Computer-Aided Design 78 (2016) 137–146

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Nearly convex segmentation of polyhedra through convex ridge separation☆

Guilin Liu, Zhonghua Xi, Jyh-Ming Lien*

George Mason University, Fairfax, VA, USA

ARTICLE INFO

Keywords:
Segmentation
Convexity
Shape approximation

ABSTRACT

Decomposing a 3D model into approximately convex components has gained more attention recently due to its ability to efficiently generate small decompositions with controllable concavity bounds. However, current methods are computationally expensive and require many user parameters. These parameters are usually unintuitive thus adding unnecessary obstacles in processing a large number of meshes with various types and shapes or meshes that are generated online within applications such as video games. In this paper, we investigate an approach that decomposes a mesh P based on the identification of *convex ridges*. Intuitively, convex ridges are the protruding parts of the mesh P. Our method, called CoRiSe, extracts nearly convex components of P by separating each convex ridge from all the other convex ridges through the new concept of *residual concavity*. CoRiSe takes a single user parameter: concavity tolerance which controls the maximum concavity of the final components, as input, along with other two fixed parameters for encoding the smoothness term of graph cut. We show that our method can generate comparable (sometimes noticeably better) segmentations in significant shorter time than the current state-of-art methods. Finally, we demonstrate applications of CoRiSe, including physically-based simulation, cage generation, model repair and mesh unfolding.

1. Introduction

In interactive applications, it is necessary to create simplified representations of a mesh to support various computationally intensive operations. In this work, we are interested in obtaining such simplified representations via decomposition. Taking deformation as an example, a mesh that has been decomposed into visually meaningful parts (e.g. head, torso, limbs) eases the process of creating deformation at semantic level. On the other hand, if a mesh is caged and partitioned by a set of convex shells, artists can use these shells to perform physically-based deformation efficiently. In many situations, both of these higher-level (semantic) and lower-level (physical or geometric) deformations are required. While it is ideal to keep both representations, it is also desirable to have an unified representation. An unified representation not only reduces space requirement but also allows deformations created at various semantic levels to be applied to the original mesh through a single approximation. Therefore, we recognize the need to have decomposition methods, such as the one proposed in this paper, that provide users both visual saliency and bounded geometric properties.

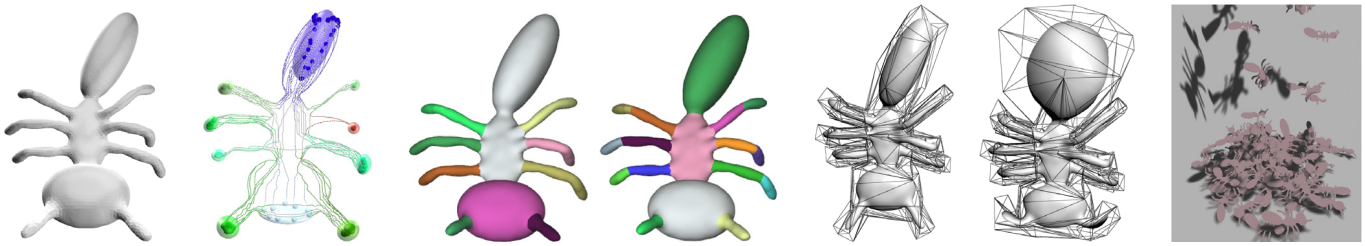
The first type of decomposition above is called *shape decomposition*. In the past decade, significant progress has been made in producing high quality results [1–7]. Shape decomposition provides implicit shape approximation and is useful for shape analysis and recognition and semantic level shape editing and deformation. The second type of decomposition can be accomplished through the Approximate Convex Decomposition (ACD) [8]. ACD decomposes a 3D mesh into nearly convex parts. Unlike shape decomposition, ACD provides explicit approximation with bounded approximation errors and is therefore suitable for lower level processing. For example, Bullet physics library uses hierarchical ACD (HACD) [9] to speed up the collision detection and response computation of the non-convex shapes, and Muller et al. [10] proposed an interactive tool to generate approximate convex parts for speeding up dynamic fracture simulation.

The most challenging task in developing such a decomposition method stems from the fact that current convex decomposition and shape segmentation methods are computationally expensive and require different parameter settings for different shapes which are usually unintuitive and make processing a massive

☆ This paper has been recommended for acceptance by Scott Schaefer and Charlie C.L. Wang.

* Corresponding author.
E-mail addresses: gliu2@gmu.edu (G. Liu), zxi@masonlive.gmu.edu (Z. Xi), jmlien@cs.gmu.edu (J.-M. Lien).

http://dx.doi.org/10.1016/j.cad.2016.05.015
0010-4485/



(a) Input. (b) Convex ridges. (c) Iterative decomposition. (d) Cage-based deformation. (e) Physically-based simulation.

Fig. 1. An example of CoRiSe with concavity tolerance $\tau = 0.05$. (a) Input mesh. (b) Ten convex ridges (shown as translucent ellipsoids) are connected by *deep valleys* (shown as the geodesic paths). Formal definition of convex ridge and valley can be found in Section 4. (c) Left is the decomposition result of the first iteration; right is final decomposition (the second iteration). Note that colors are just used to distinguish different parts. (d) CoRiSe used for automatic cage generation. (e) The convex hulls of CoRiSe components can be used to speedup the computation in physically-based simulation. (For interpretation of the references to color in this figure legend, the reader is referred to the electronic version of this article.)

number of meshes in applications difficult. Recently, learning based segmentation methods [4,5] are proposed to learn common parameters in an unsupervised or supervised manner, but the computation time of these methods is often intolerably high (minutes to hours), in particularly, for interactive applications.

Contribution. In this paper, we will describe an efficient decomposition method, called CoRiSe. The only user input parameter is a concavity tolerance which directly controls the approximation error. The novelty of CoRiSe comes from the idea of *convex ridge* which can be efficiently and robustly determined. Essentially, CoRiSe extracts nearly convex components of a 3D mesh P by separating each convex ridge from all the other convex ridges using graph cut. An example of convex ridge and CoRiSe decomposition is shown in Fig. 1. Formal definition of convex ridge can be found in Section 4. In addition, through the new concept of *residual concavity*, CoRiSe is insensitive to the parameter in graph cut and requires only a single user parameter: concavity tolerance. We will discuss residual concavity and graph cut in Section 5.

Using the Princeton Segmentation Benchmark, we show that CoRiSe generates noticeably better segmentation in significant shorter time than the existing methods [9,1]. Finally, we demonstrate applications of CoRiSe, including physically-based simulation, cage generation, model repair and mesh unfolding in Section 6. Fig. 1 shows two of these applications.

2. Related work

Many methods have been developed to decompose 3D mesh models. Comprehensive surveys can be found in [11–13]. In this section, we will review more recent works on shape segmentation and convex segmentation. After this short review, we will point out that the needs for developing a more intuitive and efficient method, such as CoRiSe.

Shape segmentation. Many of the existing single-shape segmentation methods are based on clustering mesh faces, such as k-means clustering [14], fuzzy clustering [15], mean-shift clustering [16], and spectral clustering [17]. Shape features, such as geodesic distance, local concavity, curvature, have significant impact on these clustering methods. More advanced features include shape diameter function [6] that is a measure of the diameter of the object's volume in neighborhood of a point, Mumford–Shah model [3] that measures the variation within a segment and continuous visibility feature [18] that uses more restricted visibility to better capture shape than the traditional line of sight visibility. Methods that are not clustering based do exist. For example, the method proposed by Wang et al. [19] uses the training segmentation of 2D projection images. Random cuts method [7] uses other different segmentation algorithms with various parameters to generate a collection of segmentations to find consistent cut positions.

Recently, we see more techniques using data-driven approach. These methods usually provide more consistent segmentations over a set of models [20] and produce segmentations that are more natural via machine learning approaches [4,5]. However, these methods are computationally intensive (require hours of computation) and are not suitable for interactive use.

Primitive segmentation. While many algorithms focus on decomposing a model into visually meaningful parts, other algorithms focuses on partitioning models into geometric primitives such as ellipsoids [21] and convex objects [8]. In other words, multiple primitives are used to jointly approximate the original shape.

In this paper, we are interested in producing nearly convex components. We found that many methods in the literature use again clustering (bottom-up) approach. Mamou and Ghorbel [9] proposed a method called Hierarchical Approximate Convex Decomposition (HACD) for 3D meshes. HACD iteratively clusters mesh facets by successively applying half-edge collapse decimation operations (see more detailed discussion in Section 6). Attene et al. [22] proposed to convert a model into tetrahedral mesh and then merge tetrahedra to form near convex components. However, this method requires human interaction, thus not suitable for segmenting a large number of models. The method proposed in [23] is based on a region growing approach controlled by convexity, compactness and part cost. Top-down approaches are rare. For example, Ghosh et al. [24] proposed a notion named *relative concavity* to model the concavity measure before and after every mesh cut. Nearly convex components are obtained by finding the cuts that have largest relative concavities via dynamic programming. In many of these methods, several parameters are needed to be set to balance various features.

We would like to note that there are also methods that use convexity and concavity but do not produce segmentation with bounded convexity or concavity. For example, Au et al. [2] used concavity-aware field to form potential cuts and the final cuts are achieved by maximizing the cut score. Even though concavity is used, it does not have direct control of concavity for final components and several parameters and thresholds need to be set. Asafi et al. [25] defined the convexity using the line-of-sight and applied spectral clustering on the visibility matrix to achieve segmentation. van Kaick et al. [1] applied merging on the initial result of [25] based on the Shape Diameter Function. However, since the pairwise visibility needs to be computed, the computation is time-consuming. Moreover, even though these two methods use convexity as clue for segmentation, they did not directly control the convex/concave geometric property for the final components.

Concavity and its counterpart, convexity, have shown to be valuable for various decomposition methods. Intuitively, concavity of a polyhedron P measures how different P is from a convex object, which is typically the smallest convex object enclosing P, i.e. the