Computer-Aided Design ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer-Aided Design

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

Disjoint convex shell and its applications in mesh unfolding[☆]

Yun-hyeong Kim [a], Zhonghua Xi [b], Jyh-Ming Lien [b,*]

[a] Korea Institute of Science and Technology, Seoul, Republic of Korea
[b] George Mason University, Fairfax VA, USA

ARTICLE INFO

Keywords:
Convex approximation
Decision boundary
Optimization
Folding
Unfolding
Convexification

ABSTRACT

In this work, we study a geometric structure called *disjoint convex shell* or simply DC-shell. A DC-shell of a polyhedron is a set of pairwise interior disjoint convex objects that collectively approximate the given polyhedron. Preventing convex objects from overlapping enables faster and robust collision response and more realistic fracturing simulation. Without the disjointness constraint, a physical realization of the approximation becomes impossible. This paper investigates multiple approaches that construct DC-shells from shapes that are either composed of overlapping components or segmented into parts. We show theoretically that, even under this rather simplified setting, constructing DC-shell is difficult.

To demonstrate the power of DC-shell, we studied how DC-shell can be used in mesh unfolding, an important computational method in manufacturing 3D shape from the 2D material. Approximating a given polyhedron model by DC-shells provides two major benefits. First, they are much easier to unfold using the existing unfolding methods. Second, they can be folded easily by both human folder or self-folding machines. Consequently, DC-shell makes paper craft creation and design more accessible to younger children and provides chances to enrich their education experiences.

1. Introduction

Approximating a 3D mesh by disjoint convex objects enables more applications than approximating it using overlapping convex objects. Examples include faster and more robust collision response, better local penetration depth estimation, faster volumetric meshing and more realistic fracturing simulation. Without this disjointness constraint, physical realization of the approximation is not even possible. We call this geometric structure, disjoint convex shell or simply **DC-shell**. The word *shell* is used to avoid confusion with convex hull.

There exist several ways to produce DC-shell, including solid convex segmentation [1]. To the best of our knowledge, our work is the first attempt that produces practical non-overlapping convex approximation whose number of disjoint convex components is significantly smaller than those created by existing methods. The most relevant work that we found is in computational geometry. There, researchers studied methods to cover disjoint convex sets with non-overlapping convex polygons [2]. Our problem, on the other hand, is to find non-overlapping convex shapes approximating non-convex overlapping sets.

In this work, we propose an optimization method that constructs DC-shells from the convex hulls enclosing the parts of a composite shape, i.e., shape that is composed of multiple parts or is segmented either manually or algorithmically from a single mesh. The convex hulls of these parts often overlap. Many meshes come in this form, in particular those available in public model-sharing sites, such as *Thingiverse* or Google 3D warehouse, or those created for video games and animations. Later in this paper, we will show that, even under this somehow simplified setting, the problem of converting overlapping convex objects to disjoint convex objects is difficult and computationally expensive. Moreover, straightforward heuristic methods often generate undesirable convex shells. Fig. 1 illustrates some failed examples obtained from a method that trims the overlapping convex shapes using the cut planes that least-squares fit the boundary intersections. This heuristic method (detailed in Section 3.1) in many cases generates results with significant volume loss.

The proposed method can produce consistent results with little volume loss. The key contribution of this method is an easy-to-implement and efficient approach that first obtains an initial guess of the cut planes via support vector machine (SVM) [3] of points sampled from the interior of the input convex hulls, and then minimizes the volume loss via the derivative of exact volume computation between a moving cutting plane and a pair of convex shapes. Because the convex hulls of the parts in a composite shape often contain long and skinny triangles and, because the cutting

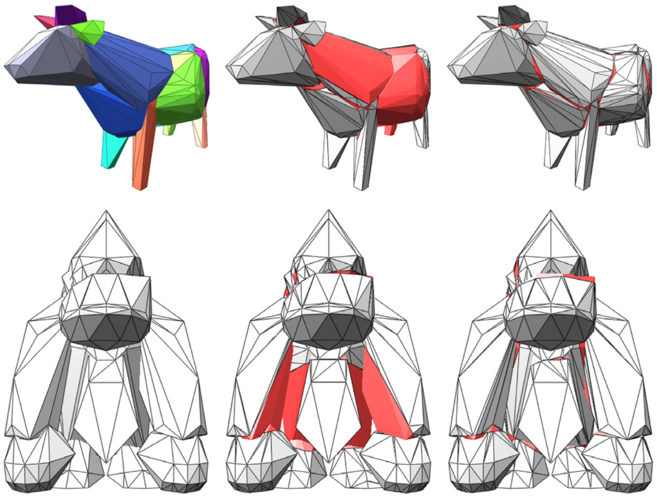
---

[☆] This paper has been recommended for acceptance by Dr. Mario Botsch, Dr. Yongjie Jessica Zhang and Dr. Stefanie Hahmann.

[*] Corresponding author.
E-mail address: jmlien@cs.gmu.edu (J.-M. Lien).

http://dx.doi.org/10.1016/j.cad.2017.05.014
0010-4485/© 2017 Elsevier Ltd. All rights reserved.

Please cite this article in press as: Kim Y.-h., et al. Disjoint convex shell and its applications in mesh unfolding. Computer-Aided Design (2017), http://dx.doi.org/10.1016/j.cad.2017.05.014.



(a) Input. (b) Heuristics. (c) Optimized.

Fig. 1. Left: Overlapping convex shapes depicting a cow and Donkey Kong. Middle: Cutting overlapping convex objects via boundary intersections (detailed in Section 3.1) results in significant volume loss (shown in red), which is defined as the volume inside the input shape but outside the disjoint convex shell. Right: Disjoint convex shells created by our optimization method described in Section 3.2 have significantly smaller volume loss. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

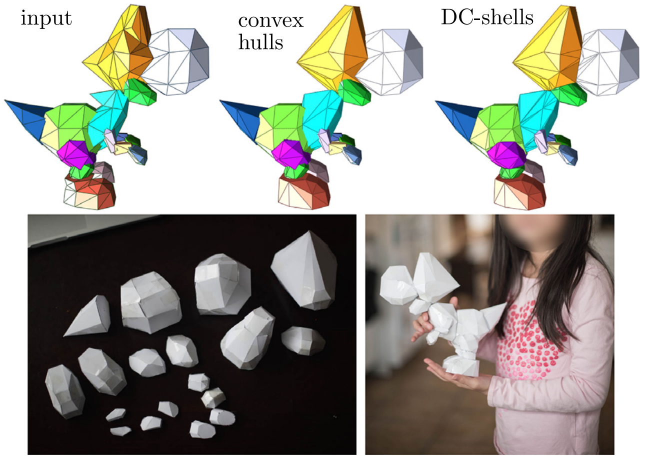


Fig. 2. The top three figures, from left to right, show the composite shape of Yoshi model obtained from www.thingiverse.com, its convex hulls and its DC-shells. The bottom two photos show the folded disjoint convex shells and assembled Yoshi model.

process often introduces new sliver triangles near the cuts, it is critical that these slivers are removed. To remove slivers of DC-shells while guaranteeing the convexity and disjointness, we contribute a new remeshing method in Section 4. As illustrated in Fig. 1 and later in Section 6, our results show that the DC-shells created by the proposed method have significantly lower volume loss than those created by least-squares-fit and SVM-only methods.

Throughout the paper, we will use mesh unfolding to demonstrate the power of DC-shell. Mesh unfolding is an important computational problem in manufacturing a 3D shape from a 2D material [4–6]. Designing a foldable 3D shape usually involves two main foldability analysis steps: instantaneous unfolding and continuous folding. Instantaneous unfolding transforms a polyhedron to a 2D representation instantaneously. The continuous folding problem aims to find a foldable path that transforms the net to its folded shape continuously without self-intersection. Both steps are significantly easier for convex shapes than non-convex shapes [7,8].

Through its application in mesh unfolding, we show that DC-shells can be rigidly unfolded into high-quality nets with minimal cut length and coverage. We verify the foldability of these nets created from DC-shells algorithmically and through a user study with 102 school-age children. We show that children can work together and create rather complex 3D paper crafts around an hour in a hands-on story-telling class. Because these 3D paper crafts are made of blocks of convex objects, students can freely reconfigure their poses and even enlarge certain parts before assembly. Fig. 2 shows an example of a composite shape (Yoshi) obtained from www.thingiverse.com that contains overlapping parts and its DC-shells and fabricated paper model.

2. Related works

2.1. Convex decomposition and approximation

Convexity provides significant algorithmic benefits in many problems and motivates researchers to approximate shapes with convex objects. Covering a shape with non-overlapping convex shapes has been extensively studied. For example, exact convex decomposition [1,9,10] segments mesh into disjoint convex components. Another example is tetrahedral mesh generation. Both approaches can produce a large number of convex components. Exact convex decomposition is known to produce $O(r^2)$ convex components for a polyhedron with r reflex edges.

Approximations that allow overlapping convex shapes are more prevailing. For instance, approximating a shape with a set of primitive convex shapes, such as spheres, boxes, ellipsoids, and capsules, usually, forms a hierarchy of overlapping volumes. Another example is approximate convex decomposition (ACD) [11–14] that approximates the input mesh by a set of nearly convex shapes. Applications of ACD usually work solely with the convex hulls of the segmented parts and ignore the input mesh. These convex hulls usually overlap.

2.2. Polyhedra unfolding

In the problem of rigidly unfolding the polyhedron to a 2D structure called the *net*, convexity admits simple edge unfolding of every convex polyhedron [8]. Finding a valid net of a given polyhedron is known to be nontrivial because a polyhedron with $\|F\|$ faces can have approximately $O(2^{\sqrt{\|F\|}})$ different unfoldings and most of them contain overlaps especially for non-convex polyhedra. However, if input shapes are convex polyhedra, heuristic methods work well in practice. Most, if not all, nets of convex polyhedra with $\|F\|$ facets can be obtained in $O(\|F\| log \|F\|)$ time. Mitani and Suzuki [4] decompose the mesh into a few patches and approximate each patch with a strip, a generalized cylinder or a developable surface. Schlickenrieder [7] proposed heuristics methods for unfolding a polyhedron to a net. Straub and Prautzsch, Takahashi et al. [15, 16] extend [7] to unfold non-convex polyhedra. Recently, Xi et al. [17] proposed an approach that segments a model into a smell set of semantic and easily unfoldable parts by learning from failed unfoldings. It is worth noting that (rigid) mesh unfolding faces computational challenges and issues different from a line of mesh flattening works that focuses on approximating surface patches with developable surfaces [18,19]. All aforementioned works generate non-overlapping unfoldings, however, it is not guaranteed that there exists a continuous folding motion that transforms the net back to its original shape.

In order to make a physical copy of the foldable shape that can be continuously folded to its target shape from the net, we need

Please cite this article in press as: Kim Y.-h., et al. Disjoint convex shell and its applications in mesh unfolding. Computer-Aided Design (2017), http://dx.doi.org/10.1016/j.cad.2017.05.014.