A SIMPLE AND EFFICIENT APPROACH FOR 3D MESH APPROXIMATE CONVEX DECOMPOSITION

Khaled Mamou and Faouzi Ghorbel

Groupe de recherche GRIFT, Laboratoire CRISTAL, Ecole Nationale des Sciences de l'Informatique
University of Manouba, 2010 Manouba-Tunisia
khaled_mamou@yahoo.fr, faouzi.ghorbel@ensi.rnu.tn

ABSTRACT

This paper presents an original approach for 3D mesh approximate convex decomposition. The proposed algorithm computes a hierarchical segmentation of the mesh triangles by applying a set of topological decimation operations to its dual graph. The decimation strategy is guided by a cost function describing the concavity and the shape of the detected clusters. The generated segmentation is finally exploited to construct a faithful approximation of the original mesh by a set of convex surfaces. This new representation is particularly adapted for collision detection. The experimental evaluation we conducted shows that the proposed technique efficiently decomposes a concave 3D mesh into a small set (with respect to the number of its facets) of nearly convex surfaces. Furthermore, it automatically detects the anatomical structure of the analyzed 3D models, which makes it an ideal candidate for skeleton extraction and patterns recognition applications.

Index Terms— Approximate convex decomposition, 3D mesh, collision detection, hierarchical segmentation

1. INTRODUCTION

Collision detection is essential for realistic physical interactions in video games, computer animation and physically-based modeling. In order to ensure real-time interactivity with the player/user, video game and 3D modeling software developers usually approximate the 3D models composing the scene (e.g. animated characters, static objects...) by a set of simple convex shapes such as ellipsoids, capsules or convex-hulls. In practice, these simple shapes provide poor approximations for concave surfaces and generate false collision detections.

In this article, we present a simple and efficient approach to decompose a 3D mesh into a set of nearly convex surfaces. This decomposition is directly exploited to compute a faithful approximation of the original 3D mesh, particularly adapted to collision detection.

The remainder of this paper is structured as follows. Section 2 introduces the approximate convex decomposition problem. The proposed segmentation technique is described in Section 3 and its performances are evaluated in Section 4. Finally, Section 5 concludes the paper and suggests possibilities for future work.

2. APPROXIMATE CONVEX DECOMPOSITION

Let S be a 3D mesh defined in $\mathbb{R}^3$, $\xi = \{v_1, v_2, ..., v_V\}$ the set of its vertices (V represents the number of vertices) and $\theta = \{t_1, t_2, ..., t_T\}$ the set of its triangles (T represents the number of triangles).

Computing an exact convex decomposition of S consists in partitioning it into a minimal set of convex sub-surfaces. In [1], the authors prove that computing such decomposition is an NP-hard problem and propose different heuristics to resolve it. In [2, 3], Lien *et al.* point out the fact that the proposed algorithms are non-practical since they produce a high number of clusters. In order to provide a tractable solution, they propose to relax the exact convexity constraint and consider instead the problem of computing an approximate convex decomposition of S. Here, for a fixed parameter ϵ, the goal is to determin a partition $\Pi = \{\pi_1, \pi_2, ..., \pi_K\}$ of θ with a minimal number of clusters K and verifying that each cluster has concavity lower than ϵ.

To the best of our knowledge, the approximate convex decomposition problem was only addressed by [2, 3]. This later technique exploits a divide-and-conquer strategy, which consists in iteratively dividing the mesh until the concavity of each sub-part is lower than the threshold ϵ. Here, at each step i, the vertex v_i^* with the highest concavity is selected and the cluster to which it belongs is divided into two sub-clusters by considering a bisection plane incident to v_i^*. The main limitation of this appraoch is related to the choice of the "best" cut plane, which requires a sophisticated analysis of the model features [2, 3]. Moreover, considering only plane-based bisections is in practice too restrictive and may lead to poor decompositions (*cf.* Figure 3).

In order to overcome such limitations, this paper introduces a novel hierarchical segmentation approach for 3D mesh approximate convex decomposition, described in the next section.

3. PROPOSED APPROACH

The proposed hierarchical segmentation approach proceeds as follows. First, the dual graph of the mesh is computed. Then its vertices are iteratively clustered by successively applying topological decimation operations, while minimizing a cost function related to the concavity and the aspect ratio of the produced segmentation clusters.

Let's first recall the definition of the dual graph of the mesh.

3.1. Dual graph

The dual graph S^* associated to the mesh S is defined as follows:

- each vertex of S^* corresponds to a triangle of S,
- two vertices of S^* are neighbours (i.e., connected by an edge of the dual graph) if and only if their corresponding triangles in S share an edge.

3.2. Decimation operator

Once the dual graph S^* is computed, the algorithm starts the decimation stage which consists in successively applying halfe-edge collapse [4] decimation operations. Each half-edge collapse operation applied to an edge (v, w), denoted $hecol(v, w)$, merges the two vertices v and w. The vertex w is removed and all its incident edges are connected to v (cf. Figure 1).
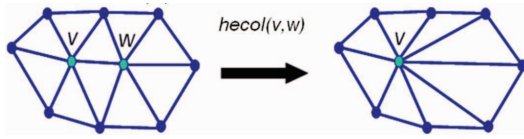

Fig. 1. *Half-edge collapse* decimation operation.

Let $A(v)$ be the list of the ancestors of the vertex v. Initially, $A(v)$ is empty. At each operation $hecol(v, w)$ applied to the vertex v, the list $A(v)$ is updated as follows :

$$A(v) \leftarrow A(v) \cup A(w) \cup \{w\}. \qquad (1)$$

3.3. Simplification strategy

The decimation process described in the previous section is guided by a cost function describing the concavity and the aspect ratio [5] of the surface $S(v, w)$ resulting from the unification of the vertices v and w and their ancestors:

$$S(v, w) = A(v) \cup A(w) \cup \{w, v\}. \qquad (2)$$

As in [5], we define the aspect ratio $E_{shape}(v, w)$ of the surface $S(v, w)$ as follows:

$$E_{shape}(v, w) = \frac{\rho^2(S(v, w))}{4\pi \times \sigma(S(v, w))}, \qquad (3)$$

where $\rho(S(v, w))$ and $\sigma(S(v, w))$ are respectively the perimeter and the area of $S(v, w)$.

The cost $E_{shape}(v, w)$ was introduced in order to favor the generation of compact clusters. In the case of a disk this cost equals one. The more irregular a surface is the highier its aspect ratio.

The decimation cost $E(v, w)$ associated to the edge (v, w) is given by:

$$E(v, w) = \frac{C(S(v, w))}{D} + \alpha E_{shape}(v, w), \qquad (4)$$

where

- $C(S(v, w))$ is the concavity of $S(v, w)$ (cf. Section 3.4),
- D is a normalization factor equal to the diagonal of the bounding box of S,
- α is a parameter controlling the contribution of the shape factor $E_{shape}(v, w)$ with respect to the concavity cost (cf. Section 3.5).

At each step of the decimation process, the *hecol* operation with the lowest decimation cost is applied and a new partition $\Pi(n) = \{\pi_1^n, \pi_2^n, \pi_3^n, ..., \pi_{K(n)}^n\}$ is computed as follows:

$$\forall k \in \{1, ..., K(n)\}, \pi_k^n = p_k^n \cup A(p_k^n), \qquad (5)$$

where $(p_k^n)_{k \in \{1,...,K(n)\}}$ represent the vertices of the dual graph S^* obtained after n half-edge collapse operations. This process is iterated until all the edges of S^* generating clusters with concavities lower than ϵ are decimated.

3.4. Concavity measure

As discussed in [2, 3], there is no consensus in the literature on a quantitative concavity measure. In this work, we define the concavity $C(S)$ of a 3D mesh S, as follows (cf. Figure 2):

$$C(S) = \arg \max_{M \in S} \|M - P(M)\|, \qquad (6)$$

where $P(M)$ represents the projection of the point M on the convex-hull $CH(S)$ [6] of S, with respect to the half-ray with origin M and direction normal to the surface S at M.

Let us note that the concavity of a convex surface is zero. Intuitively, the more concave a surface the "farther" it is from its convex-hull. The definition (6) extends directly to open meshes once oriented normals are provided for each vertex.

3.5. Choice of the parameter α

The clusters detected during the early stages of the algorithm are composed of a low number of adjacent triangles with a concavity almost equal to zero. Therfore, the decimation cost E is dominated by the aspect ratio cost E_{shape}, which favors
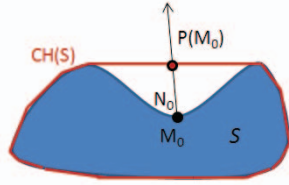
Fig. 2. Concavity measure for a 3D mesh.

the generation of compact surfaces. This behavior is progressively inverted during the decimation process since the clusters are becoming more and more concave. In order to ensure that the cost $(\alpha.E_{shape})$ has no influence on the choice of the last decimation operations, we have set the parameter α as follows:

$$\alpha = \frac{\epsilon}{10 \times D}. \qquad (7)$$

This choice garanty, for disk-shaped clusters, that the cost $(\alpha.E_{shape})$ is ten times lower than the concavity-related cost $\frac{C(S(v,w))}{D}$.

4. EXPERIMENTAL RESULTS

In order to validate our approach, we have first compared it segmentation results to those of the ACD technique[1] introduced in [2, 3]. Figure 3 presents the convex-hulls generated by our approach to those of [2, 3]. The reported results show that our approach provides better convex approximations while detecting a lower number of clusters.

Figure 4 presents the segmentation results and the convex-hull approximations generated by our approach for diffrent 3D meshes. Let us note first that, for all the models, the generated decompositions ensures a concavity lower than ϵ while generating a small number of clusters, which provides an efficient representation for collision detection. Moreover, the proposed technique successfully detects the convex parts (*e.g.* the four spheres of Figure 4.p) and the anatomical structure of the analyzed 3D models.

5. CONCLUSION

We have presented a hierarchical segmentation approach for approximate convex decomposition of 3D meshes. The generated segmentations are exploited to construct faithful approximations of the original mesh by a set of convex surfaces. This new representation is particularly adapted for collision detection. The experimental evaluation we conducted shows that the proposed technique efficiently decomposes a concave 3D model into a small set of nearly convex surfaces, while automatically detecting its anatomical structure which makes

---

[1]We have considered the only public implementation of the algorithm avaible at http://codesuppository.blogspot.com/2006/04/approximate-convex-decomposition.html

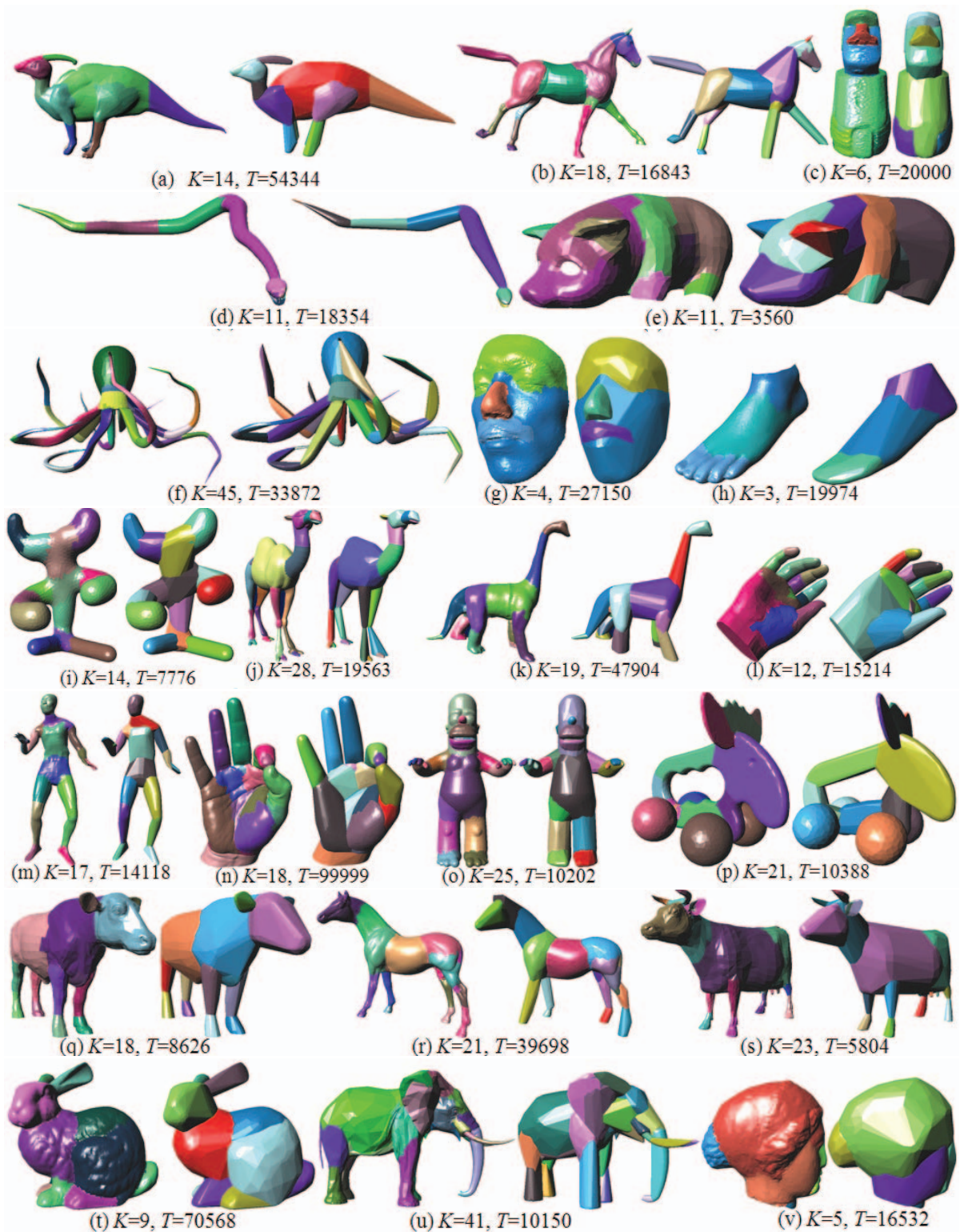it an ideal candidate for skeleton extraction and pattern recognition applications.

(a) $T = 5804$ (b) $K = 28$ (c) $K = 23$

(d) $T = 39689$ (e) $K = 12$ (f) $K = 12$

(g) $T = 19563$ (h) $K = 26$ (i) $K = 22$
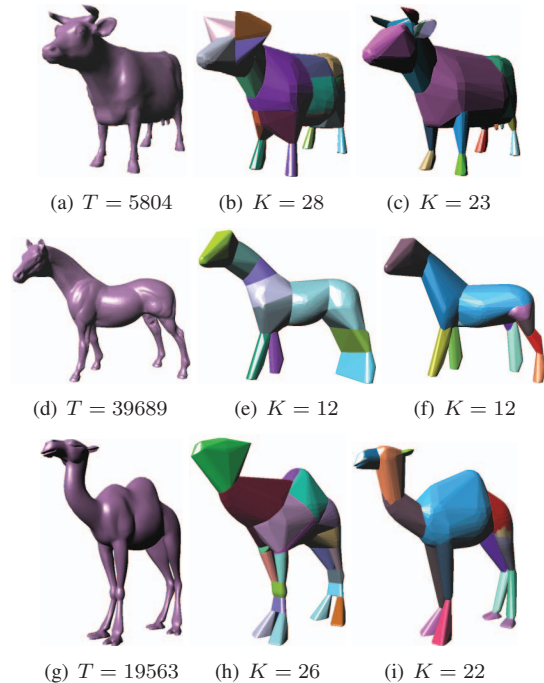
Fig. 3. Comparative evaluation: (a,d,g) original meshes, (b,e,h) convex-hulls generated by [2, 3] and (c,f,i) convex-hulls generated by our approach (K number of clusters and T number of triangles).

6. REFERENCES

[1] B. Chazelle, D.P. Dobkin, N. Shouraboura, and A. Tal, "Strategies for polyhedral surface decomposition: an experimental study," in *Symposium on Computational Geometry*, 1995, pp. 297–305.

[2] J.M. Lien and N.M. Amato, "Approximate convex decomposition," in *Symposium on Computational Geometry*, 2004, pp. 457–458.

[3] J.M. Lien and N.M. Amato, "Approximate convex decomposition of polyhedra and its applications," *Computer Aided Geometric Design*, pp. 503–522, 2008.

[4] H. Hoppe, "Progressive meshes," in *International Conference on Computer Graphics and Interactive Techniques*, 1996, pp. 99–108.

[5] A. K. Jain, "Fundamentals of digital image processing," *Prentice-Hall International*, 1989.

[6] F.P. Preparata and S.J. Hong, "Convex hulls of finite sets of points in two and three dimensions," *Communication of the ACM*, vol. 20(2), pp. 87–93, 1977.

(a) K=14, T=54344

(b) K=18, T=16843

(c) K=6, T=20000

(d) K=11, T=18354

(e) K=11, T=3560

(f) K=45, T=33872

(g) K=4, T=27150

(h) K=3, T=19974

(i) K=14, T=7776

(j) K=28, T=19563

(k) K=19, T=47904

(l) K=12, T=15214

(m) K=17, T=14118

(n) K=18, T=99999

(o) K=25, T=10202

(p) K=21, T=10388

(q) K=18, T=8626

(r) K=21, T=39698

(s) K=23, T=5804

(t) K=9, T=70568

(u) K=41, T=10150

(v) K=5, T=16532

Fig. 4. Segmentation results and generated convex-hulls ($\epsilon = 0.03 \times D$, D length of the bounding box diagonal, K number of clusters and T number of triangles).