Automated 3D Mesh Segmentation Using 2D Footprints

Wael Abdelrahman and Sara Farag*
Center of Intelligent Systems Research (CISR)
Deakin University

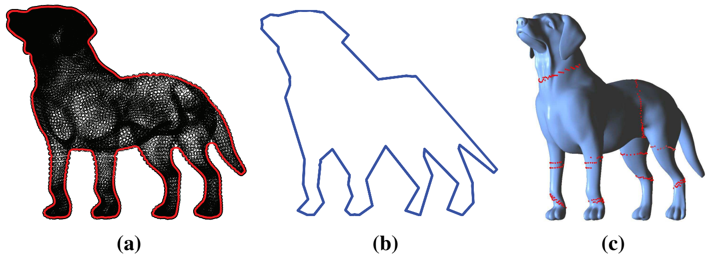

Figure 1: *(a) The 2D footprint of a dog 3D mesh. (b) The vector representation of the 2D footprint. (c) The resulting cut (segmentation) lines in 3D. These lines connects the points of concave discontinuity and their antipodals suggested by the proposed algorithm.*

1 Introduction

Segmenting 3D meshes into distinct components is vital and necessary for more efficient processing and usability. The smaller segments are usually easier to process and can be associated with with semantics or geometric features. This can be used in 3D parametrization, 3D database creation, animation, deformation transfer and many other 3D graphics applications. However, automating such a process is challenging due to the variety and complexity of the input.

To have an automated and usable algorithm, the number of parameters needs to be small or zero. An algorithm can have tuning parameters that for example change the output according to available resources (time and power) but should try to minimize or avoid knowledge based parameters that are dependent on the input such as the number of required segments. The closest theory and segmentation criteria to the human mind is the minima rule from the cognitive theory. This rule suggests that a human cuts 3D objects whenever a concave discontinuity is found. To satisfy this theory the 3D mesh needs to be searched for the discontinuities. Then, a closed contour (cut line) around the discontinuity needs to be created. Lee et al. [Lee et al. 2005] used a similar technique but required four parameters supplied to four functions (distance, normal, centricity and feature) to form the contours.

We propose a novel algorithm that satisfies the cognition theory but requires no object dependent parameters and only tuning parameters that can be fixed for all inputs. The algorithm searches for the cut lines end points and tries to create plausible closed contours.

2 Our Approach

The problem of 3D segmentation is defined mathematically on a boundary mesh $\mathcal{M}$ as generating a set of sub-segments $\mathcal{S}$, based on a certain criterion $\mathcal{C}$. [Shamir 2008]. There exist a class of implicit algorithms that produce the boundaries between the segments instead of the sub segments. The proposed algorithm uses 2D footprints of multiple 3D object poses to identify two opposite vertices that can be used to generate a closed contour.

The algorithm rotates a 3D boundary mesh around the three cartesian axes with a rate of θ. The smaller the θ value the greater the number of 2D poses and accuracy of the antipodals. For each pose the mesh is projected on the $x - y$ plane to form a set of 2D points. To get a vector format of the 2D projection, we need to have a set of lines that approximate its curvature. This requires the knowledge of the bounding curves of the 2D points set or their footprints. The footprint consist of the set of points that are defining the boundary. A recent survey by [Dupenois and Galton 2009] discusses various techniques of obtaining the 2D footprints. This footprint is then linearized to form a set of lines. Lines formation is computed by checking the angle ϕ between the pair of vectors $V_{1,2}$ and $V_{2,3}$ formed by the three points P_1, P_2 and P_3 is less than $\phi_{threshold}$. The $\phi_{threshold}$ is a tuning parameter and determines the number of the points of discontinuity to consider. This is general for any input 3d mesh.

The output of the vectorizing process is a set of n lines $\{l_1, l_2, ..., l_n\}$. Each line l approximates a set of the footprints points and a set of normals $\{N_1, N_2, ..., N_m\}$ is defined where m is the points count. Using this vector format, we can get the antipodal of the discontinuity vertices in 3D. The final step is to define a contour between each pair of points. This is accomplished by defining a plane that contains the two vertices and another very close point determined based on the pose. The adjacent vertices between the two points are selected based on their angle with the plane.

The preliminary results for a 3D dog mesh is shown in Figure1. The cut lines still need a post processing to produce smoother borders and remove redundant vertices. Nevertheless, the algorithm is fully autonomous and no object specific parameters are required. Future work will also include testing different categories of 3D objects and benching the algorithm against recent work.

References

DUPENOIS, M., AND GALTON, A., 2009. Assigning footprints to dot sets: An analytical survey.

LEE, Y., LEE, S., SHAMIR, A., COHEN-OR, D., AND SEIDEL, H.-P. 2005. Mesh scissoring with minima rule and part salience. *Comput. Aided Geom. Des. 22*, 5, 444–465.

SHAMIR, A. 2008. A survey on mesh segmentation techniques. *Computer Graphics Forum 27*, 6, 1539–1556.

*{wmam, sfara@deakin.edu.au}

SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007