Topology driven 3D mesh hierarchical segmentation

Julien Tierny, Jean-Philippe Vandeborre, Mohamed Daoudi

Topology driven 3D mesh hierarchical segmentation

Julien Tierny[1], Jean-Philippe Vandeborre[1,2] and Mohamed Daoudi [1,2]
[1] LIFL (UMR USTL/CNRS 8022), University of Lille, France
[2] GET / INT / TELECOM Lille 1
{*tierny, vandeborre, daoudi*}*@lifl.fr*

Abstract

In this paper, we propose to address the semantic-oriented 3D mesh hierarchical segmentation problem, using enhanced topological skeletons *[18]. This high level information drives both the feature boundary computation as well as the feature hierarchy definition. Proposed hierarchical scheme is based on the key idea that the topology of a feature is a more important decomposition criterion than its geometry.*

First, the enhanced topological skeleton *of the input triangulated surface is constructed. Then it is used to delimit the* core *of the object and to identify* junction areas*. This second step results in a fine segmentation of the object. Finally, a fine to coarse strategy enables a semantic-oriented hierarchical composition of features, subdividing human limbs into arms and hands for example.*

Method performance is evaluated according to seven criteria enumerated in latest segmentation surveys [3]. Thanks to the high level description it uses as an input, presented approach results, with low computation times, in robust and meaningful compatible *hierarchical decompositions.*

1. Introduction

Mesh segmentation consists in subdividing a polygonal surface into patches of uniform properties [16, 3], either from a strictly geometrical point of view or from a *perceptual* point of view. This operation has become a necessary pre-processing tool for many applications such as shape modeling [7], deformation [10], compression, texture mapping [20], retrieval, etc.

According to Attene et al. [3], mesh segmentation techniques can be classified into two categories. On the one hand, *geometry-oriented* methods aim at defining patches of homogeneous geometry. In this case, algorithms are driven by purely low level geometrical information, such as curvature [19]. On the other hand, *semantic-oriented* methods aim at distinguishing regions of perceptual interest, following high level notions such as defined in human perception theory [6]. This kind of approach is particularly suited for human shape interaction applications (texture mapping, modeling, deformation, etc.), where the decomposition has to be meaningful from a human user's point of view.

In this latter case, three main difficulties arise. (i) Characterizing high level notions from low-level measurements (such as curvature or geodesic distance evaluations) remains an open issue [3]. (ii) For most applications, a hierarchical decomposition scheme is expected, so as to provide a progressive understanding of the object, but it is often based on low-level measurement clustering [10, 2]. (iii) The *compatibility* of the segmentation – which means the ability to identically segment objects that are visually similar – has been recently expressed as a new challenging problem [9].

To address these issues, Katz et al. [9], using multi-dimensional scaling, introduce the high level notion of *feature points* – vertices located on the extremity of prominent components – to drive the segmentation of the object into core and features. Then, only patches that contain feature points are recursively sub-divided. This means that, at the finest levels of the hierarchy, the more the patches will be located near the feature points, the smaller they will be (fig. 5), which is not a *natural* hierarchical scheme.

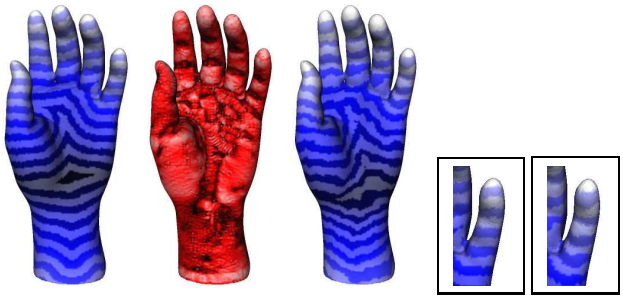
Podolak et al. [14] propose to drive the segmentation by the notion of symmetry. They describe a clustering-based algorithm using planar-reflective symmetry transform values as an input. However, the depth of their hierarchical scheme is user-defined. Moreover, the symmetry of an object depends on its pose, which impacts the global pose robustness of the overall approach.

Li et al. [12] propose to construct a skeleton of the object and to use this high level structural information to drive the segmentation. Lien et al. [13] extend this idea constructing a hierarchical skeleton, analyzing the principal axis of its convex hull. However, this method is reported to be time consuming when dealing with non-null genus surfaces. Moreover, the convex hull of an object is dependent on its pose, which also impacts the pose robustness of the overall algorithm.

Berreti et al. [4] propose a method that overcomes above

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

Figure 1. Enhanced topological skeleton extraction overview [18].



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

Figure 2. Original geodesic based function (a). Curvature index [11] (b). Curvature constrained geodesic based function (c). In (d), level sets are aligned with concave areas.

issue using topological skeletons [5], defined with regard to a geodesic based mapping function computed on the surface. However, their algorithm is not hierarchical and only segments objects into core and limbs. Zhang et al. [20] use topological skeletons as well but also propose to segment obtained patches along *separating* regions, where the area of the level sets of the mapping function suddenly varies.

In this paper, we propose to address the three previously mentioned segmentation related issues. Firstly, as the structure of an object is an important perceived shape characteristic, we propose to drive the segmentation by the topology of the object, using an enhanced version of topological skeletons [18]. Secondly, as this shape abstraction also encodes shape geometry, we propose a semantic-oriented hierarchical segmentation process that gathers object's features according to their topology and their geometry. Thirdly, as segmentation is topology driven, objects of compatible topology are segmented in a compatible manner.

This paper makes the following contributions. In the next section, we show how to adapt shape topological description techniques to the segmentation problem, with an appropriate surface mapping function. In the third section, we show how to use this high level information for the feature boundary definition. In the fourth section, we show how to benefit from the topological description of the shape to achieve a semantic-oriented feature hierarchy. Experimental results are presented in the fifth section, where method performance is evaluated according to criteria enumerated in latest surveys [3].

2. Fitting enhanced topological skeletons to the segmentation problem

In the first stage of the method, given an input triangulated surface T (manifold, connected and closed), an enhanced topological skeleton is extracted. The following paragraph sums up previous work [18] presenting this process.

2.1. Enhanced topological skeleton extraction

First, mesh feature points (in pink in figure 1(a)) are automatically extracted, intersecting geodesic-based map extrema [18]. Then, for each vertex in the mesh, a mapping function f_m is defined as the geodesic distance to the closest feature point (fig. 2(a)). Next, for each vertex $v \in T$, an upper-value approximation of $f_m^{-1}(f_m(v))$, noted $\Gamma(v)$, is computed along the edges of T. In particular, the connected component of $\Gamma(v)$ containing v is identified and noted $\gamma(v)$. Analyzing the evolution of the number of connected subsets of $\Gamma(v)$ as f_m evolves enables the construction of a *Reeb graph* [15] (fig. 1(b)). At this stage, each connected component of the Reeb graph is modeled with an ordered collection of closed curves $\gamma(v)$. The next step consists in identifying *constrictions* [8] within these collections. For each connected component of the Reeb graph, the average Gaussian curvature on each curve $\gamma(v)$ is computed. Then, local negative minima are identified as constrictions (fig. 1(c)). Finally, the connected components of the Reeb graph are subdivided using these constrictions as boundaries between subparts (fig. 1(d)). As a conclusion of this algorithm, the input surface is represented by an enhanced topological skeleton (fig. 1(d)), which encodes the topological and geometrical evolution of the contours of the mapping function f_m.

2.2. Constriction location optimization

In this paper, we use such a high level description to drive the segmentation process. In particular, as identified constrictions are located on the narrowest parts of the surface, they are good candidates for feature boundary definition (fig. 1(c)). However, in the latter algorithm, constrictions are identified along level line approximations, which do not necessarily follow the concavity of the surface. Within the framework of segmentation, to conform to human perception theory [6], feature boundaries must be aligned with concave areas. Consequently, we force the alignment of the mapping function level lines with surface bottlenecks, by integrating surface curvature into the geodesic distance computation. Such a curvature constrained geodesic distance evaluation is computed with Dijkstra's algorithm, us-

ing $w(v_i, v_j)$ as weight for every edge (v_i, v_j):

$$w(v_i, v_j) = \frac{e(v_i, v_j)}{\overline{e}} + \alpha \times \frac{|I_c(v_i) - I_c(v_j)|}{\overline{\Delta I_c}} \quad (1)$$

where $e(v_i, v_j)$ is the euclidean distance between v_i and v_j, $\overline{e}$ the average edge length in T, α a predefined ratio, $I_c(v)$ the curvature index [11] in v ($I_c(v) \in [-1;1]$) and $\overline{\Delta I_c}$ the average curvature index difference between two adjacent vertices in T. The motivation of this computation is to increase the distance between two vertices when a concave region separates them. Consequently, when $I_c(v_i)$ or $I_c(v_j)$ (or both) are negative (which corresponds to concavity), α is set to α_0 and 0 otherwise. Therefore, level lines *quickly* visit convex areas and *slowly* get aligned with concave ones. This effect can be observed in figure 2, which shows on a hand model the mapping function computed in [18] (defined as the geodesic distance to the closest feature point), the curvature index distribution and the mapping function used in this paper (defined as the curvature constrained geodesic distance to the closest feature point, noted $\widehat{f_m}$). In particular, the reader can notice in figure 2(d) that displayed level sets are aligned with the concave parts of the finger. In these illustrations, α_0 has been set to a high value for display purpose. In practice, it has been set to 0.05 for every model. Once this curvature constrained function is computed, the Reeb graph is constructed. Then, for each connected component of the graph, a curvature estimation $\zeta(\gamma(v))$ is computed for each of its curve $\gamma(v)$ as follows:

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

where $\mathcal{P}(\gamma(v))$ is the perimeter of $\gamma(v)$ and where $\mathcal{L}_{e_1}(v_i)$ and $\mathcal{L}_{e_2}(v_i)$ are the lengths of the edges adjacent to v_i on $\gamma(v)$. This computation gives a robust and relevant estimation of the curvature along each curve $\gamma(v)$. Then, curves that locally minimize $\zeta(\gamma(v))$ are identified as constrictions.

In this section, we introduced an adaptation of shape topological description techniques to the segmentation problem. We proposed a novel mapping function, based on a modified computation of geodesic distances. This computation integrates curvature so that the mapping function level lines are forced to be located along concave areas. Then, estimating the curvature on these collections of curves enables the detection of the most concave ones, that we refer as constrictions. In the next section, these constrictions are used for feature boundary computations.

3. Feature boundary computation

In the previous stage of the approach, a modified version of enhanced topological skeletons has been computed. Each node of this skeleton references a surface patch of the mesh. In figure 3(b), each patch has been displayed with a distinctive color, resulting in an over-segmentation of the object. In this section, we describe a region merging algorithm based on the notions of *core* and *junction areas*, which results in a fine segmentation of the object (fig. 3(d)).

(a) Original graph. (b) Raw segmentation. (c) Simplified graph. (d) Fine segmentation.

Figure 3. Core and junction computation results in the finest segmentation of the object.

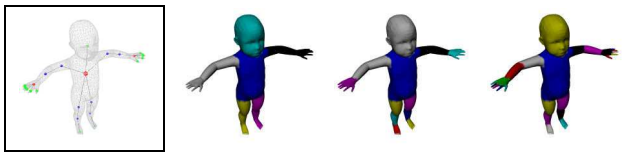
3.1. Junction areas

We classify the nodes of the skeleton into three categories, according to their degree. A node N is an *extremity node* if its degree ($deg(N)$) equals 1 (related surface patch contains a feature point, in green in fig. 3(a)). A node N is a *tubular node* if $deg(N) = 2$ (in blue in fig. 3(a)). Its surface patch has two boundaries: an *outer* one (γ_o), directed towards feature points, and an *inner* one (γ_i). A node N is a *junction node* if $deg(N) > 2$ (in red in fig. 3(a)). We define a *junction area* as a connected set of junction nodes. Concretely speaking, junction areas correspond to surface patches adjacent to several large features, such as the palm of the hand on a humanoid model. In figure 3(a), the set of four red nodes forms a junction area. At this stage of the algorithm, junction nodes are merged into junction areas.

3.2. Core boundary definition

We define the *core* of an object as a connected set of nodes, including the *root* node of the skeleton and its possible adjacent junction areas. The root node is displayed in figure 3(a) with a bigger radius. It corresponds to the surface patch containing the vertex which is the furthest away from feature points. Concretely speaking, the core of the object corresponds to its global inner part, like the torso of a humanoid for example. The presence of tubular nodes on prominent components prevents an undesired expansion of the core. In figure 4(d), the core expansion is stopped at the basis of the head, the arms and the legs because constrictions have been detected on these features.

An additional mechanism is proposed to optimize core boundary computation. Constrictions are often identified at the basis of prominent components, delimiting thin patches. Consequently, to prevent over-segmentation in those areas (see the fingers in figure 3(b)), we propose to merge the first tubular node of a component with the core in the following cases:



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

Figure 4. Hierarchical segmentation of a baby model.

1. Related surface patch is small: its interval on $\widehat{f_m}$ is lower than a fixed threshold $\Delta\widehat{f_m}$ ($\Delta\widehat{f_m} = 0.1$);
2. Its outer boundary (γ_o) is a better candidate for core boundary definition than its inner boundary (γ_i): $\zeta(\gamma_o) < \zeta_0$ and $\zeta(\gamma_o) < \zeta(\gamma_i)$ ($\zeta_0 = -0.2$).

In figure 3, the first tubular node of the inch (white patch in fig. 3(b)) is large but its outer boundary (with the cyan patch) is deeply concave and better suits the purpose of segmentation as a delimitation between the inch and the palm. This tubular node is merged into the core. The first condition makes, for example, the first tubular node of the middle finger (pink patch in fig. 3(b)) merge with the core.

As a conclusion, in figure 3, junction nodes first merges to form a junction area. Then this junction area merges with the root node. Finally, core boundary is optimized merging first tubular nodes if necessary (see figure 3(c)).

4. Topology driven feature hierarchy

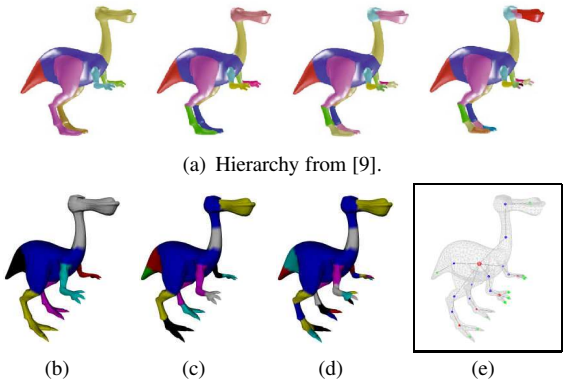
In the previous stage of the approach, a fine segmentation of the object has been obtained running a node merging algorithm on the skeleton (figs. 4(a), 5(e)). Based on this fine segmentation, we propose in this section a fine to coarse strategy for the semantic-oriented composition of features.

Traditional hierarchical segmentation algorithms [10, 9, 2] base their hierarchy scheme on low level measurement clustering. In this paper, we propose to drive the hierarchy composition using a high level shape description so as to provide a semantic-oriented feature hierarchy. The key idea of our hierarchical scheme is that the topology of a feature is a more important decomposition criterion than its geometry. Consequently, we propose to subdivide features in priority along junction areas, which corresponds to skeleton's topological variations.

First, each core-adjacent connected component of the skeleton is identified as a *feature*. This identification results in the coarsest level of hierarchy (figs. 4(b) and 5(b)).

Then *features* are recursively subdivided according to the following strategy. Each *feature* is swept from its inner boundary to its extremity (extremity node). If a junction area is encountered, the previously swept nodes are gathered into a new *feature*, and the junction area is gathered with the remaining nodes into another *feature*. If there is no junction area in the feature, it is subdivided into the nodes it is composed of and the recursive algorithm stops.

In figure 4(a), the baby's left arm is swept from its basis until the hand palm (junction area, in red) is reached. The two tubular nodes (in blue) are gathered into a feature (in white in fig. 4(c)) and the hand into another (in pink in fig. 4(c)). At this level of hierarchy, arms are divided into hands and arms. Then, these features are subdivided into the nodes they are composed of in the last hierarchical level.



(a) Hierarchy from [9].

(b) (c) (d) (e)

Figure 5. Comparison to the hierarchical decomposition from [9]. At the second level, our algorithm segments dinopet's limbs into arms and hands or legs and feet.

In this section, we presented a semantic-oriented hierarchical mesh decomposition algorithm. Contrary to state of the art methods, it subdivides features according to their topological characteristics, providing meaningful decompositions of limbs into arms and hands or legs and feet (fig. 5(c)) for example, which benefits shape understanding.

5. Experiments and results

In this section, we present experimental results obtained on manifold, connected and closed triangulated surfaces extracted from the Princeton [17] and the INRIA [1] shape repositories.

5.1. Segmentation evaluation

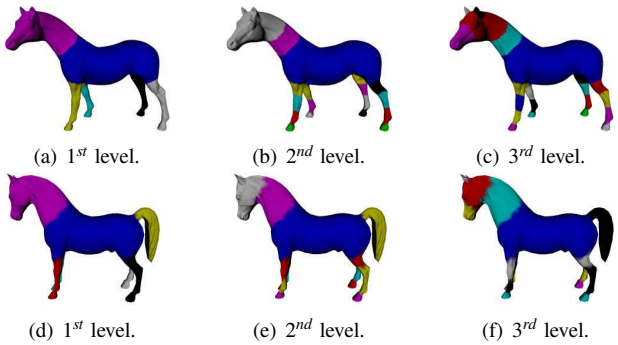
To our knowledge, no ground truth 3D mesh segmentation evaluation process has been proposed in the past. However, Attene et al. [3] enumerate seven criteria for evaluating a segmentation algorithm.

1. Versatility: figure 8 shows that visually meaningful segmentations are obtained on primitive shapes (with genus 2), animals, anatomic and mechanical parts.

2. Decomposition *semantic*: in figures 4, 5, and 8, animals and humanoids are first segmented into core and limbs. Moreover, figure 6 shows a compatible segmentation of two horse models (into core and limbs, at first level).

3. Boundary location: a curvature constrained geodesic distance computation has been proposed in section 2 so as to align constrictions along concavities. This can be observed on models with frank or coarse concavities (figs. 8(a), 8(e)). However, when no constriction is identified (fig. 8(d)), the object is segmented with coarse boundaries.

4. Hierarchy *semantic*: on the contrary to related work,



(a) 1^{st} level. (b) 2^{nd} level. (c) 3^{rd} level.

(d) 1^{st} level. (e) 2^{nd} level. (f) 3^{rd} level.

Figure 6. Compatible hierarchical segmentation on two horses. The models are identically segmented at each step while the second one has an additional feature (tail).

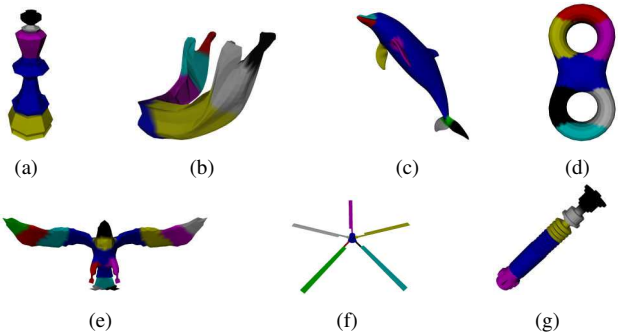
(a) Original (N_P = 12). (b) Under sampling (N_P = 13). (c) Surface noise (N_P = 13). (d) Deformations (N_P = 12).

Figure 7. Algorithm robustness against various surface degradations.

our hierarchy scheme is driven by high level shape descriptions. Consequently, it produces a more natural categorisation of features. In figure 5(c), dinopet's limbs are decomposed into arms and hands or legs and feet while those limbs are simply subdivided in the middle in [9] at the same level of hierarchy. Moreover, figure 6 shows that this hierarchy is compatible when dealing with objects that are visually similar. Horses are first decomposed into core and limbs, then limbs are subdivided and the head is decomposed at the last level of hierarchy in both cases.

5. Robustness: In fig. 7(b), the number of vertices has been reduced by 5. In fig. 7(c), each vertex has been moved randomly into a volume equal to 1% of that of the object, reproducing scanner noise for example. In fig. 7(d), the object has been deformed via user-interaction as described in [18]. After these degradations, the number of extracted patches N_P is quite stable, even with strong surface noise. Moreover, the deformed object is segmented identically to the original one, which underlines the pose robustness of the algorithm.

6. Time complexity: enhanced topological skeletons are extracted in $O(n^2)$ steps in the worst case, with n the number of vertices in T [18]. Feature boundary computation takes $O(N)$ steps with N the number of nodes in the skeleton. Feature hierarchy composition takes $O((N_J+1)\times N)$ with N_J the number of junction areas in the feature. Table 1 shows running times obtained on a 3GHz P4 PC. For information, the dinopet is segmented in 28 s. in [9] while our method takes 3.9 s.



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

(e) (f) (g)

Figure 8. Finest level of hierarchy for various objects.

Model	Faces	1^{st} level	Last level	Time (s.)
Hand	52 000	7	12	107
Baby	10 000	6	24	5.4
Chess piece	600	3	5	0.05
Human jaw	2 300	3	7	0.43
Dolphin	4 200	6	8	0.89
Bi-torus	6 000	5	7	1.3
Bird	2 000	6	14	0.4
Fan	450	6	8	0.06
Mechanical	2800	3	5	0.53
Horse 1	40 000	6	21	34
Dinopet	9 000	7	28	3.9
Horse 2	5 000	7	20	1.35

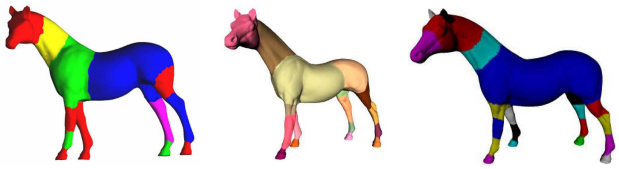
Table 1. Computation times and number of surface patches at the coarsest and the finest levels of hierarchy.

7. Control parameters: some low level parameters are used in our method (α_0, $\Delta\widehat{f_m}$ and ζ_0) but they have been fixed experimentally for all models. Therefore, our algorithm is fully automatic.

As a comparison to related work, feature points are accurately extracted, even on small features such as the horse's ears, where segments are isolated in figure 9 (black and white patches), on the contrary to [20, 13]. Also notice that limbs are more accurately subdivided with our method than in [20] (where the front right leg, in red, is not subdivided). Moreover, our hierarchical scheme decomposes features in a more semantic-oriented manner than [9] (fig. 5). Finally, running times are significantly lower than clustering based methods [10, 9]. On the contrary to [13], the genus of the surface does not impact the computation time (cf. table 1).

5.2. Limitations

Because it is topology based, our method cannot distinguish features when they form a compact connected com-



(a) Zhang [20]. (b) Lien [13]. (c) Our algorithm.

Figure 9. Comparison to other skeleton driven segmentation algorithms [20, 13].

ponent. For example, a closed fist will not be decomposed if the fingers are *stuck* to each other. This drawback can be observed in figure 3(d) where fingers are stuck at their basis, preventing the extraction of the first phalanx as an individual feature. The same effect can be observed in figure 5, where the upper part of the dinopet's legs cannot be distinguished from the core. To overcome this issue, extending extracted boundaries with an optimization algorithm could be an interesting direction for future work. Moreover, local re-meshing techniques could improve boundary locations on coarsely designed objects, such as the bi-torus (fig. 8(d)).

6. Conclusion

In this paper, we presented a fully automatic topology driven 3D mesh hierarchical segmentation algorithm. Feature boundaries and feature hierarchy are both computed in a *semantic-oriented* manner, subdividing character limbs into arms and hands for example. Method performance has been evaluated according to latest evaluation methodology [3]. Experiments shown the rapidity of our algorithm as well as its robustness against various surface degradations.

We made the following contributions. We adapted shape topological description techniques to the segmentation problem, using a curvature constrained geodesic based mapping function and identifying constrictions along resulting level lines. We also proposed a region merging strategy that produces a fine segmentation, based on the previously computed topological decomposition. Finally, we introduced a new semantic-oriented hierarchical scheme, where features are subdivided according to their topological complexity. In the future, thanks to the progressive shape understanding provided by the algorithm, we would like to address the partial shape retrieval problem [7].

Acknowledgements

This work is partially supported by the European Network of Excellence *Delos* No. 507618 – http://www.delos.info.

References

[1] INRIA Gamma research group repository: http://www-c.inria.fr/gamma/disclaimer.php.

[2] M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22:181–193, 2006.

[3] M. Attene, S. Katz, M. Mortara, G. Patané, M. Spagnuolo, and A. Tal. Mesh segmentation: A comparative study. In *Shape Modeling International*, pages 14–25, 2006.

[4] S. Berreti, A. Del Bimbo, and P. Pala. Partitioning of 3D meshes using Reeb graphs. In *IEEE ICPR*, pages 19–22, 2006.

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

[6] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[7] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin. Modeling by example. *ACM Transactions on Graphics*, 23:652–663, 2004.

[8] F. Hétroy. Constriction computation using surface curvature. In *Eurographics*, pages 1–4, 2005.

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

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

[11] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10:557–565, 1992.

[12] X. Li, T. W. Toon, and H. Z. Decomposing polygon meshes for interactive applications. In *Symposium on Interactive 3D Graphics*, pages 35–42, 2001.

[13] J.-M. Lien, J. Keyser, and N. M. Amato. Simultaneous shape decomposition and skeletonization. In *ACM Symposium on Solid and Physical Modeling*, pages 219–228, 2006.

[14] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser. A planar-reflective symmetry transform for 3D shapes. *ACM Transactions on Graphics*, 25:549–559, 2006.

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

[16] A. Shamir. Segmentation and shape extraction of 3D boundary meshes. In *Eurographics*, pages 137–149, 2006.

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

[18] J. Tierny, J.-P. Vandeborre, and M. Daoudi. 3D mesh skeleton extraction using topological and geometrical analyses. In *Pacific Graphics*, pages 85–94, 2006.

[19] H. Yamauchi, S. Gumhold, R. Zayer, and H.-P. Seidel. Mesh segmentation driven by gaussian curvature. *The Visual Computer*, 21:649–658, 2005.

[20] E. Zhang, K. Mischaikow, and G. Turk. Feature-based surface parametrization and texture mapping. *ACM Transactions on Graphics*, 24:1–27, 2005.