Proceedings of ESDA2008
Proceedings of the 9th Biennial ASME Conference on Engineering Systems Design and Analysis
July 7-9, 2008, Haifa, Israel

ESDA2008-59565

MESH SEGMENTATION FOR CAD APPLICATIONS

Ayellet Tal
Departmet of Electrical Engineering
Technion
Haifa, Israel 32000
Email: ayellet@ee.technion.ac.il

ABSTRACT

Segmentation of meshes has received a lot of attention in recent years, due to its growing number of applications. In this paper, we discuss properties that have been used in the literature to evaluate segmentation algorithms. Then, we describe some applications of segmentation in CAD. For each application, we review one of our segmentation algorithms that is suitable for the problem. We focus on four applications: modeling by example, shape-based retrieval, skeleton extraction, and paper crafting.

INTRODUCTION

Segmentation has been an important challenge in various disciplines for many years. Examples include image segmentation [25,29], volumetric mesh segmentation [13,24], point-based clustering [1,7], and boundary mesh segmentation [3,27]. This paper focuses on the latter.

Mesh segmentation is defined as follows. Given a boundary mesh $M = \{V, E, F\}$, defined by its sets of vertices V, edges E, and faces F, the goal is segment M into face-wise disjoint sub-meshes $\Sigma = \{M_1, \cdots M_k\}$. The partitioning of the faces induces a partition on the edges and the vertices.

Though mesh segmentation can benefit a variety of applications, such as computer graphics animation, shape matching, texture mapping, and more [2,15,17,18,22,23,28,31], the current paper concentrates on its applications to CAD. The recent developments in shape segmentation and analysis indicate that it may be a suitable basis for a variety of activities in CAD, including modeling, retrieval, skeleton extraction, and construction of paper-crafts.

The key consideration in mesh segmentation is the criteria according to which a mesh partitioning should be evaluated. It is evident that these criteria must depend largely on the expected use of the segmentation, as different uses require different algorithms. In this paper, we discuss some common criteria used in the literature.

Then, we review some of our recent segmentation algorithms, in light of their usages in CAD. For each application, we set the appropriate requirements and analyze the quality of the algorithm accordingly. We show that certain criteria are less essential than others, for a specific use. We also demonstrate the results of each of these algorithms.

The paper is structured as follows. In the first section, we propose several general criteria for evaluating a segmentation algorithm. In the next section, we discuss four applications of segmentation in CAD. For each application, we discuss the essential properties required and review an algorithm that is characterized by these properties. The last section concludes the paper.

SEGMENTATION PROPERTIES

A variety of ways to evaluate the quality of a segmentation exists. Below, we list criteria that have been used in the literature. In the next section, we examine these properties in light of specific applications.

1. SEGMENTATION TYPE: It is common to classify segmentation's objectives into two classes [26]: segmentation into meaningful components (i.e., following the minima

rule [11]) and segmentation into purely geometric shapes, i.e., planar or conic segments.

2. GEOMETRIC CONSTRAINTS: Sometimes, each component is restricted to certain geometric constraints, such as convexity [6].
3. TOPOLOGICAL CONSTRAINTS: Other common restrictions are topological. For instance, each component should be a single connected component or should be topologically equivalent to a disk.
4. HIERARCHY: Often, a single segmentation of a model would not exactly fit the segmentation the user "has in mind". Therefore, it is desirable to produce either a hierarchical segmentation or a multi-scaled one.
5. EXACT CUTS: Though defining "correct" boundaries between segments is infeasible, desirable geometric properties of the boundaries can be characterized. Such properties can include smoothness, length, and location along concave features.
6. INTERACTIVITY: Fully automatic segmentation is a difficult problem. Therefore, user-guided segmentation techniques may be a more practical solution for applications that require a considerable accuracy of the components.
7. POSE INVARIANCE: For certain applications it is important that models of similar objects in different poses, will be segmented compatibly. For instance, a sitting man and a standing man should have the same segments.
8. PROPORTION INVARIANCE: Similarly, it is sometimes important to compatibly segment models of similar objects, having different proportions between their parts. For example, a chair having a long back and a chair having a short one, should have the same segments.
9. EXTRACTION OF SMALL FEATURES: The extraction of very small features, like parts of the fingers of a human, is still considered difficult. The ability to achieve it is an objective for some algorithms.
10. PERFORMANCE: Both the worst-time asymptotic complexity and the actual running times of the algorithms are important factors in the choice of an algorithm.
11. CONTROL PARAMETERS: The number of control parameters gives some indication regarding the difficulty of tuning the algorithms, in order to produce the desirable segmentation. Most segmentation algorithms have parameters. The goal is to bring the number of parameters to minimum.
12. BOUND ON THE MAXIMUM AND/OR MINIMUM NUMBER OF ELEMENTS IN A COMPONENT OR THE NUMBER OF COMPONENTS: This constraint is often used too eliminate over-segmentation or too large and unbalanced components.

SEGMENTATION IN CAD

Over the years we have developed a variety of segmentation algorithms [6,8,15,31,32], each satisfying a different subset of the properties listed above. We have demonstrated their uses in metamorphosis, skeleton extraction, animation, modeling, retrieval, and paper crafting. This section focuses on CAD-related applications.

We discuss four uses: modeling by example, shape-based retrieval, skeleton extraction, and paper crafting. For each application, we discuss the essential requirements from the list above and then briefly describe the algorithm that attempts to achieve these properties.

Modeling by example [8]: We investigate modeling by example, a data driven approach to constructing new 3D models by assembling parts from previously existing ones. We have built an interactive tool that allows the user to find and extract parts from a large database of 3D models and composite them together to create new 3D models. This approach is useful for creating objects with interchangeable parts, which includes most man-made objects (vehicles, machines, furniture, etc.), as illustrated in Figure 1.

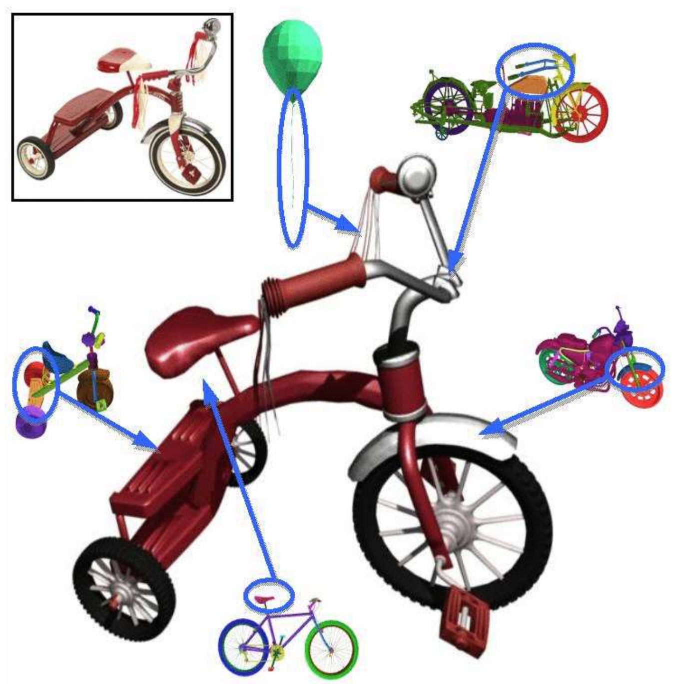


Figure 1. Modeling a Radio Flyer tricycle

From a segmentation viewpoint, the major consideration is how to provide an interactive segmentation tool that provides "exact" cuts. This is so, since only one, very exact piece is needed. There are several possible approaches to interactive segmentation of 3D meshes [10,16,33]. In our system, the user is allowed to paint strokes on the mesh surface to specify where cuts should be made. This stroke specifies a constraint that the cut must pass within r pixels of every point on the stroke. Then a cost function of a cut can be computed by:

$$cost(e) = c_{len}(e) \times c_{ang}(e)^{\alpha} \times c_{dist}(e)^{\beta} \times c_{vis}(e)^{\gamma} \times c_{dot}(e)^{\delta},$$

where c_{len} is the edge's length, c_{ang} is a function of the dihedral angle of its adjoining faces, c_{dist} depends on the distance to the stroke, c_{vis} indicates the visibility of the edge, and c_{dot} is a function of the normal. $\alpha, \beta, \gamma, \delta$ trade off between the factors. A constrained least cost path problem is solved in order to find the optimal cut specified by the user s stroke.

Shape-based retrieval [32]: Retrieval of objects, which is based on similarity between them, can also be based on segmentation. This approach attempts to succeed the theories of [4, 5, 21]. The key idea is to decompose each object into its "meaningful" components at the object's deep concavities, and to match each component to a basic shape. After determining the relations between these components, an attributed graph that represents the segmentation is constructed and considered the object's *signature*.

Given a database of signatures and one specific signature, the latter is compared to other signatures in the database, and the most similar objects are retrieved.

This graph-based approach has a few important properties. First, it is invariant to non-rigid-transformations. For instance, given a human object, we expect its signature to be similar to signatures of other humans, whether they bend, fold their legs, or point forward, as illustrated in Figure 2. In this figure, all the 19 humans in a database consisting of 388 objects, are ranked among the top 21 objects, and 17 among the top 17. Second, normalization is not required, since the signature is a graph that is invariant to rigid transformations. Third, the signature tolerates degenerated meshes and noise. This is so because the object is represented by its general structure, ignoring small features. Finally, the proposed signature is very compact. Thus, signatures can be easily stored and transfered.

For this application, neither accuracy nor hierarchy are needed. Five requirements are posed: good performance, a limited number of segments, no control parameters, pose invariance, and proportion invariance. Performance is vital since the algorithm should be executed on a whole database of objects and not on a single model. A limited number of segments is important since the segmentation graphs should be compared. Parameters should not be controlled for every object, since again, the database might be large. Obviously, pose and proportion should not influence the segmentation

Therefore, for this application two extremely simple and linear segmentation algorithms are utilized: The Watershed decomposition [20] and a BFS-based heuristic [6].

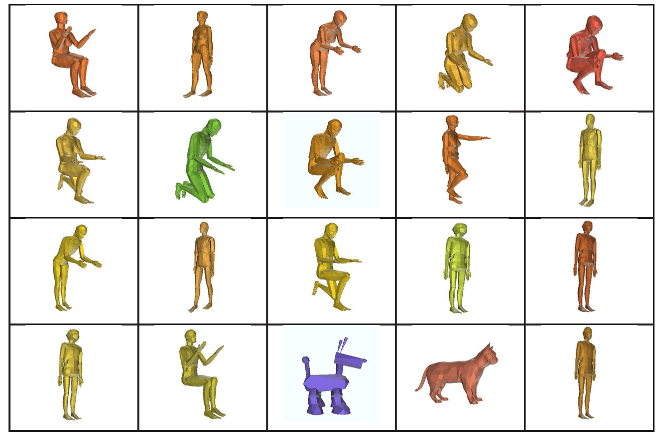


Figure 2. Retrieval of the top 20 objects similar to to the top left-most human figure

Skeleton extraction: It has been shown in [15] that given a segmentation, a control-skeleton can be extracted. This is done by traversing the segmentation tree and generating joints at the boundary between the segments. In [19], it is shown that the opposite is also correct. Skeletons can facilitate the creation of segmentations.

A segmentation algorithms for skeleton extraction should satisfy the following properties. It should segment the patches into meaningful components; it should be hierarchical; the cuts between components should be exact, since otherwise joint binding will be difficult; it should be both pose invariant and proportion invariant, since similar objects should have similar skeletons; and small features should be extracted, so as to facilitate the skeleton at these parts.

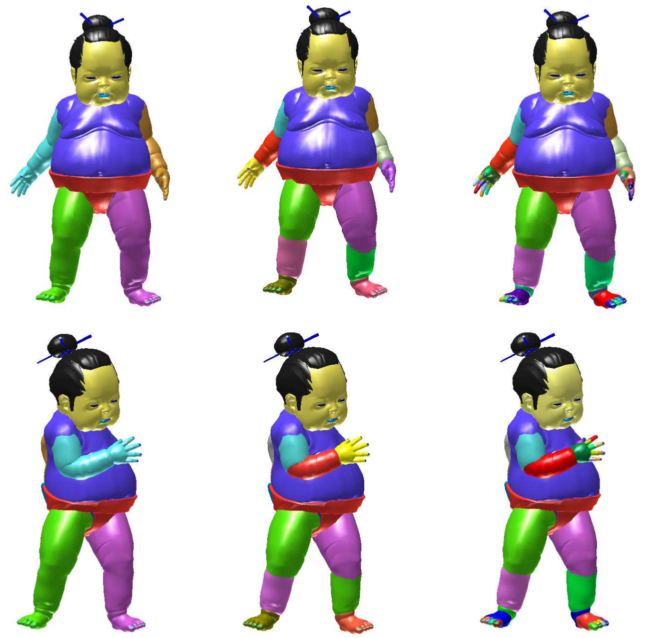
The algorithm described in [14] qualifies for this application, because it satisfies these requirements, as demonstrated in Figure 3.

The algorithm proceeds from coarse to fine. For each node in the hierarchy tree, the algorithm consists of the following stages.

1. Mesh coarsening: Mesh coarsening is applied as a preprocessing step [9]. It assists not only in accelerating the algorithm when executed on large meshes, but also in decreasing the sensitivity of the algorithm to the presence of noise.
2. Pose-insensitive representation: Multi-dimensional scaling is used to transform the mesh S into a canonical mesh S_{MDS}. Euclidean distances between points on S_{MDS} are similar to the geodesic distances between their corresponding points on S. This property makes the representation pose insensitive, because folded organs (e.g., arms) are "straightened" up by the transformation.
3. Feature point detection: A few points, the *prominent feature points*, are computed on S_{MDS} and mapped back to their

prevent situations in which objects without prominent components (i.e., almost convex objects), get further segmented.

Paper craft [30]: The aim of the algorithm is to segment a mesh into a small number of segments that can be well approximated by developable surfaces and whose boundaries can be easily cut and glued. The results are demonstrated in Figure 4.



(a) First level (b) Third level (c) Sixth level

Figure 3. Pose-insensitive segmentations: Two sumo wrestlers in different poses are segmented separately (top and bottom). The segmentations are similar in all levels of the hierarchy. Note also the extraction of the small features (The wrestler's hair, facial features, nails and mawashi (belt) originally belong to different connected components.)

Figure 4. Paper-craft models

corresponding points on S. Intuitively, points on the tips of components, such as the tail, the legs and the head of an animal, are prominent feature points. The algorithm is based on the observation that feature points can be characterized by local as well as global conditions, in terms of their geodesic distances.

4. Core component extraction: The core component is extracted using a *spherical flipping* operation.
5. Mesh segmentation: The algorithm computes the other segments, each containing at least one feature point.
6. Cut refinement: The boundaries between the segments, which were found in the previous stage, are refined. The goal is to find boundaries that go along the "natural" seams of the mesh.
7. Mesh refinement: After the segmentation of the coarse-resolution mesh (Step 1) is computed, it is mapped to the input, fine-resolution mesh, and the cut is refined again, similarly to Step 6.

The hierarchical segmentation continues as long as the current segment S_i contains feature points and the ratio between the number of vertices contained in the convex hulls of both S_i and $S_{i_{MDS}}$ and the total number of vertices is low. These conditions

In this case, the following requirements are relevant: The segmentation type should be purely geometrical; the boundaries between segments should be piecewise smooth, and thus easy to cut; and the number of segments should be controlled by a parameter, either indicating the number of pieces or indicating the allowed error.

A surface is developable if it has a zero Gaussian curvature at all points. Since this definition does not provide a practical algorithm for generating a segmentation, our algorithm uses two types of surfaces known to be developable: a planar surface and a conic surface [12]. Our general scheme, however, can incorporate other pre-defined types of developable surfaces.

The algorithm begins with an initial over-segmentation of the mesh into trivial developable segments. This initial segmentation is iteratively modified, by decreasing the number of segments, while increasing the error. Each such iteration approximates the current segments, by fitting each segment to a conic(/plane), using weights specific to our problem. Once the segmentation is determined, the approximations are modified, in order to accommodate for “good” boundaries. Then, the analytical boundaries between the approximations are computed, therefore not restricting the boundaries to pass through edges of the original mesh.

CONCLUSION

In this paper we have analyzed the necessary requirements that segmentation algorithms should satisfy, for specific problems in CAD. We have shown that every application requires different properties. For instance, in retrieval, a very simple algorithm suffices, even though it cannot produce exact segments, while for skeleton extraction, precises segments are vital. As another example, while hierarchical segmentation should be attained for skeleton extraction, a flat one suits paper crafting, and extraction of a single component is suitable for modeling by example. The conclusion is that before choosing a segmentation algorithm, one should characterize carefully the properties required for the specific application.

ACKNOWLEDGMENT

This work was partially supported by the European FP6 NoE grant 506766 (AIM@SHAPE), by the Israeli Ministry of Science, Culture & Sports, grant 3-3421, and by the Fund for Promotion of Research at the Technion,.

REFERENCES

[1] P. Arabie, L.J. Hubert, and G. De Soete. *Clustering and Classification*. World Scientific Publishing Company, NJ, 1996.

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

[3] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A.Tal. Mesh segmentation - a comparative study. In *Shape Modeling International (SMI)*, 2006.

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

[5] I. Biederman. Aspects and extensions of a theory of human image understanding. *Pylyshyn Z. editor, Computational Processes in Human Vision: An Interdisciplinary Perspective*, pages 370–428, 1988.

[6] B. Chazelle, D.P. Dobkin, N. Shourhura, and A. Tal. Strategies for polyhedral surface decomposition: An experimental study. *Computational Geometry: Theory and Applications*, 7(4-5):327–342, 1997.

[7] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2000.

[8] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D.P. Dobkin. Modeling by example. *ACM Trans. Graph. (SIGGRAPH)*, 23(3):652–663, 2004.

[9] M. Garland and P.S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of SIGGRAPH 1997*, pages 209–216, 1997.

[10] A. Gregory, A. State, M.C. Lin, D. Manocha, and M.A. Livingston. Interactive surface decomposition for polyhedral morphing. *The Visual Computer*, 15:453–470, 1999.

[11] D.D. Hoffman and W.A. Richards. Parts of recognition. In S. Pinker, editor, *Visual Cognition*, pages 65–96. MIT Press, London, 1985.

[12] D. Julius, V. Kraevoy, and A. Sheffer. D-charts: Quasi-developable mesh segmentation. In *Computer Graphics Forum*, volume 24, pages 581–590, 2005.

[13] G. Karypis and V. Kumar. Multilevel algorithms for generating coarse grids for multigrid methods. In *Supercomputing*, 1998.

[14] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):865–875, 2005.

[15] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph. (SIGGRAPH)*, 22(3):954–961, 2003.

[16] Y. Lee, S. Lee, A. Shamir, D. Cohen-Or, and H-P. Seidel. Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design*, 2005.

[17] B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. In *Proceedings of SIGGRAPH 2002*, pages 362–371. ACM SIGGRAPH, 2002.

[18] X. Li, T.W. Toon, T.S. Tan, and Z. Huang. Decomposing polygon meshes for interactive applications. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 35–42, 2001.

[19] 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.

[20] A.P. Mangan and R.T. Whitaker. Partitioning 3D surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):308–321, 1999.

[21] D. Marr. *Vision - A computational investigation into the human representation and processing of visual information*. W.H. Freeman, San Francisco, 1982.

[22] M. Mortara, G. Patanè, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Blowing bubbles for the multi-scale analysis and decomposition of triangle meshes. *Algorithmica, Special Issues on Shape Algorithms*, 38(2):227–248, 2004.

[23] M. Mortara, G. Patanè, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Plumber: A multi-scale decomposition of 3d

shapes into tubular primitives and bodies. *Proc. of Solid Modeling and Applications*, pages 139–158, 2004.

[24] I. Moulitsas and G. Karypis. Multilevel algorithms for generating coarse grids for multigrid methods. In *Supercomputing*, pages 45–45, 2001.

[25] N.R. Pal and S.K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.

[26] A Shamir. A formalization of boundary mesh segmentation. In *Proceedings of the second International Symposium on 3DPVT*, 2004.

[27] A. Shamir. Segmentation and shape extraction of 3d boundary meshes. In *Eurographics, State-of-the-Art Report*, 2006.

[28] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonization using the shape diameter function. *The Visual Computer*, page to appear, 2008.

[29] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):810–813, June 2006.

[30] I. Shatz, A. Tal, and G. Leifman. Paper craft models from meshes. *The Visual Computer*, 22(9):825–834, 2006.

[31] S. Shlafman, A. Tal, and S. Katz. Metamorphosis of polyhedral surfaces using decomposition. *Eurographics*, pages 219–228, September 2002.

[32] A. Tal and E. Zuckerberger. Mesh retrieval by components. pages 142–149, 2006.

[33] K.C.-H Wong, Y-H.S SIU, and H. Sun P.-A. Heng. Interactive volume cutting. In *Graphics Interface Forum*, 1998.