3D MESH PARTITIONING FOR RETRIEVAL BY PARTS APPLICATIONS

G.Antini, S.Berretti, A.Del Bimbo and P.Pala

Dipartimento di Sistemi e Informatica
Università degli Studi di Firenze
50139 Firenze, Italy

ABSTRACT

A solution for part segmentation of 3D objects is proposed in this paper. The approach is targeted to identify salient visual parts of a mesh by determining its main protrusions and discarding, at the same time, parts originated by un-relevant local properties. This is obtained by first breaking the 3D mesh into seed regions according to the sum of geodesic distances between vertices, then by using topological and curvature information to refine the number of regions and their boundaries. In so doing, effective segmentation is regarded as a prerequisite to enable retrieval of 3D objects based on similarity of parts. Experimental results show the applicability of the proposed solution to complex shapes and its effectiveness in the identification of object parts.

1. INTRODUCTION

Multimedia applications are becoming more demanding in terms of the type of data to be handled in an effective, efficient and natural way. In particular, the search for new and realistic ways to experience and manipulate data, has attracted a lot of interest towards 3D data. This has been favored also by the large number of devices able to capture and store the 3D nature of the real world. As a consequence, the task of modeling 3D data is assuming an increasing relevance in many application fields.

Usually, 3D objects are represented as collections of polygons and vertices (*mesh*) that approximate a 3D surface. Though useful for visualization purposes, this low-level representation is often inadequate for more complex tasks, such as scene understanding, object recognition or retrieval by content. As a consequence, higher level representations are derived following different approaches. In many solutions, objects are represented by descriptors capturing global properties of objects surface, such as moments, distributions of vertex distances, surface curvature or angles between faces, or the surface properties in transformed domains, like the wavelets and the spherical harmonics. Their global nature makes these approaches useful for condensed representations of a mesh, but usually these methods cannot effectively capture local properties of a mesh.

More powerful representations can be derived by extracting local information of the object surface. However, this has the main difficulty in identifying and segmenting objects into significant visual parts retaining, at the same time, information on the connectivity and spatial relationships between individual parts. In fact, 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 [2]. Unfortunately, the problem of mesh segmentation is not well posed, in the sense that solutions are often application specific, depending on the particular optimization criterion and segmentation objective. As a consequence, previous works have followed several different solutions in order to develop effective and efficient approaches for mesh segmentation. In particular, it is possible to distinguish between solutions for *patch* and solutions for *part* segmentation. The former create patches which obey certain geometric properties, such as planarity, size or convexity, while the latter are more targeted at partitioning an object into meaningful components.

In [8] 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 a sharp convexity and vanish at a sharp concavity is used. In this way, object part boundaries are detected by locating surface points exhibiting minima of the local charge density. This theory leads to meaningful segmentations, but authors list some necessary assumptions, such as genus-one topology, no dents on the surface of the object, and the presence of at least one part boundary. In [5], a watershed decomposition approach to mesh segmentation is proposed. This generalizes morphological watersheds originally proposed for image segmentation to the case of 3D surfaces. 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. Improvements on the watershed approach are reported in [6]. An approach based on skeletonization is proposed in [4]. In this solution, first an approximation of the skeleton of a mesh is extracted, then critical points 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 to different parts. Interesting results are obtained with this algorithm. However, smoothing is also associated with potential loss of information about mesh details. More recently, a hierarchical mesh decomposition algorithm has been proposed in [3]. 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 done by defining pairwise 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 face are clustered into patches.

In this paper, we propose an original object partitioning ap-

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).

proach based on the visual salience of individual parts. In particular, object decomposition is obtained by a two steps approach. First, the construction of a Reeb-graph guides the vertex partitioning into different regions by using distances between vertices. This is obtained by computing, for every vertex of the mesh, the sum of geodesic distances between the vertex and any other vertex of the mesh. Dividing the range of the sum of geodesic distances in a set of levels, vertices are partitioned in a number of seed regions, each one corresponding to an equivalence class between vertices. A graph model is then constructed on these regions that accounts for their boundary, curvature properties and spatial adjacency. In the second step, the graph is processed in order to merge or grow initial regions according to adjacency, curvature and boundary information. This allows the regions to modify their boundaries and to be extended up to coincide with homogeneous parts of the mesh. Experiments have shown that this approach is able to reduce possible artifacts originated by local curvature changes. In particular, we observed that, in most of the cases, parts identified at the end of the segmentation process have a direct correspondence with the main protrusions of a 3D model. This is particularly useful in order to enable retrieval of 3D objects based on similarity of parts.

The remainder of the paper is organized in two Sections and a Conclusion. In Sect.2, an approach to decompose a 3D model in its constitutive salient parts is proposed. In particular, we first show the identification of the seed regions and the construction of their graph, then we describe the graph reduction process which determines the final object decomposition. The effectiveness of this segmentation approach is experimented in Sect.3 through a set of preliminary tests on complex shapes. Finally, conclusions and future research directions are drawn in Sect.4.

2. MESH PARTITIONING BY REEB-GRAPH

The proposed model develops on the assumption that distinguishing perceptual features of a 3D object are related to its main protrusions. According to this, topological information is used to identify protrusions by computing Reeb graphs, which have been successfully employed to describe the topological structure of 3D objects [1].

Given a 3D model M, a continuous real function $f : M \rightarrow \Re$ can be defined on M. The Reeb graph is the quotient space of f in $M \times \Re$ formed by the equivalence relation $(v_1, f(v_1)) \sim (v_2, f(v_2))$. This establishes that, given two points v_1 and v_2 on the model surface, the equivalence relation holds if and only if the two following conditions are fulfilled:

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

Moreover, points of M where the gradient of f vanishes (the *critical points* of f) correspond to topological changes of the Reeb graph of M.

Practically, the model M is approximated by a polygonal mesh P, and the function f is defined on the vertices of the mesh. As a consequence, a discrete number of values of f are computed. According to this, Reeb graph partitions the surface of the model into connected components of vertices with the same value of f. Since subsets of vertices are induced by the function f, different functions induce different partitioning. Examples of different choices for the function f range from the simple height function of the model vertexes in the 3D reference system (used in terrain modeling), to the distance measured from each vertex of the model to other reference points of the model, such as the model barycenter, curvature extrema or source points [1]. The choice of f can also impact on the resulting properties of the graph, like its robustness to model rotation, noise or slight irregularities of the model mesh.

To overcome such difficulties, in the following we define f based on the average geodesic distance between vertices computed on the model surface. In particular, for every vertex v of the model, the sum of the geodesic distances between v and any other vertex x of the mesh is computed as:

$$f(v) = \sum_{x \in P} \mu(v, x) \tag{1}$$

where $\mu(v, x) \geq 0$ returns the geodesic distance between any two vertices v and x on the surface of the mesh P. Values of $f(v)$ are then normalized in [0,1]:

$$f_n(v) = \begin{cases} \frac{f(v) - f_{min}}{f_{max} - f_{min}} & if \ \ f_{max} \neq f_{min} \\ \\ 0 & otherwise \end{cases} \tag{2}$$

being $f_{min} = \min_{v \in P}\{f(v)\}$, and $f_{max} = \max_{v \in P}\{f(v)\}$ the minimum and maximum sum of geodesic distances, respectively.

Due to the polygonal approximation of the model M, geodesic distance is computed as the shortest path on the edges of the mesh by using the Dijkstra's algorithm. Given a vertex v this algorithm permits the identification of the path with minimum length which connects this vertex with any other vertex on the mesh, by connecting them through the edges of the mesh. In the case of regular, or sufficiently regular meshes, this provides an acceptable approximation of the real distance computed on a continuous model.

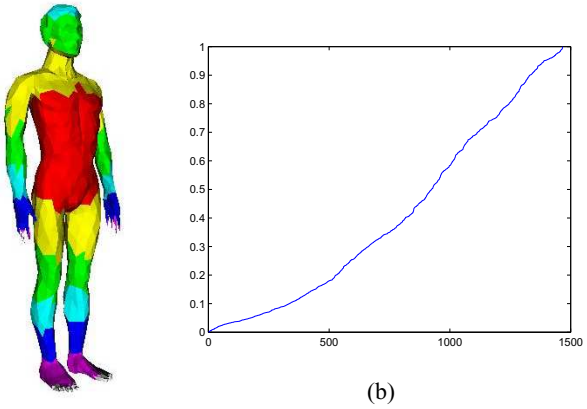
It is worth noting that the computation of geodesic distances on the mesh can be affected by the regularity of the mesh. In fact, since the Djikstra's algorithm approximates actual geodesic distances through edge lengths, a non-triangular mesh (i.e. a mesh of general polygons) or a non-regular mesh (i.e. a mesh composed of triangles of different sizes) make the estimate less accurate. So, optimization of the algorithm effectiveness requires a pre-processing step for triangularization and regularization of the mesh. Anyway, in principle, the following discussion is still valid for a generic mesh.

2.1. Part Identification

After the function $f_n(v)$ is computed for every vertex of the mesh, the range of f_n values is divided into N intervals (*levels*) of equal size $I_k = [k, (k+1)/N), k = 0, \ldots N-1$. In this way, each vertex maps to one and only one of these levels, thus determining a labeling of mesh vertices. Sets of vertices with the same label are not necessarily connected, in that vertices located in several different parts of the surface can share the same level. 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_n fall in the same interval I_k. These regions R_i are the equivalence classes defined by the Reeb graph. According to this, the Reeb graph of the model is finally derived by assigning nodes to regions, while edges of the graph connect nodes corresponding to adjacent regions. In addition, the average value of the *mean curvature* within

each region, and the lists of its boundary vertices are used as node attributes. Following [7], given a vertex $v \in P$, and its principal curvature indicated with $k_1(v)$ and $k_2(v)$, the mean curvature is computed as $c(v) = (k_1(v) + k_2(v))/2$.

Fig.1(a) highlights with different colors, vertices of the mesh of a human body which belong to different intervals I_k, of the f_n values. It can be seen that vertices at the same level can be split into several connected regions. In Fig.1(b) the distribution of f_n values for the model in Fig.1(a) is shown.



(a) (b)

Fig. 1. (a) Different colors identify vertices at seven different levels; (b) Distribution of f_n values for the model in (a).

At this stage, the number and extent of regions on the mesh depend on the number of intervals I_k which partition the values of f_n. This number can be fixed or chosen following some heuristics, which tries to optimize the initial number of levels according to the distribution of the sum of geodesic distances. In the proposed approach, we considered a fixed number of levels equal to $N = 7$. With this solution, a mesh is typically over-segmented in a set of regions which might not have a direct correspondence with the perceptually relevant parts of an object. However, experimentally, we observed that with the graph-reduction process, which edits the graph structure, regions (i.e. graph node) can be merged, or their boundaries can be reshaped to better fit significant parts of the mesh.

The identification of model parts starts from the observation that nodes of the Reeb-graph are not all topologically equivalent. In particular, each node v can be associated with a *degree* (d_v), defined as the number of edges which connect v to its adjacent nodes. In some way, this number measures the topological relevance of a node. In fact, a node with degree greater than two corresponds to a singularity in the function f_n. Visually, a singularity is equivalent to the split of a region into disjoint connected components occurring as result of passing from one level of f_n to another.

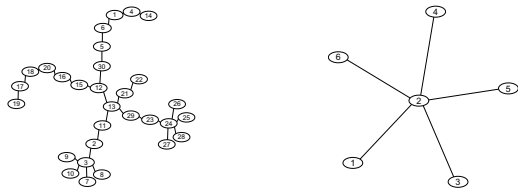
However, the discrete nature of levels I_k poses two problems in the decomposition reached at this stage: *(i)* the mesh can be over-segmented, and vertices and faces perceived as belonging to a single part can be separated into several regions; *(ii)* boundaries between regions can be not accurate. These problems have been solved with a graph-reduction process which takes into account the graph topology, and boundaries and curvature attributes of graph nodes.

In order to reduce the number of nodes retaining, at the same time, the structural information, we allow adjacent nodes to join together based on topological and curvature information. According to this, the graph-reduction process is as follows:

1. Repeat until a node i, with $d_i = 1$ exists: If the only adjacent node j has $d_j <= 2$ and $sign(c_i) = sign(c_j)$, merge the two nodes together; otherwise go back to 1.
2. Repeat until a node i, with $d_i = 2$ exists: If an adjacent node j has $d_j <= 2$ and $sign(c_i) = sign(c_j)$, merge the two nodes together; otherwise go back to 2.
3. Repeat until a node i, with $d_i \geq 2$ exists: If an adjacent node j has $d_j \geq 2$ and $sign(c_i) = sign(c_j)$, merge the two nodes together; otherwise go back to 3.

In practice, the procedure aims to retain: *i)* nodes with degree ≥ 3; *ii)* nodes adjacent to them; *iii)* nodes v_i such that $adj(v_i) < 0$, being $adj(v_i) = 1$ if the sign of the average curvature of nodes adjacent to v_i is the same of v_i, $adj(v_i) = -1$ otherwise. In fact, in step 1, regions which belong to the same protrusion can be recombined together moving from peripheral seed nodes (*source* nodes, with degree = 1) to center nodes of the object (*sink* nodes); steps 2 and 3 operate in a similar manner for nodes of higher degree, allowing them to be combined together.

Fig.2 shows the effect of the reduction process for the graph of the human body model of Fig.1. In particular, in (a) the initial Reeb graph originated by assigning a node to each region is reported. Starting from this graph, reduction operations are able to decrement the number of nodes from 30 to 6 (in (b)), identifying the main components of the model (compare Fig.1(a) with Fig.3(a)).



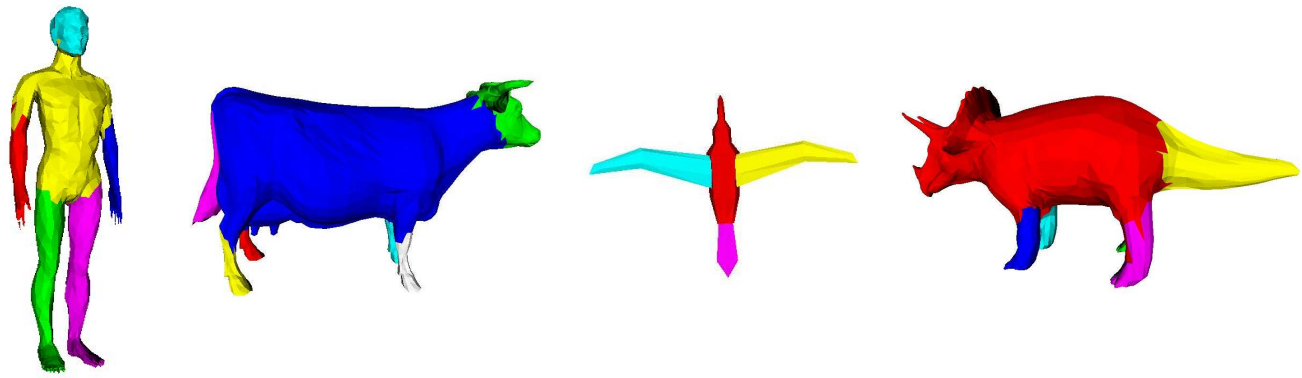
(a) (b)

Fig. 2. Graphs for the human body model of Fig.1: (a) Original Reeb-graph; (b) Final graph after the graph-reduction process.

The identification of object parts is completed by refining boundaries between regions. This is made starting from the boundary vertices of a region and evaluating the possibility to extend them towards boundary vertices of adjacent regions. In particular, a vertex v_i^R is considered a boundary vertex for the region R, if it is connected by at least one edge on the mesh with a vertex v_j^S of an adjacent region S. According to this, boundary of source regions can expand towards sink regions by using the morphological operation of *dilation* with step 1. In so doing, dilation stops when curvature changes sign passing from boundary vertices of two adjacent regions.

3. EXPERIMENTAL RESULTS

In order to test the effectiveness of the proposed solution, the segmentation algorithm has been applied to various 3D models. So far, experiments have been conducted on roughs models, in the



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

Fig. 3. Partitioning results for some test objects.

sense that models do not undergo to any process of mesh triangularization or regularization. We discussed as this can negatively impact on the segmentation algorithm, but at this stage of the research we are interested in the evaluation of robustness of the algorithm also with respect to this type of irregularities. This has also a practical relevance, in that in many real contexts of application there is not a direct control on the quality of models and a preprocessing step can be either not possible or convenient.

Due to space limitations, we report here results only for some reference models (see Fig.3). The final partitioning of the human body previously reported in Fig.1(a) is shown in Fig.3(a). The body is effectively segmented into six components which respectively identify the four arts, the main structure of the body and the head. These capture the main object protrusions and seem to provide the more plausible decomposition of the model for an application aiming to retrieve similar human bodies based on their partial similarity. Models in Fig.3(b) and (d), are quite similar in their topological structure, and are basically decomposed in the same way: the legs are separated into four individual components, while the body, the tail and the head are the other parts of the caw model; body and head are not separated in the dinosaur model. Segmentation for the bird model of Fig.3(c), correctly identifies body, wings and tail.

Finally, in Fig.4, segmentation of an objects with genus one topology is reported. In particular, the toroid mesh is an easy to segment shape according to its curvature. Actually, topological information based on the computation of geodesic distances on a polygonal mesh initially produces a cycle in the Reeb graph (as shown in Fig.4(b)), that can be removed by applying the subsequent reduction operations to merge adjacent nodes.

(a) (b)

Fig. 4. (a) Partitioning for a genus one object; (b) its Reeb graph.

4. DISCUSSION AND FUTURE WORK

In this paper, we have presented and discussed the theoretical applicability of a 3D mesh segmentation method, which combines topological and curvature information. Preliminary segmentation results show the plausibility of the proposed solution.

Future work will address a multiresolution extension of the segmentation algorithm and investigate its use as a first step in the construction of a structural representation for 3D models, based on their constitutive parts. The ultimate goal of this representation will be to support retrieval of 3D objects based on their visual appearance and similarity matching of individual parts.

5. REFERENCES

[1] M. Hilaga, Y. Shinagawa, T. Kohmura, T. L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," In *Proc. of SIGGRAPH*, pp.203-212, 2001.

[2] D. Hoffman, W. Richards, "Parts of Recognition," *Cognition*, vol.18, pp.65-96, 1984.

[3] S. Katz, A. Tal, "Hierarchical Mesh Decomposition Using Fuzzy Clustering and Cuts," *ACM Trans. on Graphics*, vol.22, no.3, pp.954-961, 2003.

[4] X. Li, T.W. Woon, T.S. Tan, Z. Huang, "Decomposing Polygon Meshes for Interactive Applications", *ACM Symposium on Interactive 3D Graphics*, pp.35-42, 2001.

[5] A.P. Mangan, R.T. Whitaker, "Partitioning 3D Surface Meshes Using Watershed Segmentation," *IEEE Trans. on Visualization and Computer Graphics*, vol.5, no.4, pp.308-321, 1999.

[6] D.L. Page, A.F. Koschan, M.A. Abidi, "Perception-based 3D Triangle Mesh Segmentation Using Fast Marching Watershed," In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp.27-32, 2003.

[7] G. Taubin, "Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation," In *Proc. Int. Conf. on Computer Vision*, pp.902-907, 1995.

[8] K. Wu, M.D. Levine, "3D Part Segmentation Using Simulated Electrical Charge Distributions", *IEEE Trans. on Pattern Analysis and Machine Intelligence* vol.19, no.11, pp.1223-1235, 1997.