Segmentation and Surface Characterization of Arbitrary 3D Meshes for Object Reconstruction and Recognition

Georgios Papaioannou   Evaggelia-Aggeliki Karabassi   Theoharis Theoharis

Department of Informatics, University of Athens, GREECE

georgep@di.uoa.gr   *aggeliki@di.uoa.gr*   *theotheo@di.uoa.gr*

Abstract

Polygonal models are the most common representation of structured 3D data in computer graphics, pattern recognition and machine vision. The method presented here automatically identifies and labels all compact surface regions of a polygonal mesh, visible or not, and extracts valuable invariant features regarding their geometric attributes. A method that is independent of the mesh topology is also presented for the surface bumpiness estimation and the identification of coarse surface regions.

1. Introduction

Recognition and classification of three-dimensional objects is a research area with increasing interest and applicability. The aim of a large category of recognition techniques, often adopted in computer vision problems, is the extraction of information associated with the object structure. As a rule, these methods operate on surfaces, and involve representations of the original object that allow efficient structural analysis. The structural information is obtained by partitioning the object in surface patches (varying from single polygons to entire object regions) and incorporating the patch connectivity information in the object representation, using region adjacency graphs [1]. Different partitioning criteria and representations lead to different techniques. In this area, significant work has been conducted on variations of the Gaussian Image representation [2], [3], [4].

In this work, we present a region growing mesh segmentation strategy, which partitions the object into distinct compact surfaces of similar orientation, corresponding to the 'sides' or 'facets' of the object. The result is a unique object representation as a graph of surface regions, which fully describes the object topology. Our method calculates a set of surface parameters that can be used for the characterization of each facet of the three-dimensional object, including surface bumpiness. All extracted parameters are invariant to object translations, rotations and uniform scaling.

The proposed method was originally intended for the identification of cracked or broken surfaces of 3D-scanned polygonal meshes, as part of a system for the reconstruction of objects from broken parts [6].

The surface segmentation algorithm requires no special polygonal mesh representation apart from the raw triangles themselves. As a preprocessing step, a list of the adjacent faces of each polygon must be constructed. However, if this information can be extracted from the mesh representation, as is the case in the popular edge-list and winged-edge representations, this step can be significantly accelerated or entirely omitted. The (planar) polygons included in the object mesh may have an arbitrary number of edges and an arbitrary number of polygons may share the same edge.

2. Formal conventions

In the following text, $p_{ij} \in \Re^3$ is the j-th vertex of the i-th polygon P_i of a mesh. $\vec{N}_i$ is the unitary length normal vector of P_i. The operand $i \oplus j$, i=1…n denotes a modulo n addition: $i \oplus j$ = (i+j) mod n. Finally, given a vector $\vec{F} \in \Re^3$, $\|\vec{F}\| = \sqrt{\vec{F} \cdot \vec{F}}$ is the length of $\vec{F}$.

3. Surface segmentation

A surface region, $\mathbf{R}_k$, k=1,…, N_{reg} represents a facet of the polygonal mesh and consists of a collection of connected polygons of similar orientation. The orientation similarity criterion for the inclusion of a polygon in a region is based on the deviation of the polygon's normal vector $\vec{N}_i$ from the average normal vector $\vec{N}_{ave}(\mathbf{R}_k)$ of the region $\mathbf{R}_k$:

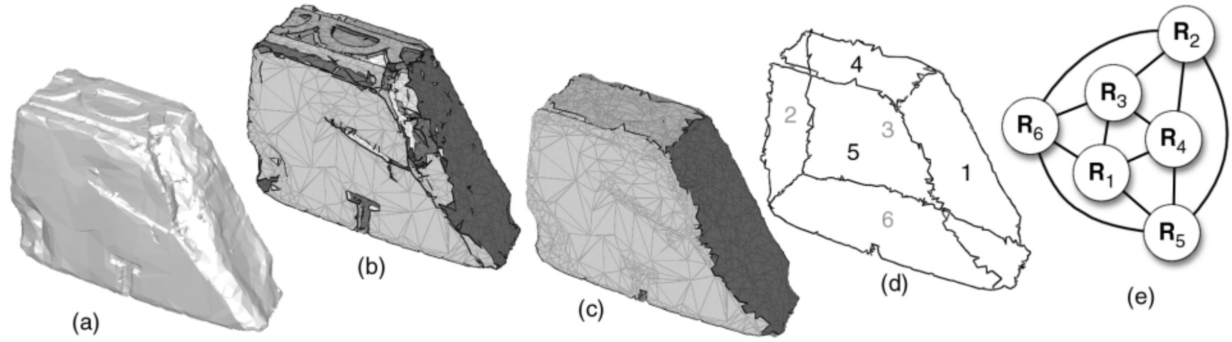

Figure 1. The mesh segmentation. (a) Original object. (b) Surface regions after region growing. (c) Surface regions after clean-up (d) Region boundaries. (e) The regional graph of the object.

$$\vec{N}_{ave}(\mathbf{R}_k) = \left\| \sum_{j|P_j \in \mathbf{R}_k} A_j \vec{N}_j \right\|^{-1} \cdot \sum_{j|P_j \in \mathbf{R}_k} A_j \vec{N}_j$$

where $A_j = \frac{1}{2} \left| \vec{N}_j \cdot \left\{ \sum_q p_{jq} \times p_{jq \oplus 1} \right\} \right|$ is the area of P_j.

If $\vec{N}_i \cdot \vec{N}_{ave}(\mathbf{R}_k) \geq e_N$ then $P_i \in \mathbf{R}_k$.

$e_N \in [-1,1]$ is the *facet segmentation threshold* and can be defined with respect to the maximum curvature $\theta_{\max}$ of a mesh facet as: $e_N = \cos\theta_{\max}$. The maximum curvature is, in general, application dependent and defines the maximum allowed angle between the normals of two polygons in the same region. Typical values for e_N are in the range [0.7,0.9].

Surface segmentation is accomplished, using a simple region-growing algorithm (Figure 1). The process begins with an arbitrary polygon. Neighboring polygons are classified to the same region if their normals satisfy the above mentioned criterion, otherwise a new region is formed.

During the region growing process, small surface regions may be created within larger ones due to a very high facet segmentation threshold or a bumpy surface (Figure 1b). As it is desirable to fragment the object mesh into crude facets, which are the ones that define the overall shape of an object, a clean-up stage is required to eliminate small erroneous regions (Figure 1c).

The elimination of the insignificant regions is achieved by iteratively assigning the polygons of regions $\mathbf{R}_k$ whose area $A_{\mathbf{R}_k}$ is smaller than an $a_{\lim}$ fraction of the entire surface area A_{mesh} of the polygonal model to adjacent regions of area larger than $a_{\lim} \cdot A_{mesh}$. Typical values for $a_{\lim}$ are 2% to 5%.

4. Region based object features

After the surface segmentation process, a number of quantitative attributes of the mesh regions is available. The simplest attribute of a region is the calculated area $A_{\mathbf{R}_k}$ of a region $\mathbf{R}_k$. In order for the region area to be invariant to uniform scaling, a normalized version should be used: $\overline{A}_{\mathbf{R}_k} = A_{\mathbf{R}_k} / A_{mesh}$. Let us define the *significant regions* subset $\mathbf{S}_{reg}$ of all regions $\{\mathbf{R}_1,...\mathbf{R}_{N_{reg}}\}$ as $\mathbf{S}_{reg} = \{\mathbf{R}_k \in \{R_1...R_{N_{reg}}\} : \overline{A}_{\mathbf{R}_k} > a_{\lim}\}$. The number of significant object regions $N_{S_{reg}}$ corresponds to the number of crude object faces.

The calculated normals of the significant regions $\vec{N}_{ave}(\mathbf{R}_k)$, $R_k \in \mathbf{S}_{reg}$, combined with the normalized area of the regions $\overline{A}_{\mathbf{R}_k}$ and the connectivity information among the regions in $\mathbf{S}_{reg}$ can result in a very compact graph representation of the object (Figure 1e). This representation is independent of the mesh complexity and mesh orientation, provided that region normals are considered relative to each other.

The partitioning of the object into a graph of surface regions has the advantage that remote object faces of similar orientation are not classified in the same region, as is the case in the Gaussian Sphere (GS) [2], the Extended Gaussian Image (EGI) [3] and the Distributed Extended Gaussian Image (DEGI) [5]. This advantage is also encountered in the Spherical Attribute Image representation (SAI) [4].

Apart from the region features already calculated during the segmentation, other features can be measured

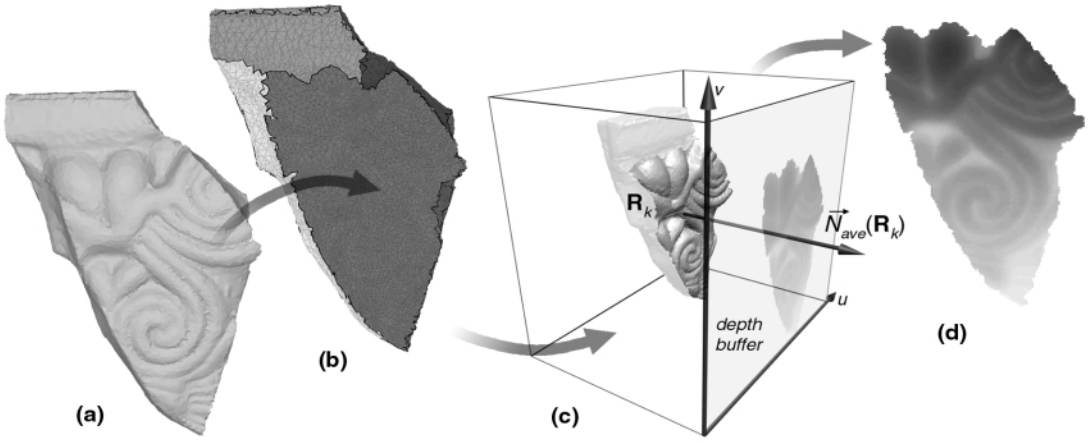

Figure 2. Bumpiness measurement: the construction of the elevation map. (a) The original object. (b) The segmented surface. (c) Depth buffer calculation. (d) the region elevation map.

such as the local mean curvature and the average normal deviation: $dev_N(\mathbf{R}_k) = \arccos(\sum_{i|P_i \in \mathbf{R}_k} \vec{N}_{ave}(\mathbf{R}_k) \cdot \vec{N}_i)$.

Such mesh-based features cannot be used for the accurate description of overall qualitative features of a surface region such as the bumpiness, due to the non-uniform sampling of the original polygonal mesh. An approximation of the polygonal object by a uniform mesh, as in the SAI representation, would prove impractical in this case because it would require a very large number of tessellated patches and would cause an inevitable smoothing of the resulting mesh.

5. The detection of bumpy surfaces

Bumpiness as a surface quality refers to the local variations of the surface elevation or equivalently, the amount of local perturbation of the surface normal vector. Depending on the spatial resolution of these variations, a surface can be flat, smooth or coarse. An engraved surface is usually a smooth surface while a cracked or generally noisy surface is a coarse one.

The method presented, uses conventional two-dimensional texture analysis to extract information about the surface elevation. The distance of a region surface $\mathbf{R}_k$ from a plane perpendicular to $\vec{N}_{ave}(\mathbf{R}_k)$ is measured at equidistant locations (*u*, *v*) on the plane to obtain a uniformly sampled elevation map (Figure 2). The elevation map is then used as an image upon which a textural coarseness measure is estimated [7]. A large variety of statistical, spectral and structural methods for the texture classification can be found in the image processing bibliography.

In our application, where cracked or patterned surfaces must be identified, a simple and effective bumpiness measure $B_{\mathbf{R}_k}$ can be derived from the second-order partial derivatives of the elevation map of region $\mathbf{R}_k$ with regard to the map parameters (*u*, *v*):

$$B_{\mathbf{R}_k} = \frac{1}{N_{Depth}} \sum_{\substack{(u,v)| \\ Depth_{\mathbf{R}k}(u,v) \neq \infty}} \left| \nabla^2 Depth_{\mathbf{R}_k}(u,v) \right|$$

where $Depth_{\mathbf{R}_k}(u,v)$ is the elevation map of region $\mathbf{R}_k$ and N_{depth} is the number of non-infinite values in $Depth_{\mathbf{R}_k}(u,v)$. The above relation means that $B_{\mathbf{R}_k}$ reflects the average steep transitions per surface region.

The elevation map is a discrete image of resolution $M_{Depth} \times M_{Depth}$ and therefore $\nabla^2 Depth_{\mathbf{R}_k}(u,v)$ is the common Laplacian image operator.

An elevation map can be easily obtained from polygonal meshes by rendering (scan-converting) [8] the polygons of the region $\mathbf{R}_k$ after having aligned $\vec{N}_{ave}(\mathbf{R}_k)$ with the direction of the viewer (Figure 2c). In the majority of polygon rendering algorithms used in computer graphics, a special buffer, called the *depth buffer* or *Z-buffer* [8], is maintained for hidden surface removal. At each location (u, v) of the depth buffer, the depth value closer to the viewer of the (u, v) pixel is stored. A pixel at the location (u, v) is drawn in the image buffer after its depth component indicates that the current pixel is closer to the viewer than the corresponding value stored in the depth buffer. The depth buffer is implemented in hardware even in the low-cost graphics accelerators, allowing for tens of elevation map

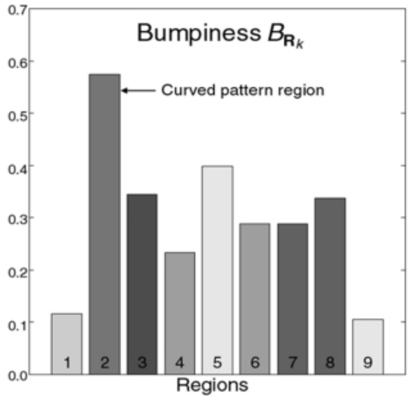

Figure 3. Bumpiness of the regions of the segmented mesh of figure 2.

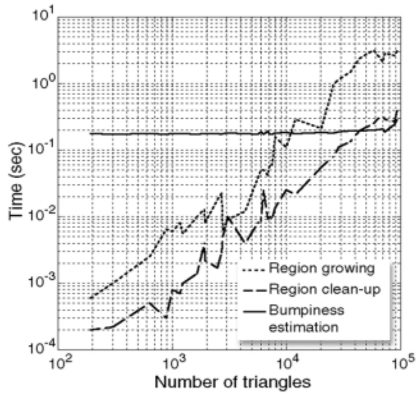

Figure 4. Execution time of the region growing, region cleanup and bumpiness estimation.

calculations per second, even for complex polygonal meshes.

6. Results

The region segmentation algorithm has been thoroughly tested with both computer-generated models and 3D-scanned or digitized objects, all represented as triangular meshes. The bumpiness criterion described in section 5 has been used to detect the engraved and cracked sides of real objects and the noisy or patterned sides of synthetic models.

The results of the tests were rated according to the similarity between the algorithm output and the manual partitioning or manual bumpy region selection. For scanned objects, bearing more than one engraved or broken sides, we considered as valid regions all those that stood out of the rest with respect to their bumpiness. The success rate, calculated by comparing these regions with the manually selected ones, was very satisfactory, being 100% for the segmentation of real objects and 95% for the detection of their bumpy surfaces. Figure 3 presents the results of the bumpiness measurement on the regions of the object in Figure 2.

Figure 4 shows a logarithmic chart of the execution time in seconds for the two surface segmentation stages as well as the average execution time for the bumpiness measurement on each mesh region. For all test objects (ranging from100 to 100,000 polygons) execution time of the overall process did not exceed 10 seconds. Notice that the bumpiness calculation, involving the elevation map extraction, runs in an almost constant time. This time depends only on the elevation map (depth buffer) resolution and on the ability of the rendering pipeline hardware to maintain a constant throughput of polygons. All tests were conducted on a PentiumIII/450MHz system equipped with a TNT2/185MHz graphics accelerator, using a 256×256 depth buffer.

The results of the bumpy face detection stage are used to drive a surface-by-surface matching system for the restoration of fragmented objects [6].

References

[1] Faugeras, O., *Three-Dimensional Computer Vision: A Geometric Viewpoint*, The MIT Press, Cambridge, Massachusetts, 1993.

[2] Gauss, K.F., *General Investigations of Curved Surfaces*, Raven Press, New York, 1965.

[3] B.K.P. Horn, “Extended Gaussian Image”, Proceedings of IEEE, IEEE, 1984, pp. 1671-1686.

[4] H. Delingette, M. Hebert, and K. Ikeuchi, “A Spherical Representation for the Recognition of Curved Objects”, Proceedings of International Conference on Computer Vision, 1993, pp. 103-112.

[5] K. Ikeuchi, and B.K.P. Horn, “Picking Up an Object from a Pile of Objects”, In: *Robotics Research: The First International Symposium*, J.M. Brady and R. Paul (eds.), The MIT Press, Cambridge, Massachusetts, 1984, pp.139-162.

[6] G. Papaioannou, E.A. Karabassi, T. Theoharis, “Reconstruction of Three-dimensional Objects through matching of their Parts”, submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.

[7] Jain, A.K., *Fundamentals of Digital Image Processing*, Prentice Hall, New Jersey, 1989.

[8] Foley, J.D., A. van Dam, S.K. Feiner and J.F. Hughes, *Computer Graphics: Principal and Practice*, Addison-Wesley, 1991.