Approximate Convex Decomposition*

Jyh-Ming Lien
Parasol Lab, Dept. of Computer Science
Texas A&M University
neilien@cs.tamu.edu

Nancy M. Amato
Parasol Lab, Dept. of Computer Science
Texas A&M University
amato@cs.tamu.edu

Categories and Subject Descriptors: I.3.5 [Computational Geometry and Object Modeling]: Geometric algorithms, languages, and systems

General Terms: Algorithms, Experimentation

Keywords: convex decomposition, concavity measure, polygon, polyhedron.

1. INTRODUCTION

Decomposition is a technique commonly used to break complex models into sub-models that are easier to handle. Convex decomposition, which partitions the model into convex components, is interesting because many algorithms perform more efficiently on convex objects than on non-convex objects. One issue with convex decompositions, however, is that they can be costly to construct and can result in representations with an unmanageable number of components. In many applications, the detailed features of the model are not crucial and in fact considering them only serves to obscure important structural features and adds to the processing cost. In such cases, an approximate representation of the model that captures the key structural features would be preferable.

Motivated by such issues, we propose a partitioning strategy that decomposes a given 2D or 3D model into *"approximately convex"* pieces. We propose a simple algorithm that computes an *approximate convex decomposition* (ACD) of a polygon or a 3D polyhedron. It proceeds by iteratively removing (*resolving*) the most significant non-convex feature (*notch*) until all components meet a user specified convexity tolerance. As a by product, it produces an elegant hierarchical representation that provides a series of 'increasingly convex' decompositions.

We have implemented our general approach for computing ACDs for polygons in the plane [2] and for polyhedra in three-dimensions [1]. In the next sections we briefly describe our approach and then present some experimental results. Please see the above cited papers for more details.

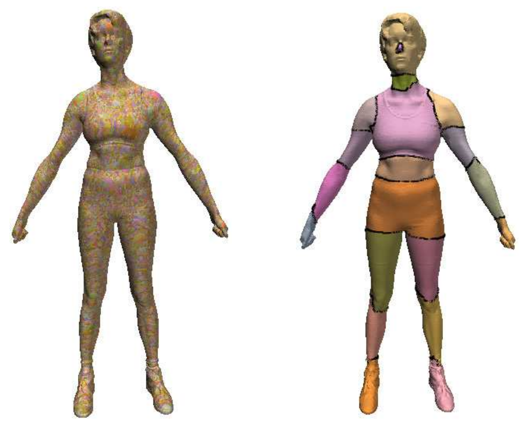

Figure 1. This model has 243,442 triangles and 141,837 notches. Left: Exact convex surface decomposition – 44,461 components. Right: Approximate convex surface decomposition – 20 components (concavity < 0.05).

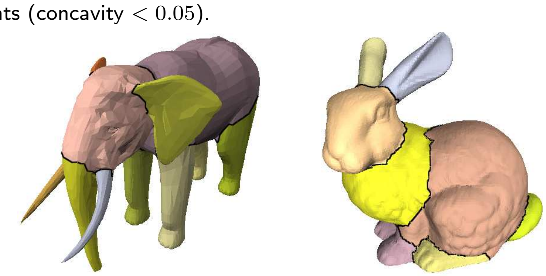

Figure 2. Both models are decomposed into 0.1-approximate convex components. The elephant and the bunny are decomposed into 14 and 10 components, respectively.

2. OUR APPROACH

Our approach is based on the premise that for some models and applications, some of the non-convex (concave) features can be considered *less significant*, and allowed to remain in the final decomposition, while others are more important, and must be removed (resolved). Accordingly, our strategy is to identify and resolve the non-convex features in order of importance. Due to the recursive applica-

*This research supported in part by NSF Grants ACI-9872126, EIA-9975018, EIA-0103742, EIA-9805823, ACR-0081510, ACR-0113971, CCR-0113974, EIA-9810937, EIA-0079874, and by Texas HECB grant ATP-000512-0261-2001.

SCG'04, June 8–11, 2004, Brooklyn, New York, USA.
ACM 1-58113-885-7/04/0006.

tion, the resulting decomposition is a hierarchical binary tree. If the process is halted before convex components are obtained, then the leaves of the tree are approximate convex components. Thus, our approach also constructs a hierarchical representation that provides multiple Levels of Detail (LOD). A single decomposition is constructed based on the highest accuracy needed, but coarser, "less convex" models can be retrieved from higher levels in the decomposition hierarchy when the computation does not require that accuracy.

Our goal is to generate τ-*approximate* convex decompositions. For a given model P, P is said to be τ-*approximate* convex if concave$(P) < \tau$, where concave(ρ) denotes the concavity measurement of ρ. Here, τ represents a tunable parameter denoting the non-concavity tolerance for the application. A τ-*approximate* convex decomposition of P, $\mathrm{CD}_\tau(P)$, is defined as a decomposition $D(P)$ that contains only τ-*approximate* convex components; i.e.,

$$\mathrm{CD}_\tau(P) = \{C_i \mid C_i \in \mathrm{D}(P) \text{ and } \mathrm{concave}(C_i) \leq \tau\}. \quad (1)$$

The success of our approach depends critically on the quality of the methods we use to prioritize the importance of the non-convex features. Intuitively, important features provide key structural information for *the application.* Although curvature has been one of the most popular measures used to extract visually salient features, it is quite unstable because it identifies features from local variations on the model's boundary. In contrast, the concavity measures we consider for computing ACDs identify features using global properties of the boundary.

We define the *concavity of a point* x on P as the distance from x to H, the convex hull of P. Then, the *concavity of* P is defined as the maximum concavity of its vertices. For polygons, a notch (concave feature) x is enclosed by exactly one line segment β of the convex hull H. In our methods, we measure the concavity by computing the distance from x to β; in [2] we propose and compare three methods for computing $\mathrm{dist}(x, \beta)$. For polyhedra, a notch x may be enclosed by more than one facet of the convex hull H of P. To identify which hull facet is closest to x, we project the hull facets onto P and find the facet that covers x. See Figure 3.

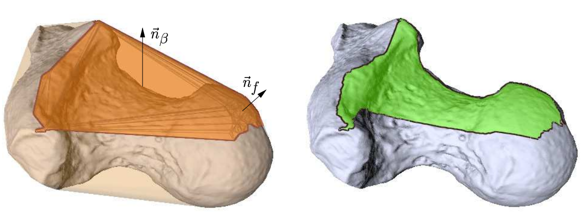

Figure 3. Associating convex hull facets (left) with vertices (right). A set of facets are grouped and projected onto P together.

After the concavity is measured, the model is decomposed if its concavity exceeds the threshold τ. To decompose a polygon, a diagonal is added to the vertex with maximum concavity. To decompose a polyhedron into solid parts, the model is bisected by a cut plane incident to the

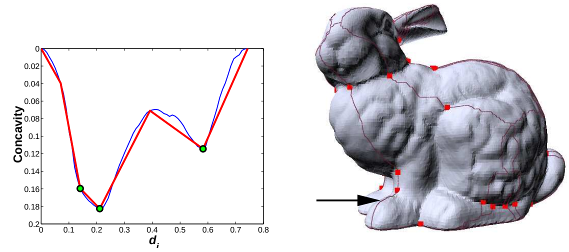

Figure 4. A line (the arrow) on the Stanford Bunny is mapped to the plot (left). Features are identified from this line and are marked as dots. Red dots on the bunny show the features found.

(a) (b) (c)

Figure 5. (a) Red dots are identified features. (b) Dark area indicate high concavity regions. (c) High concavity features are connected and the model is decomposed along these paths.

most concave notch. To decompose a polyhedron into approximately convex surface patches, the model is cut along "concave" paths on the model's surface that are projections of edges bounding convex hull facets. Figures 4 and 5 show this process.

3. RESULTS

In the video, we show approximate decompositions computed by our method for polygons, with or without holes, and for polyhedra. In three-dimensions, we show examples of decomposing the polyhedron into approximately convex solid components and for partitioning its surface into approximately convex surface patches. From our experimental results, we observe that if an application can sacrifice a little convexity, then our algorithm can produce fewer components than the exact convex decompositions in significantly less time. Figure 1 shows the difference between exact and approximate convex surface decomposition.

Another important feature of our approximate convex decomposition is its ability to identify key structural features of the model. For instance, the Stanford Bunny and the elephant model in Figure 2 are decomposed into submodels that reflect anatomical structures.

References

[1] J.-M. Lien and N. M. Amato. Approximate convex decomposition. Technical Report TR03-001, Parasol Lab, Dept. of ComputerScience, Texas A&M University, Jan 2003.

[2] J.-M. Lien and N. M. Amato. Approximate convex decomposition for polygons. In *Proceedings of the 20th Annual ACM Symposium on Computational Geometry (SoCG 2004)*, 2004. To appear.