Computational Geometry: Theory and Applications 98 (2021) 101773

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Continuous flattening of all polyhedral manifolds using countably infinite creases

Zachary Abel [a], Erik D. Demaine [b,*], Martin L. Demaine [b], Jason S. Ku [a], Jayson Lynch [b], Jin-ichi Itoh [c], Chie Nara [d]

[a] MIT, EECS Department, United States of America
[b] MIT, Computer Science and Artificial Intelligence Laboratory, United States of America
[c] School of Education, Sugiyama Jogakuen University, Japan
[d] Meiji Institute for Advanced Study of Mathematical Sciences, Meiji University, Japan

ARTICLE INFO

Article history:
Received 15 March 2020
Received in revised form 31 January 2021
Accepted 10 April 2021
Available online 26 April 2021

Keywords:
Flattening
Folding
Origami

ABSTRACT

We prove that any finite polyhedral manifold in 3D can be continuously flattened into 2D while preserving intrinsic distances and avoiding crossings, answering a 19-year-old open problem, if we extend standard folding models to allow for countably infinite creases. The most general cases previously known to be continuously flattenable were convex polyhedra and semi-orthogonal polyhedra. For non-orientable manifolds, even the existence of an instantaneous flattening (flat folded state) is a new result. Our solution extends a method for flattening semi-orthogonal polyhedra: slice the polyhedron along parallel planes and flatten the polyhedral strips between consecutive planes. We adapt this approach to arbitrary nonconvex polyhedra by generalizing strip flattening to nonorthogonal corners and slicing along a countably infinite number of parallel planes, with slices densely approaching every vertex of the manifold. We also show that the area of the polyhedron that needs to support moving creases (which are necessary for closed polyhedra by the Bellows Theorem) can be made arbitrarily small.

1. Introduction

We crush polyhedra flat all the time, such as when we recycle cereal boxes or store airbags in a steering wheel. But is this actually possible without tearing or stretching the material? This problem was first posed in 2001 [7] (see [9, Chapter 18]): does every polyhedron have a continuous motion that preserves the metric (intrinsic shortest paths), avoids crossings, and ends in a flat folded state? This problem is Open Problem 18.1 of the book *Geometric Folding Algorithms* [9]. In this paper, we solve this 19-year-old open problem with a positive answer: every polyhedron can be continuously flattend. Specifically, we prove for a broad definition of *polyhedron*: any compact polyhedral 2-manifold (possibly with boundary) embedded in 3D and having finitely many polygonal faces. However, our result is arguably in a model not intended by the original problem: our folding has countably infinitely many creases at all times.

A necessary first step is to show that every polyhedron has a flat folded state (the end of the desired flattening motion). This problem was also first posed in 2001 [7], where it was solved for convex and semi-orthogonal polyhedra.[1] Later, Bern

* Corresponding author.
E-mail address: edemaine@mit.edu (E.D. Demaine).

[1] In a *semi-orthogonal* polyhedron, every facet is either parallel or perpendicular to a common plane. Thus, in some orientation, the faces are all horizontal (parallel to the floor) or vertical (perpendicular to the floor).

https://doi.org/10.1016/j.comgeo.2021.101773
0925-7721/

and Hayes [3] solved the problem for *orientable* polyhedral manifolds, generalizing a previous solution for sphere or disk topology [9]. This result solved Open Problem 18.2 of [9] (also originally posed in 2001 [7]), except for non-orientable polyhedral manifolds, which we solve here.

Continuous flattening necessarily requires continuously moving/sliding the creases on the surface over time (for polyhedra enclosing a volume): if all creases remained fixed throughout the motion (and the set of creases is finite), then the Bellows Theorem [4] tells us that the volume would remain fixed, so could not decrease to zero. A natural question, though, is how much area of the surface needs to be flexible in the sense of supporting moving creases, and how much can be made of rigid panels connected by hinges. Abel et al. [1] showed that a surprisingly small but finite slit suffices for continuous flattening of a regular tetrahedron. Matsubara and Nara [13] recently showed that an arbitrarily small area of flexibility suffices for α-trapezoidal polyhedra. In this paper, we show that an arbitrarily small area of flexibility suffices for any polyhedral manifold.

Several previous results constructed continuous flattenings of special classes of polyhedra. Itoh and Nara [10] solved Platonic solids while preserving two faces, and later with Vîlcu [11] solved convex polyhedra using Alexandrov surgery (which is difficult to compute). At SoCG 2014, Abel et al. [2] solved convex polyhedra using a simple algorithm that respects the *straight skeleton gluing*, corresponding to the intuitive way to flatten a polyhedron, and solving Open Problem 18.3 of [9] (the last open problem of Chapter 18, also originally posed in 2001 [7]). Unfortunately, this approach seems difficult to extend to nonconvex polyhedra. More recently, a slicing approach (dating back to [7]) was shown to continuously flatten semi-orthogonal polyhedra [6]. In this paper, we extend this slicing approach in several ways to solve arbitrary polyhedral manifolds.

1.1. Approach

We generalize the slicing approach of [6], which conceptually cuts the polyhedron along parallel planes through every vertex, and several additional planes in between so that the resulting *slabs* (portions of the polyhedron between consecutive planes) are "short". In [6], each slab is an orthogonal band, which is relatively easy to flatten continuously. The key difference in our case is that the slabs are much more general: in general, a slab in a polyhedron is a *prismatoid* (excluding the top and bottom faces), that is, a polyhedron whose vertices lie in two parallel planes, whose faces are triangles and trapezoids spanning both planes. Unfortunately, prismatoids seem extremely difficult to flatten continuously, as original polyhedron vertices are particularly difficult to handle in the general case.

To circumvent this challenge, we instead target the flattening of *prismoids*: prismatoids whose spanning faces are only trapezoids having parallel top and bottom edges (i.e., no triangular spanning faces), where every vertex is incident to at most two spanning trapezoids. We will use the term *cylindrical prismoid* to refer to the spanning faces of a prismoid, without the top and bottom face. A key innovation in our approach is to divide a polyhedral manifold using *countably infinitely* many parallel planar cuts, with slabs approaching zero height as we approach polyhedron vertices. As a result, *all* slabs consist of disjoint cylindrical prismoids. The key property is that original polyhedron vertices do not appear on the boundary of *any* slab, because any such slab would get divided in half through countably infinite recursion.

Note that since we allow polyhedral manifolds with boundary, a component within a slab may only be a subset of a cylindrical prismatoid, we call a *prismoidal wall*; this generalization is discussed in Section 4.

1.2. Outline

We implement the approach described above in a bottom-up fashion. First, Section 2 formally defines our model of folding. Next, Section 3 shows how to collapse prismoid edges and faces by constructing generalized In-Out and Out-Out gadgets. Then, Section 4 shows how to slice the input polyhedral manifold so that we can flatten subsets of it using the methods from Section 3. Finally, Section 5 puts these algorithms together to prove the following theorem:

Theorem 1. *Any compact polyhedral 2-manifold (possibly with boundary) embedded in 3D and having finitely many polygonal faces can be continuously flattened while preserving intrinsic distances and avoiding crossings. A flattening motion exists such that at all times during the flattening motion (except the beginning), the folded form consists of countably infinitely many creases, with finitely many accumulation lines. Furthermore, the area supporting moving creases can be made arbitrarily small.*

2. Model

The standard model of folding 2D surfaces in 3D [9, Chapter 11] assumes finitely many creases, as that is the primary case of interest for origami. A full definition supporting countably infinitely many creases is likely possible, but difficult, as it is no longer possible to focus on well-behaved positive-area neighborhoods. For the purposes of this paper, we define a limited model of folding with countably infinite creases, where the folding decomposes into components separated by horizontal planes, and each component is a finite-crease folding according to [9, Chapter 11].

Specifically, define a *stacked folded state* of a polygon P of paper to consist of two components:

1. A decomposition of P into countably many topologically closed polygonal regions $P_1, P_2, \ldots$ (the unfolded slices, each of which can be disconnected), whose interior-disjoint union $\cup_{i=1}^{\infty} P_i$ equals P. The sequence $P_1, P_2, \ldots$ can be (countably) infinite, and is in no particular order (in particular, it does not match the stacking order defined below in Property 3). Because of the infinite decomposition, some points of P belong to one or two P_i (two in the case of shared boundary), while other points of P may not belong to any P_i but rather exist in the limit of some sequence $P_{k_1}, P_{k_2}, \ldots$.
2. A finite-crease folded state (f_i, λ_i) of each region P_i (the folded slice), consisting of a geometry $f_i : P_i \to \mathbb{R}^3$ and a layer-ordering partial function $\lambda_i : P_i^2 \to \{-1, +1\}$ (as in [9, Chapter 11]).

These components must satisfy the following constraints:

3. The decomposition $P_1, P_2, \ldots$ has a total ordering $\prec$ for which each P_i intersects only its immediate predecessor and successor in $\prec$.
4. The folded states meet on their shared boundaries, i.e., $f_i(P_i \cap P_j) = f_j(P_i \cap P_j)$ for all i, j.
5. For every point $q \in P$, there is a unique point $r \in \mathbb{R}^3$ (more naturally notated $f(q)$) such that, for every sequence $q_1, q_2, \ldots$ of points in P converging to q, if each q_i belongs to a corresponding region P_{k_i}, then sequence $f_{k_i}(q_i)$ converges to r. This property guarantees a global folded-state geometry f on all of P, in particular for points that do not belong to any P_i.
6. The folded states live in interior-disjoint horizontal slices of space, i.e., all points in $f_i(P_i)$ have z coordinates in the range $Z_i = [z_i^-, z_i^+]$, where the intervals $Z_0, Z_1, Z_2, \ldots$ are interior-disjoint and $P_i \prec P_j$ implies $z_i^+ \le z_j^-$.

Intuitively, these constraints guarantee that there are no proper collisions between different folded states (f_i, λ_i) where they join. Although two folded states may touch in a shared horizontal plane, the total ordering from Property 3 provides a stacking order for such overlapping layers. A subtlety here is that, to allow the final flat folded state where $Z_1 = Z_2 = \cdots = [z, z]$ for some z, we need to allow each interval Z_i to degenerate to a point, allowing for potentially many folded states to overlap in that single z plane (without violating interior-disjointness of Property 6).

With this notion in hand, we can define *(stacked) folding motions* as in [9, Chapter 11]: a continuous function M mapping each time $t \in [0, 1]$ to a stacked folded state, where each $P_i(t)$ and $z_i(t)$ varies continuously with time, and the restriction of M to each $P_i(t)$ produces a valid folding motion of the finite-crease folded state $(f_i(t), \lambda_i(t))$. A *(stacked) flattening motion* is a (stacked) folding motion M such that the final folded state $M(1)$ lies in a single z plane.

3. Flattening prismoids

In this section, we show how to flatten prismoids which have a small height relative to their other features. We will then use this technique to flatten arbitrary polyhedral manifolds after slicing them into a countable set of such prismoids, as detailed in Section 4. Section 3.1 describes an overview of our approach, and the rest of this section describes the details of how to locally flatten the edges and faces of a prismoid.

3.1. Approach

To specify the approach in more detail, let us recall the overall approach for semi-orthogonal polyhedra from [6]. Call a prismoid edge *spanning* if its endpoints lie in the top and bottom planes, and a prismoid face *spanning* if it includes vertices in both the top and bottom planes. Unlike in the Introduction, here we include the top and bottom horizontal faces as part of the prismoid (which will later represent attachments to neighboring prismoids), and we will continuously flatten while moving the horizontal faces only vertically. The approach of [6] flattens orthogonal spanning edges of a (possibly nonconvex) prism using two methods. One method bends both faces adjacent to a spanning edge *toward* the convex side of the edge, while the other method bends one face *toward*, and one face *away*, from the convex side; we call these general strategies In-In and In-Out respectively. In each method, the top face is translated down normal to the face onto the bottom face; faces adjacent to the edge are bent using a single crease far from each spanning edge; while additional local creases are added in order to collapse each edge. This strategy allows a common interface between spanning edge collapsing crease patterns so each edge can be dealt with independently, assuming the edges are far enough apart. By alternately labeling each face around the prism as In or Out, each spanning edge can then be collapsed using their In-Out method, with the exception of perhaps one spanning edge collapsed using their In-In method; see Fig. 1.

Our edge flattening construction generalizes their orthogonal approach for nonorthogonal edges, allowing us to flatten general prismoids. There are a few key differences between the gadgets presented here and the gadgets presented in [6]. While we give a construction for a generalized In-Out gadget, we provide a construction for an Out-Out gadget, bending both faces adjacent to an edge away from the convex side, instead of an In-In gadget. While their orthogonal In-Out gadget constructs three new crease pattern vertices at any intermediate folded state to flatten each spanning edge, our generalized In-Out gadget requires construction of only two new vertices, simplifying the structure. Additionally, our Out-Out gadget has the same topological complexity as the orthogonal In-In gadget, both requiring construction of two new vertices. Lastly, both orthogonal gadgets require some adjacent faces to be coplanar and touching throughout the folding motion, which

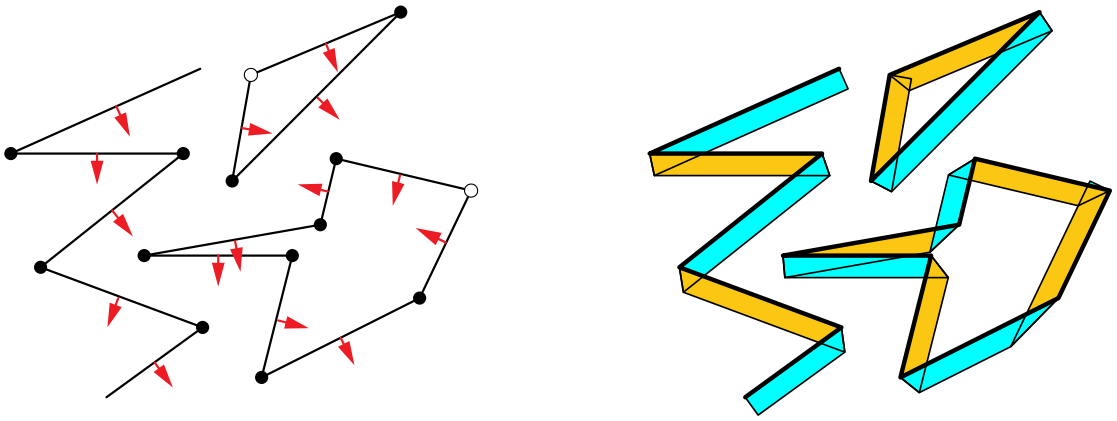

Fig. 1. [Left] Top view of a semi-orthogonal set of walls, assigning a direction to each edge and labeling non-terminating vertices as either IN-IN in white or IN-OUT in black. [Right] The flattened state associated with this direction assignment.

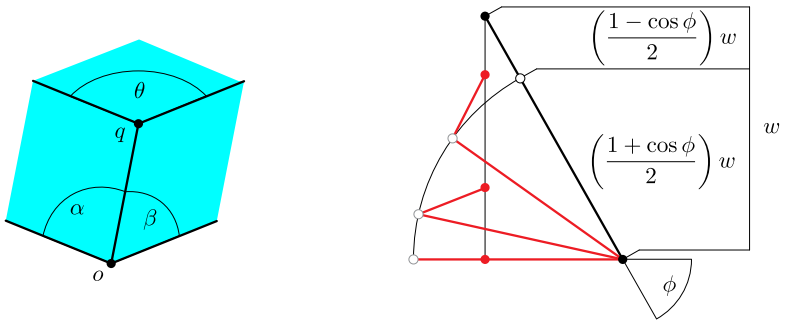

Fig. 2. [Left] Parameterization of a spanning edge up to affine transformation. [Right] Cross section of a spanning face flattening along a single crease.

may not be desirable; by contrast, faces in our generalized gadgets never touch face to face except in the final flat-folded state.

3.2. Gadget parameterization

The next three sections describe how to locally flatten spanning edges of a prismoid by detailing two gadgets: an IN-OUT gadget and an OUT-OUT gadget. Because the top and bottom faces of a prismoid must all collapse consistently and simultaneously, we give a single parameterization for the entire collapse; see Fig. 2 [Left]. Of the two prismoid vertices incident to the spanning edge, at least one has an angular deficit no greater than π. We choose such a vertex to be the *primary vertex*, and let θ, α, and β be the three face angles incident to it, with θ the angle at the base, and α and β the two angles of the incident spanning faces. When we speak locally of a spanning edge, the *primary vertex* will be vertex o, with the other *non-primary vertex* being q. By fixing the spanning edge to have unit length, we can uniquely specify any prismoid spanning edge up to affine transformations by choosing θ, α, and β such that:

- $0 < \theta$ because we forbid touching faces in the input polyhedron;
- $|\alpha - \beta| < \theta < \alpha + \beta$ or else the prismoid is already flat; and
- $\alpha + \beta \leq \pi$ as defined for a primary vertex.

3.3. Spanning face collapse

Each spanning face is angled relative to the top and bottom face of the prismoid. The dihedral angle ϕ of the face relative to the base uniquely determines the crease line that will collapse the face flat when sufficiently far from a spanning edge. We will translate the top face down normal to the face onto the bottom face; see Fig. 2 [Right] for a cross section of the face collapse. Two different single-crease solutions can allow this flattening to occur, either flattening the face to one side or the other. Call the width the distance between the top and bottom edge of the face. In either case, the crease we introduce will separate the width of the face w into sections of width

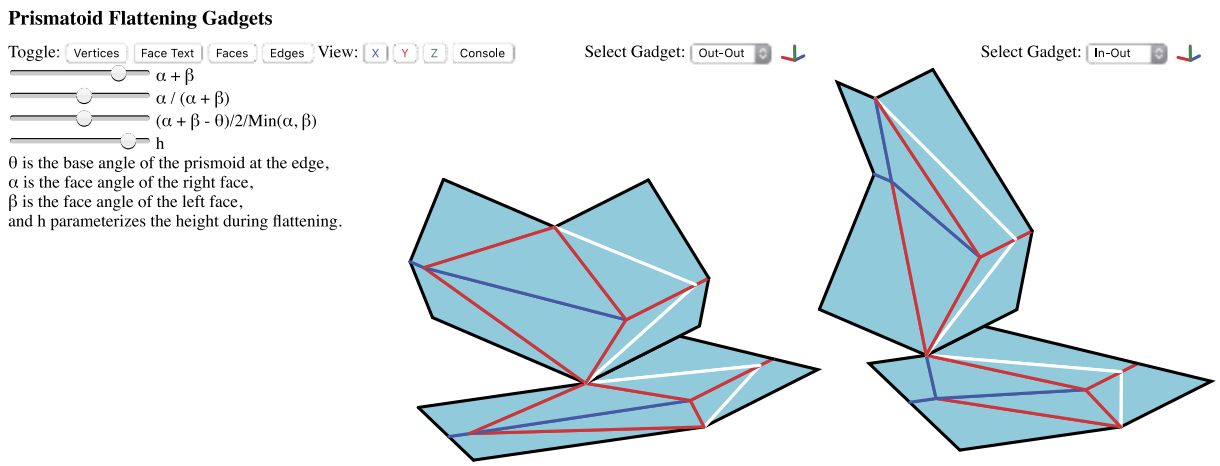

Fig. 3. View of the web application [12] for interacting with the prismoid spanning edge flattening gadgets. The Out-Out gadget is shown on the [Left] and the In-Out gadget is shown on the [Right].

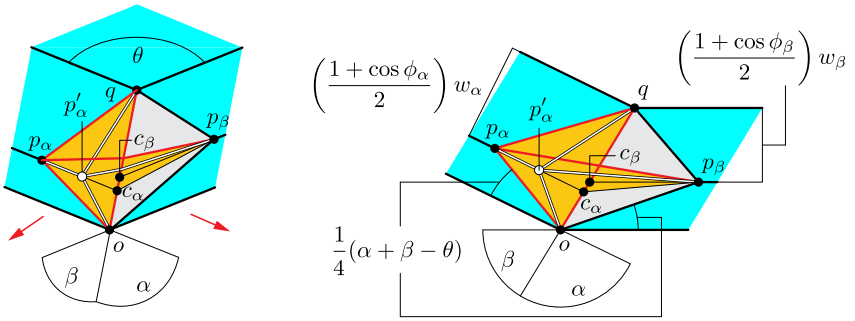

Fig. 4. Reference points and creases for the Out-Out gadget drawn on [Left] the surface of a prismoid local to a spanning edge, and [Right] the development of the spanning faces adjacent to the edge. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\left(\frac{1 \pm \cos\phi}{2}\right) w. \tag{1}$$

We call such a crease a *spanning face crease*. The crease will be closer to the bottom if the face folds toward the bottom edge, and closer to the top if the face folds toward the top edge. Local to a vertex, we can write w and ϕ on both the α and β sides of the edge in terms of our parameterization:

$$w_\alpha = \sin\alpha, \quad \cos\phi_\alpha = \csc\alpha(\cos\beta\csc\theta - \cos\alpha\cot\theta), \tag{2}$$

$$w_\beta = \sin\beta, \quad \cos\phi_\beta = \csc\beta(\cos\alpha\csc\theta - \cos\beta\cot\theta). \tag{3}$$

We note also that collapsing the face along this crease keeps folded material within distance $(1 - \cos\phi)w/2$ of the projection of the face onto the prismoid base, when ϕ and w are strictly positive. Further, it is easy to verify that during a collapse, spanning face creases always exist between the top and bottom faces (strictly between except in the final flat-folded state).

3.4. Interactive gadget visualization

In the following two sections, we describe and analyze our Out-Out and In-Out gadgets for flattening spanning edges. To supplement understanding, we have implemented a web application to visualize these gadgets over the parameterized space of possible spanning edges. You can find it here [12]. The application is written in CoffeeScript and is open source. Using the app, you can explore the different spanning edges over the parameterized space as well as intermediate folded states of the continuous folding motion. Fig. 3 shows a view of the interface and display. Toggling "Vertices" will show numeric labels for the vertices. Vertices {10, 13, 16, 17} in the animations for both the Out-Out and In-Out gadgets correspond to respective points $\{o, q, p_\beta, p_\alpha\}$ in Figs. 4 and 5.

3.5. Out-Out *gadget*

We describe the construction of our Out-Out gadget, and then show that it folds continuously while preserving intrinsic distances and avoiding crossings. We begin by constructing a flat folded state, introducing two crease pattern vertices and show that moving one of these vertices along a line provides the desired folding motion. Fig. 4 corresponds to our construction described below. Consider a prismoid spanning edge parameterized by θ, α, and β. First we construct the spanning face creases on each side according to the characterization in Section 3.3, and let c_α and c_β be the locations where respective spanning face creases meet the spanning edge. Then we can flatten the primary vertex o locally using two creases so the adjacent faces collapse away from the convex side of the edge. In order to flatten angle $\alpha+\beta$ of material with two creases while keeping the bounding edges angle θ apart, the angle between the creases must be $(\alpha+\beta+\theta)/2$. Any such creases will suffice for our construction, but choosing one that is somewhat centered will keep the gadget closer to the spanning edge. We choose the pair centered on the primary vertex o, so that the angle between a crease and its adjacent bottom edge is the same, $(\alpha+\beta-\theta)/4$. Terminate each of these creases when they meet their respective spanning face crease. Let these termination points be p_α and p_β respectively. Complete the crease pattern by adding creases along the three pairwise shortest paths between these two points and non-primary vertex q. Some tedious but straightforward algebra confirms that this crease pattern is always flat-foldable, with each vertex satisfying Kawasaki's Theorem [9].

This flat-foldable crease pattern corresponds to a folding mechanism that has a single-degree of freedom because the internal vertices are nondegenerate and degree-four. However, when this crease pattern unfolds rigidly, the *base angle* spanned by the two boundary edges incident to the primary vertex o will open monotonically from θ to $\alpha+\beta$ when fully unfolded. As a thought exercise, let us fix the crease pattern folded to some three-dimensional intermediate folded state so that the base angle is strictly between θ and $\alpha+\beta$, and then remove the two triangular faces from the crease pattern. We have removed a quadrilateral of material that was creased from p_α to p_β, i.e. in the folded state, the distance between p_α and p_β is the same as when the material is unfolded. Now we rotate the bent α and β spanning faces together around the axis from o to q until the base angle is θ, which brings points p_α and p_β closer together. What remains is a folding of a subset of the prismoid corner that matches the top and bottom face angles in an intermediate folded state. It remains to replace the hole with the quadrilateral of paper we removed. Noting that nonadjacent vertices of quadrilateral hole are now strictly contractive in this intermediate folded state, we appeal to the construction in [8] to construct an isometry. Extend the spanning face crease incident to p_α to a point p'_α whose distance to p_β is equal to the intrinsic distance along the surface, which exists by [8, Lemma 5]. Extending creases from p'_α to o, q, and p_β provides the crease pattern for this intermediate state. In fact we parameterize the continuous family of crease patterns that folds this spanning edge flat according to the location of p'_α along the segment between c_α and p_α, mapping the surface continuously to its flattened state.

Lemma 2. *The* Out-Out *gadget has bounded size and stays between the top and bottom faces during folding, while preserving intrinsic distances and avoiding crossings. Further, the area supporting moving creases is also bounded and is proportional to the square of the gadget's height.*

Proof. The Out-Out gadget has finite size; specifically, the introduced points p_α and p_β are within bounded projected distances from point o relative to θ, α, and β:

$$(p_\alpha - o)\cdot u_\alpha = \frac{1}{2}\csc\theta\cot\left(\frac{\alpha+\beta-\theta}{4}\right)(\cos(\alpha+\theta)-\cos\beta)\,, \tag{4}$$

$$(p_\beta - o)\cdot u_\beta = \frac{1}{2}\csc\theta\cot\left(\frac{\alpha+\beta-\theta}{4}\right)(\cos(\beta+\theta)-\cos\alpha)\,, \tag{5}$$

where u_α is the unit direction along the bottom edge from o adjacent to α and similarly for u_β. These distances are bounded when $0<\theta<\alpha+\beta\le\pi$ as is required. Also, points p'_α and p_β remain between the top and bottom faces because they exist on the spanning face creases; thus the entire gadget folds between the top and bottom faces.

Isometry is satisfied by construction. It remains to show that faces do not intersect. First, dihedral angles between adjacent faces in the construction are always positive except in the flat state, so local crossing does not occur between adjacent faces. Alternatively, the faces bounding the α and β spanning face creases cannot intersect each other as they will always exist on opposite sides of a plane passing through o and q, in particular any such plane that also contains a base edge. Thus, the constructed Out-Out gadget avoids crossings local to the gadget throughout the folding motion.

The area supporting moving creases is shown in yellow in Fig. 4. The area is bounded by product of the distance between o and q and $(p_\alpha - o)\cdot u_\alpha + (p_\beta - o)\cdot u_\beta$, which is proportional to the square of the gadget's height. □

3.6. In-Out *gadget*

Now we describe the construction of our In-Out gadget, and show that it also folds continuously while preserving intrinsic distances and avoiding crossings. We again construct a flat folded state, introducing two crease pattern vertices,

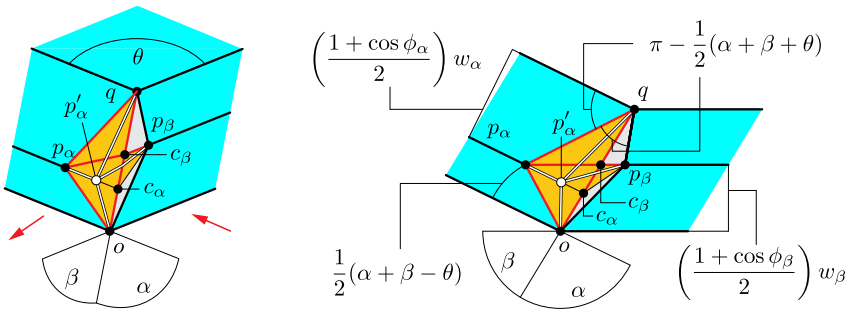

Fig. 5. Reference points and creases for the In-Out gadget drawn on [Left] the surface of a prismoid local to a spanning edge, and [Right] the development of the spanning faces adjacent to the edge.

moving one of which provides the desired folding motion. Fig. 5 corresponds to our construction described below. Again we construct the spanning face creases for a prismoid spanning edge parameterized by θ, α, and β. But this time we flatten the primary vertex o locally using only one crease so that one face collapses toward the convex side of the edge while the other face collapses away. Without loss of generality, let the α side collapse away from the convex side of the edge. We flatten the angle $\alpha+\beta$ of material with one crease while keeping the bounding edges angle θ apart, yielding a crease with angle $(\alpha+\beta-\theta)/2$ on the α side and angle $(\alpha+\beta+\theta)/2$ on the β side. We terminate the crease when it meets the α spanning face crease at point p_α. Complete the crease pattern by adding a crease from p_α to q. Again, trivial but tedious algebra confirms that this crease pattern is always flat-foldable.

Similarly to the construction of the Out-Out gadget, we would like to identify a quadrilateral of paper with contractive diagonals at intermediate folded states. However, in this case there is no obvious single choice for where to locate our stationary point p_β along the β spanning face crease. Nevertheless, we continue using the same strategy as before. Again, we have a single degree of freedom flat-foldable crease pattern whose base angle opens monotonically from θ to $\alpha+\beta$ when unfolded. Fix this crease pattern in some three-dimensional intermediate folded state so that the base angle is strictly between θ and $\alpha+\beta$. But this time, cut the folding along the segments from p_α to o and q. Now when we rotate the bent α and β spanning faces toward each other. Every point on the β spanning face crease is separated by exactly the intrinsic distance from point p_α before rotation. Additionally, point c_β and any point further from o along the β spanning face crease will be closer to point p_α after rotation. Thus choosing any point p_β along the ray from c_β would yield a quadrilateral with contractive diagonals, upon which we could apply the construction in [8] to find point p'_α along the α spanning face crease that results in an isometry. However, we cannot choose any such point. Consider for example point c_β. When θ is less than $\pi/2$, c_β can penetrate the bent α spanning face which we cannot allow. Thus we must choose some point along the β spanning face crease for which intersection does not occur.

With the Out-Out gadget, we were able to argue that the bent α and β spanning faces do not interact with each other by identifying a separating plane. We use that same strategy to pick point p_β. Let p_β be the point on the β spanning face crease such that the angle at q bounded by p_α and the top face edge on the α side equals the angle at q between p_α and p_β. This choice ensures that the faces bounding the top face edges do not overlap in the folded state, so the plane containing the top edge on the α side and point o will always separate the bent α and β spanning faces. Note that when $\pi-\beta<\theta$, this choice of p_β will lie on the α side of the line from o to q. In such cases, c_β being further away also avoids intersection, so we use it for p_β instead. Now, having fixed p_β for our spanning edge folded to some intermediate state, we have a quadrilateral hole with vertices o, p_α, q, and p_β with contractive diagonals. We again extend the spanning face crease incident to p_α to a point p'_α whose distance to p_β is equal to the intrinsic distance along the surface, which exists by [8, Lemma 5], and extending creases from p'_α to o, q, and p_β provides the crease pattern for this intermediate state. We parameterize the continuous family of crease patterns in the same way as the Out-Out gadget, by the location of p'_α along the segment between c_α and p_α, mapping the surface continuously to its flattened state.

Lemma 3. *The* In-Out *gadget has bounded size and stays between the top and bottom faces during folding, while preserving intrinsic distances and avoiding crossings. Further, the area supporting moving creases is also bounded and is proportional to the square of the gadget's height.*

Proof. The In-Out gadget has finite size; specifically, the introduced points p_α and p_β are within constant projected distances from point o relative to θ, α, and β:

$$(p_\alpha - o)\cdot u_\alpha = \frac{1}{2}\csc\theta\cot\left(\frac{\alpha+\beta-\theta}{2}\right)(\cos\beta-\cos(\alpha+\theta)), \tag{6}$$

$$\begin{aligned}(p_\beta - o)\cdot u_\beta = \frac{1}{2}\csc\theta\max[&\cot\beta\,(\cos(\beta+\theta)-\cos\alpha)\,,\\ &\cot\theta\,(\cos(\beta-\theta)-\cos\alpha)+2\cos\beta\sin\theta],\end{aligned} \tag{7}$$

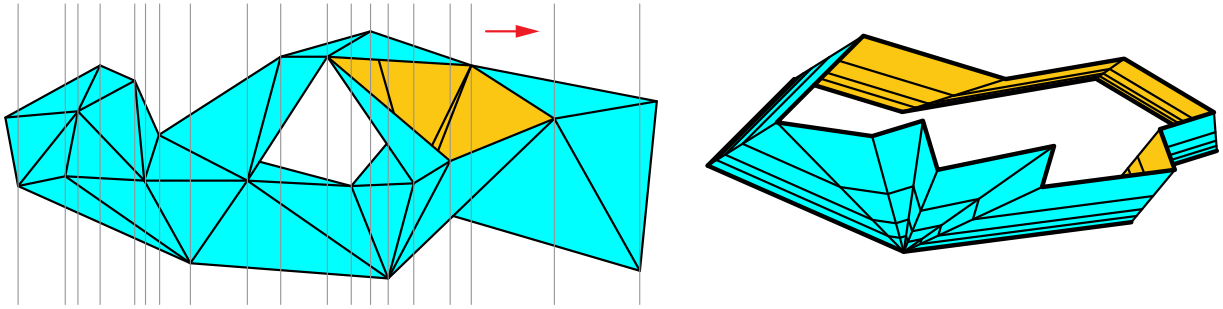

Fig. 6. [Left] Slicing a polyhedral manifold through vertices along planes normal to a direction (indicated by the red arrow) onto which vertices have unique projection. [Right] Slicing a prismatoid with one vertex that is not degree-three into an infinite set of prismoids.

where u_α is the unit direction along the bottom edge from o adjacent to and α and similarly for u_β. These distances are bounded when $0 < \theta < \alpha + \beta \leq \pi$ as is required. The remaining argument is identical to the proof of Lemma 2.

The area supporting moving creases is shown in yellow in Fig. 5. The area is bounded by product of the distance between o and q and $(p_\alpha - o) \cdot u_\alpha + (p_\beta - o) \cdot u_\beta$, which is proportional to the square of the gadget's height. □

4. Slicing

In this section, we show how to slice our polyhedral manifold into *prismoids* so techniques from the previous section can be applied. Once sliced, we can collapse the subset in each slab separately. Because we are not restricting our input to be homeomorphic to a sphere, components within a slab might not be prismoids, but instead subsets of prismoids (i.e., missing faces). To deal with this generalization, we define a *prismoidal wall* to be a (non-strict) subset of the spanning faces of some prismoid. Further, we define a *prismoidal slab* to be a finite set of disjoint prismoidal walls spanning two planes, where each such plane is a *base* of the slab.

We will slice prismoidal slabs along *slice planes*, planes parallel to the base strictly between the top and bottom of the slab. Subdividing a prismoidal slab along a slice plane results in two prismoidal slabs with smaller height than the original. Slicing will occur in multiple stages. First, we show how to break a polyhedral manifold into countably infinitely many prismoidal slabs. Next, we further subdivide each slab so the prismoidal walls in a slab have disjoint projections onto the slab's base. Finally, we slice prismoidal walls one last time in order to accommodate the local bounds for flattening spanning edges and faces using the gadgets described in Section 3.

4.1. Prismoidal slab decomposition

Lemma 4. *Any polyhedral manifold can be decomposed into a countably infinite set of prismoidal slabs.*

Proof. Orient the polyhedron so that each vertex has unique projection along some axis, which will be the case for a generic choice of axis. Then slice a plane through each vertex normal to that axis; see Fig. 6 [Left]. This division decomposes the polyhedral manifold into a set of prismatoidal slabs (not necessarily prismoidal slabs as faces adjacent to a sliced vertex may be triangles). For every non-prismoidal slab, slice along the bisecting plane between its top and bottom plane. Because each non-prismoidal slab contains exactly one vertex by construction, bisecting them yields one prismoidal slab and one non-prismoidal slab. Recursively bisecting all non-prismoidal slabs in this way will decompose the polyhedral manifold into a countably infinite set of prismoidal slabs, with slab heights approaching zero near each vertex. Fig. 6 [Right] shows this division applied to a prismatoid with one vertex that is not degree-three. □

4.2. Projection disjoint decomposition

We call a prismoidal slab to be *projection disjoint* if nonadjacent faces in the slab do not overlap in the projection of the slab onto its base.

Lemma 5. *Any prismoidal slab can be decomposed into a finite set of prismoidal slabs that are each projection disjoint.*

Proof. By definition, the prismoidal walls in a prismoidal slab are disjoint, and because each face spans the top and bottom planes, each face has finite slope. Let ψ be the smallest tilt angle of any face in the prismoidal slab, and let s be the shortest distance between any vertex and a non-adjacent edge in the same plane; see Fig. 7. Slice the prismoidal slab into a set of shorter prismoidal slabs, each with height no greater than $(s/2)\sin\psi$. Then the width of the projection of any face of the new slabs onto the base will be no larger than $s/2$. Because s is the minimum distance between a vertex and an non-adjacent edge, these prismoidal slabs must be projection disjoint. □

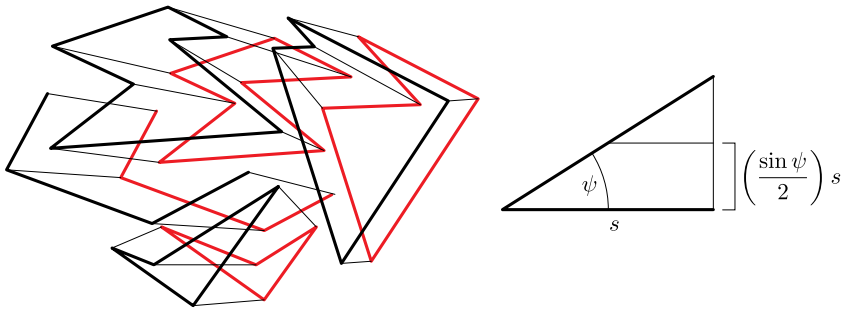

Fig. 7. [Left] A prismoidal slab that is not projection disjoint. [Right] Calculating a split height that will allow decomposition into a set of uniform height slabs that are projection disjoint. Angle ψ is the smallest angle of a prismoidal spanning face relative to the base and s is the shortest distance between a vertex and a nonadjacent edge in either the top or bottom planes.

4.3. Flattening projection disjoint prismoidal slabs

Lemma 6. *Any projection-disjoint prismoidal slab can be continuously flattened while preserving intrinsic distances and avoiding crossings. Further, the area supporting moving creases can be made arbitrarily small.*

Proof. Our approach will be to use the In-Out and Out-Out gadgets from Section 3 to collapse the prismoidal walls contained in the prismoidal slab. However, there may not be room to construct the gadgets if the spanning edges are too close together. To ensure spanning edges are well separated, we slice the prismoidal slab one last time. The proofs of Lemmas 2 and 3 show that our spanning edge gadgets are local to their spanning edge, existing within a distance proportional to the height of the gadget and Section 3.1 specifies how these gadgets can be assigned to a prismoid. Further, Section 3.3 bounds the extension of spanning faces outside their projection onto the base, again within a distance proportional to the height of the gadget. Slicing the slab in half will reduce the reach of each edge gadget and face extension in half, while the distances between them will stay fixed. Thus we can decompose the projection disjoint prismoidal slab into a finite set of prismoidal slabs that each has room to fold the gadgets at each spanning edge, and collapse each spanning face. Constructing these gadgets, we can flatten them while preserving intrinsic distances and avoiding crossing because we have sliced such that non-local interactions do not occur.

The proofs of Lemmas 2 and 3 also show that for each gadget, the area supporting moving creases is proportional to the height of the prismoidal wall squared. Thus, we can reduce this area arbitrarily by further subdivision. □

5. Flattening polyhedral manifolds

Knowing how to locally flatten prismoids using the gadgets from Section 3, and how to split polyhedral manifolds into projection-disjoint prismoidal slabs whose geometry is well-separated relative to slab height from Section 4, the proof of Theorem 1 follows directly:

Proof of Theorem 1. Slice the prismoidal manifold into a countably infinite set of prismoidal slabs using the construction in Lemma 4. Then decompose each prismoidal slab into projection disjoint prismoidal slabs with well-separated geometry according to the constructions in Lemma 5 and Lemma 6. Then by Lemma 6, we can continuously and independently flatten each slab while preserving intrinsic distances and avoiding crossings within each slab. The flattening motion of each slab brings together the top and bottom without transverse translation perpendicular to the axis, while Lemmas 2 and 3 guarantee that geometry within each slab stays between the slab's top and bottom planes during folding; so crossing cannot occur between slabs. Lastly, if we want to bound the area supporting moving creases below any positive value, Lemma 6 guarantees that we can with further subdivision. □

6. Conclusion

In this paper, we showed how to continuously flatten finite polyhedral manifolds using countably many creases with only finitely many accumulation lines. The obvious open problem is whether continuous flattening is possible with only finitely many creases at each time (still, of course, with movable creases that slide over a 2D region of points over time). In the other direction, perhaps our approach could be generalized to polyhedral manifolds with countably many vertices, edges, and faces. Such a result may require a more general model of what folding means for countably many creases (say, directly generalizing [9, Chapter 11]), which is another interesting direction for pursuit; our current model is very specific to our slice-based approach.

Our result is tight in a couple of senses. We cannot hope to flatten surfaces with a positive area of nonflat points (e.g., smooth surfaces like a sphere), even instantaneously, because a flat folded state would need to be creased at all of those

points. (A more reasonable model in this case is contractive folding; see [5].) Similarly, we cannot hope to further flatten from a plane to a line (or point) without creases becoming everywhere-dense.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Zachary Abel, Robert Connelly, Erik D. Demaine, Martin L. Demaine, Thomas C. Hull, Anna Lubiw, Tomohiro Tachi, Rigid flattening of polyhedra with slits, in: Origami^6: I. Mathematics, American Mathematical Soc., 2015, pp. 109–118.
[2] Zachary Abel, Erik D. Demaine, Martin L. Demaine, Jin-ichi Itoh, Anna Lubiw, Chie Nara, Joseph O'Rourke, Continuously flattening polyhedra using straight skeletons, in: Proceedings of the 30th Annual Symposium on Computational Geometry, 2014, p. 396.
[3] Marshall Bern, Barry Hayes, Origami embedding of piecewise-linear two-manifolds, Algorithmica 59 (1) (2011) 3–15.
[4] R. Connelly, I. Sabitov, A. Walz, The bellows conjecture, Beitr. Algebra Geom. (Contributions to Algebra and Geometry) 38 (1) (1997) 1–10.
[5] Erik D. Demaine, Martin L. Demaine, John Iacono, Stefan Langerman, Wrapping spheres with flat paper, Comput. Geom. Theory Appl. 42 (8) (2009) 748–757.
[6] Erik D. Demaine, Martin L. Demaine, Jin-Ichi Itoh, Chie Nara, Continuous flattening of orthogonal polyhedra, in: Revised Papers from the 18th Japan Conference on Discrete and Computational Geometry and Graphs, (JCDCGG 2015), Kyoto, Japan, September 14–16, in: Lecture Notes in Computer Science, vol. 9943, 2015, pp. 85–93.
[7] Erik D. Demaine, Martin L. Demaine, Anna Lubiw, Flattening polyhedra, Manuscript, 2001.
[8] Erik D. Demaine, Jason S. Ku, Filling a hole in a crease pattern: isometric mapping from prescribed boundary folding, in: Origami^6: Sixth International Meeting of Origami Science, Mathematics, and Education, 2015, pp. 177–188.
[9] Erik D. Demaine, Joseph O'Rourke, Geometric Folding Algorithms: Linkages, Origami, Polyhedra, Cambridge University Press, July 2007.
[10] Jin-ichi Itoh, Chie Nara, Continuous flattening of platonic polyhedra, in: Jin Akiyama, Jiang Bo, Mikio Kano, Xuehou Tan (Eds.), Revised Papers from the 9th International Conference on Computational Geometry, Graphs and Applications, Dalian, China, November 2010, pp. 108–121.
[11] Jin-ichi Itoh, Chie Nara, Costin Vîlcu, Continuous flattening of convex polyhedra, in: Revised Papers from the 14th Spanish Meeting on Computational Geometry, Alcalá de Henares, Spain, Springer, 2011, pp. 85–97.
[12] Jason S. Ku, Prismatoid flattening gadgets, http://jasonku.mit.edu/slicefold/.
[13] Kazuki Matsubara, Chie Nara, Continuous flattening of α-trapezoidal polyhedra, J. Inf. Process. 25 (2017) 554–558.