to the cluster it belongs to. Connected component analysis and region growing are performed to remove small regions and identify relevant object parts. Since the region labeling tends to produce over-partitioning, a merging phase eventually combines together adjacent regions based on their curvature and boundary information.

SE performs mesh decomposition via recursive bisection, where at each step a sub-mesh embedded in 3D is first spectrally projected into the plane and then a contour is extracted from the planar embedding. The graph Laplacian and a geometric operator designed to emphasize concavity are used to compute the projection. The two embeddings reveal distinctive shape semantics of the 3D model and complement each other in capturing the structural or geometrical aspect of a partitioning.

### 4.1 Decomposition of Sample 3D Objects

Fig. 6 presents decomposition results for some reference models using RG (leftmost column), CC (central column) and SE (rightmost column) approaches. Reference models include 3D objects that are commonly used to compare partitioning methods (like the Stanford horse and bunny), and other models of different structural complexity in terms of local surface variations and presence of large protrusions.

These examples confirm the main drawbacks of CC approach that were evidenced in Sect. 1.1. In fact, it can be noticed that perceptually salient parts do not always correspond to parts with homogeneous curvature. Rather, identification of regions with homogeneous curvature is often associated with over-decomposition of perceptually salient parts (e.g., the body of the rabbit, the legs of the horse, the arms, legs and head of the human body). Using CC, better results are obtained with the decomposition of mechanical components (see for example the oil-pump in Fig. 6(d)), where the presence of large areas of uniform curvature better fits the requirements of the approach.

Regarding SE, it provides nice partitioning in all the cases reported in Fig. 6. However, for some models (like the horse in Fig. 6(a), and the human model in Fig. 6(c)) a non-intuitive decomposition of the body is evidenced (e.g., in the horse model the shoulder region is identified as a separate part). The RG approach produces satisfactory results for all the reference models. Some criticality are shown in separating the back legs from the body of the rabbit in Fig. 6(b), or in identifying some minor details of the oil-pump model in Fig. 6(d).
Figure 6: Decomposition of sample objects using the proposed approach (RG, leftmost column), the curvature clustering (CC, central column), and the spectral embedding (SE, rightmost column). Different colors are used to evidence regions identified by the decomposition. The vertex count is also reported for every model.
Results obtained from these sample models and tests performed on a larger data set permit to derive some general considerations about the behavior of the RG approach. As expected, RG performs very well on objects with manifest protrusions, proving to be better than the experimented geometry based solution in decomposing objects into their salient parts, and having similar performance to a state of the art semantic oriented solution. Under this perspective, it is relevant to observe that since geodesic distances basically do not change under pose variations, decompositions of articulated objects are largely invariant with respect to pose of object parts. The RG approach also tends to ignore small curvature variations, thus avoiding over-decomposition problems. Moreover, the approach performs quite well also in the case of low resolution objects (i.e., the mesh approximating a 3D object is represented by a small number of vertices and triangles). In contrast, solutions based on curvature estimation feature a loss of performance in decomposing low resolution objects due to the difficulties to obtain a reliable estimation of curvature values (which requires high resolution meshes).

4.2 Perceptual Saliency of Decomposed Objects

A key aspect of mesh decomposition approaches relates to the ability to yield object decomposition into perceptually salient parts. In order to quantitatively measure the saliency of 3D object decompositions, a test has been carried out aiming to compare results of manual versus automatic objects decomposition. The comparison has been carried out on a ground-truth archive comprising 17 reference 3D models (shown in Fig. 7) which represent real objects of different structural complexity. In particular, models belong to five categories: surfaces of revolution (SOR), vases, miscellaneous, mechanical (CAD) and medical. These five categories aim to provide a not-biased reference archive, capable to evidence the strength and weakness of different approaches for mesh decomposition. This is obtained by including objects of different complexities and surface characteristics.

The SOR category comprises objects that can be generated through the revolution of a generative profile. As a consequence, these objects have regular shapes with large parts of the surface presenting homogeneous differential properties. In particular, these models present a few and large discontinuities in the surface curvature, that occur along precise and well defined circular lines. These models are expected to favor decomposition methods based on curvature information of objects surface. Objects in the vases category are slightly more complex than SOR, still representing simple man made objects with minor protrusions. The third category comprises miscellaneous models representing animals and compound objects with complex
shape and major protrusions. With this latter category of objects, detection of perceptually salient parts is most challenging. The fourth category includes models of mechanical components that show small protrusions and large areas with uniform curvature. Finally, the medical category includes models of different complexities both in terms of curvature variations, number and relevance of protrusions.

To obtain the ground-truth decomposition of reference objects, we asked 30 users to identify the relevant parts of the 17 reference 3D models. In order to make the evaluation reproducible, we outlined the following experimental protocol. The users were selected among university students and academic personnel with sufficient skill in the use of software applications. The 3D Studio Max\(^1\) software was used to allow users to easily identify relevant parts of 3D shapes. The application interface supports basic 3D navigation functionalities plus a specific tool which permits interaction with 3D objects and to colorize their parts or regions. In this way, different colors are used to evidence different parts. During each test session, the user is first shortly trained with the interface and then asked to identify "perceptually relevant parts of an object".

The limited size of the database is mainly motivated by the time requested by users to complete objects decomposition. In particular, we observed that the testing session took an average time ranging between 19 and 23 minutes, with an average of 21.3. This appeared to be a realistic limit for the user capability and willingness in maintaining attention throughout the test.

Thus, the ground-truth archive includes 510 decomposed models, the decomposition of the \(i\)-th reference model by the \(j\)-th subject being referenced as \(m^i_j\) and corresponding to \(N(m^i_j)\) regions \(\{R_1(m^i_j), R_2(m^i_j), \ldots, R_{N(m^i_j)}(m^i_j)\}\). This ground-truth archive is used to score the decomposition of reference 3D models operated by any automatic decomposition approach. To clarify how a generic decomposition is scored, the degree of overlap and the best overlap are defined.

Given two regions \(R_a\) and \(R_b\) on the surface of a 3D model \(m\), their degree of overlap \(O(R_a, R_b)\) amounts to the number of model vertices that belong to both \(R_a\) and \(R_b\), normalized to the maximum number of vertices of \(R_a\) and \(R_b\):

\[
O(R_a, R_b) = \frac{\#(R_a \cap R_b)}{\max\{\#(R_a), \#(R_b)\}}. \quad (10)
\]

Given a region \(R_a\) on the surface of a 3D model \(m\), and a decomposition of \(m\), e.g., \(m^j = \{R_1(m^j), R_2(m^j), \ldots, R_{N(m^j)}(m^j)\}\), the best overlap of \(R_a\) onto \(m^j\) amounts to the

\(^1\)\copyright \ 2016 \ Copyright by Autodesk.
Figure 7: The ground-truth archive including 17 objects with different structural and surface complexities.

The highest value of the degree of overlap between $R_a$ and any region of the decomposition, that is:

$$bo(R_a, m^j) = \max_k O(R_a, R_k(m^j)).$$  \hspace{1cm} (11)$$

The best overlap can be used to quantify the agreement of two decompositions. Let $m^S$ be
the decomposition of model $m$ using the solution $S$ (S can be RG or CC or SE). The agreement $A$ between $m^S$ and $m^j$ is evaluated as the average value of the best overlap of regions of $m^S$ onto regions of $m^j$, that is:

$$A(m^S, m^j) = \frac{1}{N(m^S)} \sum_{k=1}^{N(m^S)} bo(R_k(m^S), m^j).$$

(12)

The agreement value is maximum, that is 1, when two decompositions are exactly the same. The lower the agreement value, the more different the two decompositions.

Fig. 8 shows ranges of the agreement for the three approaches (RG, CC and SE) on the five categories of objects included in the ground-truth archive.

Results show different behaviors for the five categories, the RG and SE approaches featuring the best performance on average. For the SOR category, on average, the three approaches score the best results in terms of average values (75.48%, 75.77% and 78.44% for RG, CC and SE, respectively). As expected, a curvature based approach like CC performs well in this case, but it is relevant that also RG and SE are capable to score results comparable to CC. Similar results are shown for the CAD models, where the CC approach is competitive with the semantic oriented solutions under test.

For the vases category, the agreement for CC is consistently lower (decreasing to 60.86%), while the mean value of the agreement for RG and SE keeps almost constant (76.08% and
77.65% for RG and SE, respectively) with respect to the results scored for the SOR category.

As expected, for the miscellaneous category including complex objects, the agreement is much lower especially for CC (45.84%), with RG outperforming the other approaches. This can be accounted to the fact that topological information is more suited to capture object protrusions as that appearing in complex objects. It should be noticed that the higher the object complexity, the lower is the agreement between the automatic and manual decomposition. In general, the difficulty to decompose complex objects is confirmed by the fact that whereas almost all subjects provide the same decomposition of simple objects, complex objects are decomposed in a variety of different ways, even by human subjects.

Finally, for the class of medical models, the RG performed the best with an agreement of 62.34% with respect to the 60.03% of SE and the 50.80% of CC.

4.3 Evaluation of the Approach

In a recent comparative study, Attene et al. [14] enumerated seven criteria for evaluating algorithms for mesh decomposition. In the following, we discuss the proposed approach with respect to these criteria.

Type of decomposition: The approach belongs to the class of semantic oriented solutions for mesh decomposition. Figure 6, shows the capability of the approach to partition complex objects (including animals, human and mechanical objects) into meaningful parts.

Extracting the “correct” parts: To overcome the difficulty in defining which the correct parts are, the user-based test reported in Sect. 4.2 has been defined and carried out. It aims to provide an evaluation of the decomposition results based on the viewers’ perspective. Results of the evaluation show the proposed solution is competitive with state of the art semantic oriented methods and can outperform geometry based approaches.

Boundaries: Minimization of a functional that accounts for proximity and surface curvature has been proposed in Sect. 3.2 to refine the position of the boundaries between objects parts.

Hierarchical/multi-scale decomposition: In the proposed solution, mesh decomposition is performed at a single scale. Multi-scale extension of the approach could be obtained by modifying the editing criteria of the Reeb-graph and the rule that eliminates small regions. Multi-resolution could improve results particularly in difficult cases where the scale at which to partition an object depends from the application (e.g., the hand/fingers of the human model of Figure 6(c)). The viability of a multi-scale extension is not addressed in this paper and will be part of future work.
Sensitivity to pose: The proposed approach is almost pose invariant (i.e., same models in different poses are decomposed in the same way). An example is reported in Fig. 9. Though the eight stylized hand models have different number of vertices and are not all derived from the same hand model, it can be noted they are decomposed in very similar ways, with slight differences in the locations of the boundaries between the fingers and the palm of the hand.

Asymptotic complexity: In the construction of the Reeb-graph, the Dijkstra’s algorithm takes $O(n \log n)$, with $n$ number of vertices of the mesh. Application to every mesh vertex induces a $O(n^2 \log n)$ complexity. Merging of the intervals in the worst case has complexity $O(n^2)$. Finally, complexity in boundary refinement depends on the number of contours and on the plain gradient descent algorithm used to optimize Eq. (8). Thus, the overall complexity is dominated by $O(n^2 \log n)$. Table 1 reports the running times required for decomposing meshes with different number of vertices.

Control parameters: Low level parameters used by our approach are the threshold for removing small regions and the parameter $\alpha$ which controls the adjustment of region boundaries according to Eq. (8). As an example, Figure 10 reports the effect of changing the threshold that regulates the pruning of small regions during the post-processing of the Reeb-graph (see Sect. 3.1). In Fig. 10(a)-(b), the decompositions obtained with this threshold set to 0.05 (i.e., the default value) and 0.01 for two different models are reported. It can be observed that acting on this parameter, those details corresponding to smaller protrusions of the models can be identified.

In addition to the above evaluation, tests have been performed to evidence the robustness of the proposed approach with respect to modifications of 3D models induced by surface noise, smoothing and simplification, and with respect to 3D models with varying topology (e.g., surfaces with different genus).

In particular: Model in Fig. 11(b) is obtained from the model in Fig. 11(a) by adding Gaussian noise over the vertices in the normal direction (80% of the model’s bounding ball radius); Model in Fig. 11(c) is obtained from the model in Fig. 11(a) by iterating 10 steps of Laplacian smoothing; Model in Fig. 11(d) is a simplified version with 10% of the vertices of the model in Fig. 11(a) (the original model has 36653 vertices). It can be observed the decompositions are almost invariant with respect to these variations.

Figure 12 shows the decomposition of objects with genus-1 and genus-2. A limitation in decomposing the genus-2 object can be observed. Basically, the correct parts of the feline
Figure 9: Decompositions of eight raw hand models in different poses ((a)-(g) are right-hand models, (h) is a left-hand model).

Figure 10: Decompositions of two objects for different settings of the threshold that regulates the pruning of small regions of the Reeb-graph. The threshold is equal to 0.05 for the models on the left in (a)-(b), and to 0.01 for the models on the right in (a)-(b).

Further decompositions are reported in Fig. 13 and Fig. 14. Fig. 13 shows decompositions of CAD models of mechanical components. In general, decompositions of mechanical objects using RG can be not accurate, in that surfaces that are perceived as individual components can be split into different parts (see for example the rocker arm in Fig. 13(a)). In Fig. 14, decompositions of medical models are reported.

Fig. 15, reports decompositions of a set of objects that are commonly used in the evaluation of algorithms for 3D mesh decomposition. Using these models, decompositions produced by the proposed approach can be easily compared with those of different methods (for example,
Figure 11: Decomposition of a sample hand model (a) and of its variations obtained by: (b) adding Gaussian noise; (c) iterating Laplacian smoothing; (d) Simplification.

Figure 12: Decomposition of a genus-1 (on the left) and a genus-2 model (on the right).

Figure 13: Decompositions of CAD models.

(a) rocker arm (10044)  (b) crank (9385)  (c) screwdriver (7152)

In the large part of applications, decomposition approaches run off-line. As a consequence, their computational complexity typically is not a stringent constraint. For comparison purposes, Table 1 reports the number of regions in the final decomposition of some sample models and the running times on a 2.2 GHz Centrino Duo PC with 2GB memory (complexities of the
Figure 14: Decompositions of medical models.

Figure 15: Decomposition of objects that are commonly used as reference.

objects are reported in number of vertices). A non-optimized Java implementation was used.

<table>
<thead>
<tr>
<th>Model</th>
<th>vertices</th>
<th>regions</th>
<th>time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>man (Fig. 6(c))</td>
<td>23280</td>
<td>6</td>
<td>71</td>
</tr>
<tr>
<td>hand (Fig. 11(a))</td>
<td>36653</td>
<td>7</td>
<td>142</td>
</tr>
<tr>
<td>hand (Fig. 11(d))</td>
<td>3665</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>feline (Fig. 12(b))</td>
<td>36000</td>
<td>9</td>
<td>139</td>
</tr>
<tr>
<td>horse (Fig. 15(a))</td>
<td>9207</td>
<td>7</td>
<td>40</td>
</tr>
<tr>
<td>bird (Fig. 15(b))</td>
<td>101</td>
<td>4</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>cow (Fig. 15(c))</td>
<td>17438</td>
<td>7</td>
<td>54</td>
</tr>
<tr>
<td>jug (Fig. 15(d))</td>
<td>504</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>dinopet (Fig. 15(e))</td>
<td>13209</td>
<td>8</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 1: Number of vertices, regions of the decomposed model and running times for meshes of different complexities.

5 Conclusions and Future Work

In this paper, we have presented and discussed the development and experimentation of a technique for automatic decomposition of 3D models. This can be relevant in many practical applications, such as modeling, metamorphosis, compression, simplification, 3D shape retrieval, collision detection, texture mapping and skeleton extraction. The proposed solution relies on Reeb-graphs to accomplish a structural based decomposition of an object into parts.
In addition to mesh topology, curvature information is exploited to refine object decomposition and adjust region boundaries so as to match deep surface concavities and to yield perceptually salient decomposition of objects. Comparative evaluations are presented both in terms of sample object decompositions and in terms of a test measuring the agreement of the decomposition to perceptually relevant parts of objects. Critical discussion of the approach to the light of seven evaluation criteria commonly used by the object decomposition community is also presented.

Future work will investigate a multi-resolution extension of the proposed approach and its inclusion into a scheme for retrieval by content of 3D objects based on similarity between object parts.

6 Acknowledgments

This work is partially supported by the Information Society Technologies (IST) Program of the European Commission as part of the DELOS Network of Excellence on Digital Libraries (Contract G038-507618).

Very preliminary versions of this work appeared in [39] and [40].

References


