Variational, meaningful shape decomposition

Vladislav Krayevoy and Alla Sheffer *
University of British Columbia

1 Introduction

Mesh decomposition into meaningful components has many applications in modeling and CG and has received a lot of attention in the last few years. The accepted notion of meaningful parts relies on human perception, and is based on the observation that human vision defines part boundaries along negative minima of principal curvatures. This definition implies that the meaningful parts are, in some sense, convex. Many existing decomposition methods, e.g. Katz et al. [2005], define meaningful components by locating farthest points using geodesic distances. Due to the reliance on surface distances, such methods do not perform well on models with no clear extremities, or models with high genus such as the fishing reel or David in Fig 1. Similar to Lien and Amato [2006] we avoid those problems by basing our algorithm on convexity criteria rather than surface distances. We develop a new quantifiable convexity metric, leading to a robust decomposition algorithm that does not require model-specific fine-tuning and outperforms Lien and Amato [2006] as discussed below. Using a single threshold parameter, our algorithm successfully segments models into meaningful parts. The method operates on models with arbitrary genus and with multiple boundaries.

2 Algorithm

The goal of our decomposition algorithm is to segment the model into a small number of nearly convex, compact patches. Given a 3D model, the sole input required from the user is a threshold value specifying the convexity of each generated part. In contrast to previous methods we do not expect the user to provide an estimated number of patches for the decomposition.

Metrics: We base our algorithm on a convexity metric that measures the distance between a mesh patch P and its convex hull $C(P)$. The distance is defined as an area weighted average of the distances from the patch triangles t to the convex hull:

$$dist(P,C(P)) = \frac{\sum_{t\in P} dist(t,C(P))\cdot area(t)}{\sum_{t\in P} area(t)} \qquad (1)$$

where $area(t)$ is the area of the triangle t, and $dist(t,C(P))$ is the distance from the triangle t to the convex hull $C(P)$ along the direction of the triangle's normal. To achieve a useful decomposition, it is not enough for the patches to be nearly convex, they must also be compact. We calculate the compactness as an area to volume ratio of the convex hull C: $comp(C) = area(C)/volume(C)^{2/3}$. We combine the convexity metric with a volumetric measure of compactness to define a cost function for a potential patch as

$$cost(P) = (1+dist(P,C(P)))\cdot(1+comp(C(P)))^{\alpha}, \qquad (2)$$

where α controls the tradeoff between the two and is a constant for all our examples.

Patch Formation: To generate the patches we use a modified Lloyd iteration scheme. In contrast to Cohen-Steiner et al. [2004] we start from zero patches and use the convexity threshold to automatically establish the necessary number of patches. Our algorithm generates patches using the following four stage procedure.

*e-mail: {vlady, sheffa}@cs.ubc.ca

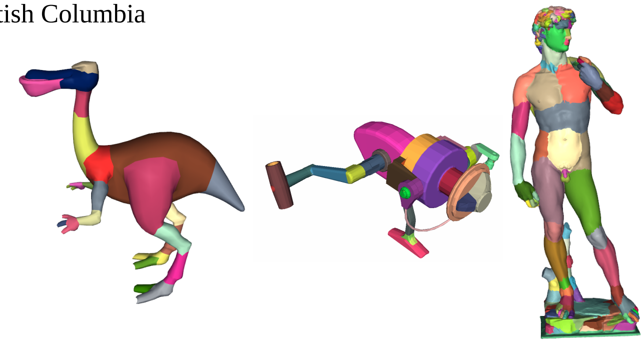

Figure 1: Mesh decomposition into meaningful components.

1. Potential patch generation: The method collects the unassigned triangles, which do not belong to any patch, into connected components classifying each as a potential new patch P and computing $C(P)$.

2. Seed generation: This stage finds seeds for re-growing existing and potential patches. A seed consists of a seed triangle and a seed convex hull. We define the seed convex hull as the tetrahedron formed by the seed triangle and the center point of the current convex hull. This selection reduces the difference between the proxies obtained at consecutive iterations. To further minimize the difference we select a seed triangle which fits well the current proxy, namely one that is close to the current convex hull. After the seeds have been selected, we reset the patches by marking the rest of the triangles as unassigned.

3. Patch growing: For each vertex v that shares an edge with a current patch, the algorithm computes the insertion cost to be the cost of the updated patch formed by adding v to P, $cost(P+v)$. At each growth step, the algorithm uses the cost function to find the best adjacent vertex to add to one of the patches. If the convexity error $dist(P+v,C(P+v))$ of the updated patch is below the threshold, the vertex is added to the patch. If no such vertex is found growth is terminated and the algorithm proceeds to Stage 4.

4. Termination: The algorithm now tests for termination conditions. If the new patches differ from the ones grown in the previous iteration, the algorithm repeats the reseeding and growing loop, Stages 2 and 3. Once the patches no longer change, we check if the patches cover the entire model. If not, the algorithm returns to Stage 1, otherwise the algorithm terminates.

Figure 1 shows a number of decompositions generated using our method. The algorithm correctly detects the perceptually meaningful parts of the models, identifying even small features like fingers and toes. Our method correctly segments high genus models such as the David, and the fishing reel which cause problems to many previous methods. Unlike Lien and Amato [2006], our algorithm correctly identifies the tail of the dinopet and correctly separates David's right leg from the base (Figure 1).

References

COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape approximation. *ACM Trans. Graph. 23*, 3, 905–914.

KATZ, S., LEIFMAN, G., AND TAL, A. 2005. Mesh segmentation using feature point and core extraction. In *Proceedings of Pacific Graphics*, vol. 21, 649–658.

LIEN, J.-M., AND AMATO, N. M. 2006. Approximate convex decomposition of polyhedra. Tech. Rep. TR06-002, Texas A&M University.