**Proceedings of the ASME 2013 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2013
August 4-7, 2013, Portland, Oregon, USA**

DETC2013-12692

JOINING UNFOLDINGS OF 3-D SURFACES

Cynthia Sung[*], Erik D. Demaine, Martin L. Demaine, Daniela Rus
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
Email: {crsung, edemaine, mdemaine, rus}@mit.edu

ABSTRACT
Origami-based design methods enable complex devices to be fabricated quickly in plane and then folded into their final 3-D shapes. So far, these folded structures have been designed manually. This paper presents a geometric approach to automatic composition of folded surfaces, which will allow existing designs to be combined and complex functionality to be produced with minimal human input. We show that given two surfaces in 3-D and their 2-D unfoldings, a surface consisting of the two originals joined along an arbitrary edge can always be achieved by connecting the two original unfoldings with some additional linking material, and we provide an algorithm to generate this composite unfolding. The algorithm is verified using various surfaces, as well as a walking and gripping robot design.

1 INTRODUCTION
Today's engineering designs are limited by practical considerations of their fabrication. Although recent advances in machining practices and 3-D printing technology have produced significant speedup in manufacturing of mechanical structures, these methods are still costly and time consuming when compared with planar fabrication alternatives [1]. Origami-based design methods aim to augment existing 2-D fabrication techniques with folding algorithms to allow rapid fabrication of 3-D structures. Fabricating structures in plane and then folding them into their final shape will allow complex devices to be created more quickly and efficiently, providing engineers with greater opportunities to prototype, test, and refine their designs.

Previous experiments with folded structures have demonstrated the feasibility of producing useful functionality via folding [1–4], but the design of these structures is often a long, iterative process and very application-specific. We present an algorithm that will introduce some automation into the design process. More specifically, we are interested in automatically composing multiple designs together so that the end product has the combined functionality of the originals. We take a primarily geometric approach and consider only the shape of the folded structure. Figure 1 illustrates the problem addressed. Given two folding patterns, in this case a walking robot and a gripper, our algorithm automatically generates a *composite folding pattern* for a walking robot with a gripper on one end.

This paper concerns *edge-compositions*, compositions involving two surfaces connected at one edge via hinge joint. For ease of assembly of the final product, we require that the folding pattern be one piece. In addition, for structural reasons, it is desirable for the composite folding pattern to contain the original folding patterns, rather than for it to be a new unfolding. This is because in a folded state, cut edges, edges corresponding to edges on the boundary of an unfolding that have been glued together, are mechanically weaker than folds. Cutting along an edge that will be subjected to large stresses may drastically weaken the final product. We assume that the two inputted folding patterns satisfactorily perform their intended functions, and therefore require that the composite folding pattern contain the original unfoldings in their entirety as subsets.

[*]Address all correspondence to this author.

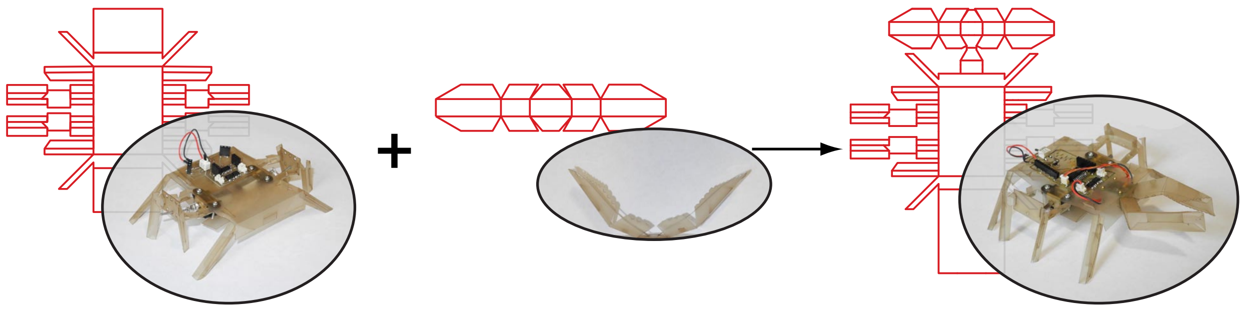


FIGURE 1. *LEFT*: WALKING AND GRIPPER ROBOTS FOLDED OUT OF THE PATTERNS SHOWN. *RIGHT*: THE *COMPOSITION*, A WALKING-GRIPPING ROBOT, WHOSE UNFOLDING WAS DESIGNED MANUALLY. OUR GOAL IS TO GENERATE SUCH AN UNFOLDING AUTOMATICALLY. *Credit: Robots were designed by Cagdas Onal and Michael Tolley.*

Related Work

The problem of finding edge-compositions of unfoldings is related to that of edge-unfolding of polyhedra, which has been studied since the 1500s [5] and has already found practical uses in such areas as sheet metal design [6]. Edge-unfolding involves flattening a polyhedron by cutting along some of its edges and unfolding the rest. The resulting planar figure should be a simple, non-overlapping polygon. The traditional problem statement for edge unfolding requires that every face be covered exactly once and that no extra material be added, leading to such results as [7] for which an unfolding does not exist. To ensure that an unfolding of an edge-composition can always be found, we allow the addition of extra material as long as it can be tucked away against an existing face.

In this sense, our problem is more similar to that of [8], where an arbitrary polyhedral surface is folded from a 2-D sheet of paper. Faces of the polyhedral surface are positioned on the 2-D plane, then excess material is tucked away to bring neighboring faces together. However, this process requires that neighboring faces on the polyhedral surface be placed adjacent in the plane. In contrast, we would like to keep our original fold patterns intact, and the faces touching the hinged edge may not always be able to be placed close to each other. Furthermore, [8] forbids cuts, requiring that the unfolding be a convex polygon, and thus may be less efficient in terms of material usage. Finally, tucks protrude perpendicularly from the final resulting polyhedral surface, although theoretically they could be crimped to arbitrarily small length. If the surfaces produced are all static, this is not a major cause of concern. However, since we would like to accommodate transformable folded surfaces, whose fold angles can change, we require a tuck that folds flat.

The idea to compose unfoldings of simple surfaces to achieve more complex ones is similar to [9], which decomposes a surface into frusta and constructs its unfolding one frustum at a time. However, this algorithm is limited in the types of surfaces that it can achieve, and, like [8], the folded surfaces are static. Edge-compositions offer greater potential in the types and the degrees of freedom of resulting folded surfaces.

Our Contributions

This paper considers edge-compositions of folded structures. Our main result is the following:

Any edge-composition of two folded surfaces has a one-piece non-self-intersecting unfolding consisting of 1) the unfoldings of the two original folded surfaces connected by 2) a bridge of linking material. (see Theorem 1)

We provide an algorithm for generating the composite unfolding and experimental evaluation across various input surfaces.

The remainder of this paper is structured as follows. Section 2 introduces necessary notation and states the problem being addressed. Section 3 gives the main insight behind generating composite fold patterns, and Section 4 provides the algorithm in greater detail. Section 5 contains the results of the algorithm for various edge-compositions. Sections 6 and 7 summarize the contributions of this work and provide directions for future study.

2 DEFINITIONS AND PROBLEM STATEMENT

A *polygon* P is a planar figure topologically equivalent to a disc and bounded by a closed non-self-intersecting path composed of a finite number of line segments. This path is called the *boundary* ∂P of P, and the area enclosed is its interior $\mathring{P}$. The line segments on the boundary are *edges*, denoted $E(P) = \{e_1^P, e_2^P, \ldots\}$, and the points where two edges meet are *vertices*, denoted $V(P) = \{v_1^P, v_2^P, \ldots\}$.

A *polyhedral complex* Q is a union of a finite set of polygons such that the intersection of any two polygons, if nonempty, is an edge or a vertex of each. The polygons making up Q are called its *faces*. The vertices and edges of Q are the vertices and edges of its faces, and are denoted by $V(Q) = \bigcup_{P \in Q} V(P)$ and $E(Q) = \bigcup_{P \in Q} E(P)$ respectively.

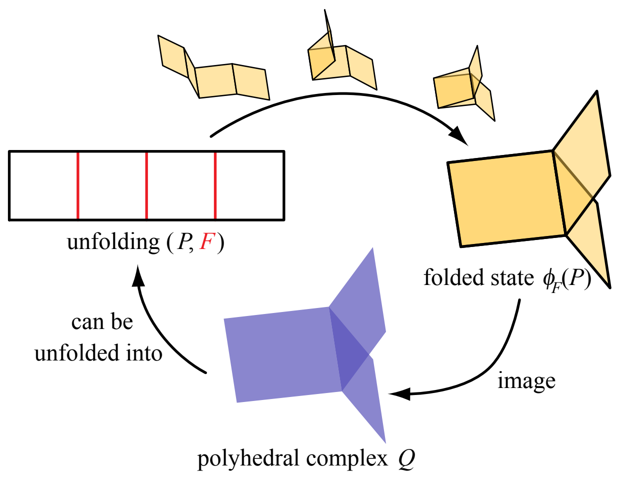


FIGURE 2. Q CAN BE *UNFOLDED* INTO P IF Q IS THE IMAGE OF A FOLDED STATE ϕ OF P

A *folded state* of a polygon P is a mapping $\phi_F : P \to \mathbb{R}^3$ with the restriction that $\phi_F : \mathring{P} \to \mathbb{R}^3$ is isometric and non-crossing. We say that a polyhedral complex Q can be *unfolded* into P if Q is the image of a folded state $\phi_F(P)$ (see Fig. 2). The folded state $\phi_F(P)$ can also be represented as the union of polygonal faces. Every edge e^ϕ of the folded state that is not on the boundary corresponds to a *fold* $f = (e^P, \theta)$, where the *fold line* e^P is the line segment on P that corresponds to e^ϕ and the *fold angle* θ is equal to the dihedral angle between the faces of $\phi_F(P)$ sharing e^ϕ (Fig. 3). We denote the set of folds $F = \{(e_1^P, \theta_1), (e_2^P, \theta_2), \ldots\}$. The polygon P and these folds together comprise an *unfolding* (P, F) of Q, and they uniquely define the folded state $\phi_F(P)$. We call the exterior of P, $\mathbb{R}^2 \setminus P$, the *free space*.

We allow the following rigid transformations of an unfolding (P, F):

- translation: P and all edges in F are translated in the plane
- rotation: P and all edges in F are rotated in the plane
- reflection: P and all edges in F are reflected in the plane. All fold angles in F are negated.

Then the problem we would like to solve is as follows.

Problem 1. *Given two polyhedral complexes Q_1 and Q_2 with unfoldings (P_1, F_1) and (P_2, F_2), and two edges $e^{Q_1} \in E(Q_1)$ and $e^{Q_2} \in E(Q_2)$, find an unfolding (P_3, F_3) such that*

1. *(P_3, F_3) is the unfolding of the union of Q_1 and Q_2 translated and rotated so that e^{Q_1} is coincident to e^{Q_2}, and*
2. *(P_3, F_3) contains translated, rotated, and/or reflected instances of (P_1, F_1) and (P_2, F_2) as subsets.*

The selected edges e^{Q_1} and e^{Q_2} act as a hinge in the combined surface. Informally, hinge joints are folds whose fold angles are not fixed but rather can take a range of values. In order for a solution to Problem 2 to make sense, the range of fold angle of this hinge must not cause Q_1 and Q_2 to collide. For the remainder of the paper, we assume that this range is nonempty.

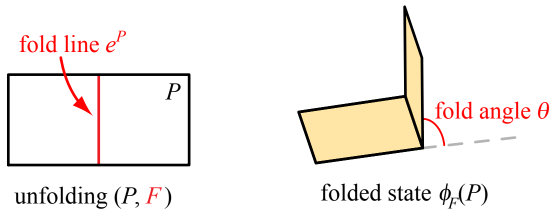


FIGURE 3. A FOLD f HAS A FOLD LINE AND A FOLD ANGLE

The edges e^{Q_1} and e^{Q_2} can be mapped to edges on the unfoldings (P_1, F_1) and (P_2, F_2). Because we allow multiple coverage of faces and edges in a folded state, it is possible for multiple edges in (P_1, F_1) to correspond to e^{Q_1}. Let E^{P_1} be the set of these edges, and similarly for E^{P_2}. By definition, if we are able to guarantee for a (P_3, F_3) that one edge in E^{P_1} will coincide with one edge in E^{P_2} in the folded state, then (P_3, F_3) will satisfy condition (1) of Problem 1. We therefore modify the problem statement slightly to concern folded states rather than their images.

Problem 2. *Given two unfoldings (P_1, F_1) and (P_2, F_2), an edge e^{P_1} in (P_1, F_1), and an edge e^{P_2} in (P_2, F_2), find an unfolding (P_3, F_3) such that*

1. *(P_3, F_3) is an unfolding of the union of translated and rotated instances of Q_1 and Q_2, the images of folded states of (P_1, F_1) and (P_2, F_2) respectively,*
2. *in the folded state, e^{P_1} and e^{P_2} coincide, and*
3. *(P_3, F_3) contains rotated, translated, and/or reflected instances of (P_1, F_1) and (P_2, F_2) as subsets.*

Solving this problem for any combination of $e^{P_1} \in E^{P_1}$ and $e^{P_2} \in E^{P_2}$ will satisfy Problem 1. The remainder of this paper is concerned with solving Problem 2.

3 EDGES ON THE CONVEX HULL BOUNDARY

In order to construct (P_3, F_3), the input unfoldings (P_1, F_1) and (P_2, F_2) must be arranged in the plane without intersection and extra material added so that the edges to be joined, e^{P_1} and e^{P_2}, are coincident in the folded state. We use the following insight.

Lemma 1. *If e^{P_1} and e^{P_2} are on the boundaries of the convex hulls of their respective unfoldings, then they may be placed coincident in the plane and will not cause (P_1, F_1) and (P_2, F_2) to intersect.*

Proof. Let $\mathrm{CH}(P_1)$ be the convex hull of P_1. By definition, $P_1 \subseteq \mathrm{CH}(P_1)$. Because $\mathrm{CH}(P_1)$ is convex and e^{P_1} is an edge

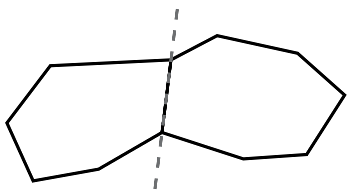


FIGURE 4. TWO CONVEX POLYGONS PLACED NEXT TO EACH OTHER ARE GUARANTEED NOT TO INTERSECT

on its boundary, $\mathrm{CH}(P_1)$ must lie entirely on one side of the line common to e^{P_1} (see Fig. 4). Similarly, $\mathrm{CH}(P_2)$ must lie entirely on one side of the line common to e^{P_2}.

Rotate, translate, and reflect (P_2, F_2) so that e^{P_2} is coincident to e^{P_1} and $\mathrm{CH}(P_2)$ is on the opposite side of e^{P_2} as $\mathrm{CH}(P_1)$. Since $\mathrm{CH}(P_1)$ and $\mathrm{CH}(P_2)$ are on opposite sides of the line now collinear to both e^{P_1} and e^{P_2}, they cannot intersect. Likewise, P_1 and P_2 cannot intersect. □

In this case, the unfolding (P_3, F_3) is simply the union of (P_1, F_1) and the transformed (P_2, F_2), with the edge e^{P_1} (also e^{P_2}) converted from a boundary edge of P_1 (resp., P_2) to a fold in F_3.

When e^{P_1} or e^{P_2} is not on the boundary of the convex hull of its unfolding, then naïvely following the above procedure may lead to self-intersection of (P_3, F_3). However, it is possible to modify (P_1, F_1) and (P_2, F_2) without changing their corresponding folded structures so that the above procedure may be used. Taking the case of (P_1, F_1), this modification would consist of constructing an additional unfolding (P^b, F^b) and attaching it to (P_1, F_1) so that in the folded state e^{P_1} becomes coincident to an edge on the boundary of the combined unfolding's convex hull. We call (P^b, F^b) a *bridge* and say that e^{P_1} has been *bridged* to the boundary of the convex hull. As we will show in Section 4,

Lemma 2. *Given an unfolding (P, F) and an edge e^P, it is always possible to bridge e^P to the boundary of the convex hull.*

Combining Lemmas 1 and 2 yields our main result.

Theorem 1. *For any unfoldings (P_1, F_1) and (P_2, F_2), and edges e^{P_1} in (P_1, F_1) and e^{P_2} in (P_2, F_2), there exists an unfolding (P_3, F_3) that satisfies Problem 2.*

Proof. According to Lemma 2, edges e^{P_1} and e^{P_2} can always be bridged to the boundaries of the convex hulls of P_1 and P_2 respectively. Let $e^{P_1}_{new}$ be the bridge edge on the boundary of the convex hull that coincides with e^{P_1} in the folded state, and similarly with $e^{P_2}_{new}$. Applying Lemma 1 to the modified (P_1, F_1) and (P_2, F_2) using $e^{P_1}_{new}$ and $e^{P_2}_{new}$ as the edges to join yields a solution to Problem 2. □

4 CONSTRUCTING THE BRIDGE

This section describes the algorithms and analysis that establish Lemma 2. There are two cases to consider:

1. e^P is on the boundary of P
2. e^P is a fold line in F

We show that in either case, it is possible to construct a bridge such that in the folded state, an edge on the boundary of the convex hull of the modified unfolding collapses onto e^P. In the course of the discussion, we make frequent use of *paths*, which we restrict to be simple polygonal chains. In so doing, it becomes possible to represent a path p of length n as a finite list of the vertices $p = p_1 p_2 \ldots p_n$ in the order they appear in the chain.

CASE 1: Edges on the Boundary

The procedure for this case is based on the following lemma.

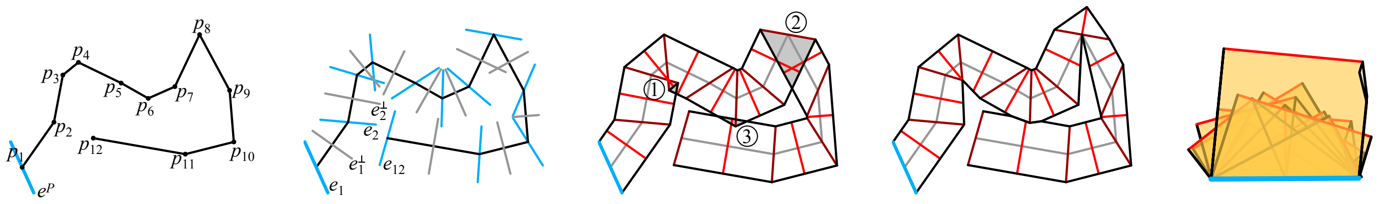
Lemma 3. *If e^P is on the boundary (i.e., $e^P \subset \partial P$), then there exists a path through the free space beginning at a point on e^P and ending on the boundary of the convex hull of P.*

Proof. Because P is a polygon, its convex hull $\mathrm{CH}(P)$ is also a polygon. The vertices of $\mathrm{CH}(P)$ are a subset of the vertices of P. If we number the vertices on the boundary $V(P) = \{v_1^P, v_2^P, \ldots, v_n^P\}$ in clockwise direction, $\mathrm{CH}(P)$ can be represented as an increasing sequence $(i_1, i_2, \ldots, i_m)$ such that $\{v_{i_1}^P, v_{i_2}^P, \ldots, v_{i_m}^P\}$ are the m vertices of $\mathrm{CH}(P)$ in clockwise order. If $e^P = (v_j^P, v_{j+1}^P)$, let P_k^{CH} be a polygon bounded by the path $p_k = v_{i_k}^P v_{i_k+1}^P \ldots v_{i_{k+1}}^P v_{i_k}^P$ where $i_k \leq j < i_{k+1}$. Since e^P is an edge of P, it is a boundary edge of P_k^{CH}. The convex hull edge $(v_{i_k}^P, v_{i_{k+1}}^P)$ is also on the boundary of P_k^{CH}. Finally, P_k^{CH} is not self-intersecting, else P would be self-intersecting or the convex hull would not completely contain P. Given these characteristics, a path from e^P to $\mathrm{CH}(P)$ that does not intersect with P or $\mathrm{CH}(P)$ except at the terminal vertices must exist inside P_k^{CH}. □

In order to bridge e^P to the boundary of the convex hull, we propose to compute such a path and overlay it with accordion-style pleats (an unfolding consisting of a sequence of non-intersecting folds with fold angles alternating between π and $-\pi$) so that the edge at the end of the path (the convex hull edge) collapses exactly onto e^P in the folded state. This is always possible by virtue of the following lemma.

Lemma 4. *Given a starting edge e^P and any simple path $p = p_1 p_2 \ldots p_{n_e}$ such that the first vertex p_1 is the midpoint of e^P, there exists a series of pleats such that every p_i lies on a fold f_i, and in the folded state, all f_i are coincident to e^P.*

Proof. We prove the existence of such a pleat structure by construction. The pleat structure we produce is based on an isosceles trapezoid. If an isosceles trapezoid is folded with a fold angle of



(a) Input (b) Lines 2–8 (c) Self-intersections (d) Line 9 (e) Folded State

FIGURE 5. ALGORITHM 1: CONSTRUCTING PLEATS TO FOLLOW A PATH. (a) THE EDGE e^P AND THE PATH p TO FOLLOW. (b) REFLECTED EDGES e_i (BLUE) AND PERPENDICULAR BISECTORS $e_i^\perp$ (GRAY). (c) RESULTING PLEATS. TYPES 1, 2, AND 3 SELF-INTERSECTIONS ARE SHADED GRAY. (d) SELF-INTERSECTIONS ARE CORRECTED. (e) FOLDED STATE OF PLEATS. ALL e_i COINCIDE.

$\pm\pi$ down its axis of symmetry, then its two legs will coincide in the folded state. In a chain of isosceles trapezoids, where every trapezoid shares a leg with at most one other, folding every trapezoid down its axis symmetry will cause the legs of all the trapezoids to coincide. Therefore, one pleat structure that satisfies the conditions of this lemma is a chain of folded isosceles trapezoids where every path vertex p_i lies on the leg of a trapezoid and e^P is also the leg of a trapezoid. The full algorithm for constructing this chain can be found in Alg. 1.

Using the perpendicular bisectors of the segments in p (line 3) ensures that the median of every trapezoidal pleat is exactly one segment of p, and that each newly created edge has as its midpoint the next vertex of p. The resulting pleats have a width of at most $\|e^P\|$. Line 9 makes the pleats non-self-intersecting. During the pleat construction in lines 2-8, three types of self-intersection may occur (see Fig. 5(c)).

1. When an edge to reflect intersects with the perpendicular bisector, the resulting pleat must be trimmed into a triangle. Note that the triangle will still contain the path vertex.
2. When a pleat overlaps with the subsequent segment of p, it results in an intersection between adjacent pleats. Let e_i be the edge shared between the two intersecting pleats, and p_i be its midpoint. The two pleats are both trimmed into non-isosceles trapezoids that meet at p_i. This operation alone would cut the bridge into two pieces. Therefore, a second set of right triangular pleats must be added in the free space next to e_i to maintain connectivity.
3. When nonadjacent segments of p are close together, their corresponding pleats may overlap. The overlap may be resolved by assigning every point in the overlapping region to the closest segment. Since the path p is simple, the pleats will remain connected and the fold lines will still contain the path vertices.

In all cases, modifications to the pleats do not prevent them from following the path p exactly or from remaining connected, so the pleats will be a valid unfolding. □

Algorithm 1: CREATEPLEATS(e^P, p)

```
Data: $e^P = (v_1^P, v_2^P)$ = starting edge
      $p = p_1 p_2 \dots p_{n_e}$ = path to follow
Result: unfolding $(P^b, F^b)$ = pleats satisfying Lemma 4

   // Beginning with $e^P$,
1  $v_{1,1} \leftarrow v_1^P$; $v_{1,2} \leftarrow v_2^P$;
   // Compute fold line locations
2  for $i = 1, \dots, n_e - 1$ do
3      $\ell_i^\perp \leftarrow$ perpendicular bisector of segment $p_i p_{i+1}$;
4      $v_{i+1,1} \leftarrow$ reflection of $v_{i,1}$ over $\ell_i^\perp$;
5      $v_{i+1,2} \leftarrow$ reflection of $v_{i,2}$ over $\ell_i^\perp$;
6      $e_i^\perp \leftarrow \left(\frac{1}{2}(v_{i,1} + v_{i+1,1}), \frac{1}{2}(v_{i,2} + v_{i+1,2})\right)$;
7      $e_{i+1} \leftarrow (v_{i+1,1}, v_{i+1,2})$;
8  end
9  Remove intersections;
   // Unfolding of pleated structure
10 $P^b \leftarrow$ polygon bounded by the path
       $v_{1,1} v_{2,1} \dots v_{n_e,1} v_{n_e,2} \dots v_{2,2} v_{1,2}$;
11 $F^b \leftarrow \{(e_1^\perp, -\pi)\} \cup \bigcup_{i=2,\dots,n_e-1} \{(e_i, \pi), (e_i^\perp, -\pi)\}$;
```

We now give the full bridge constructing algorithm (Alg. 2), illustrated in Fig. 6.

Line 1: Find a Path to the Convex Hull Boundary. According to Lemma 3, a path from e^P to the boundary of the convex hull must exist. Theoretically, any such path will suffice. Since the goal is to overlay the path with pleats, we choose a path surrounded on both sides by as much free space as possible. A path along the medial axis of P_k^{CH} satisfies this criterion.

The medial axis of P_k^{CH} is the set of all points having more than one closest point on p_k. Since p_k is the boundary of a polygon, the medial axis is a tree that partitions the polygon into regions, each of which contains one segment of p_k. The path that we use to construct the pleats is the path in that tree from the region containing e^P to the region containing the convex hull edge (Fig. 6(c)). Let $p = p_1 p_2 \dots p_{n_e}$ be this path, with the first

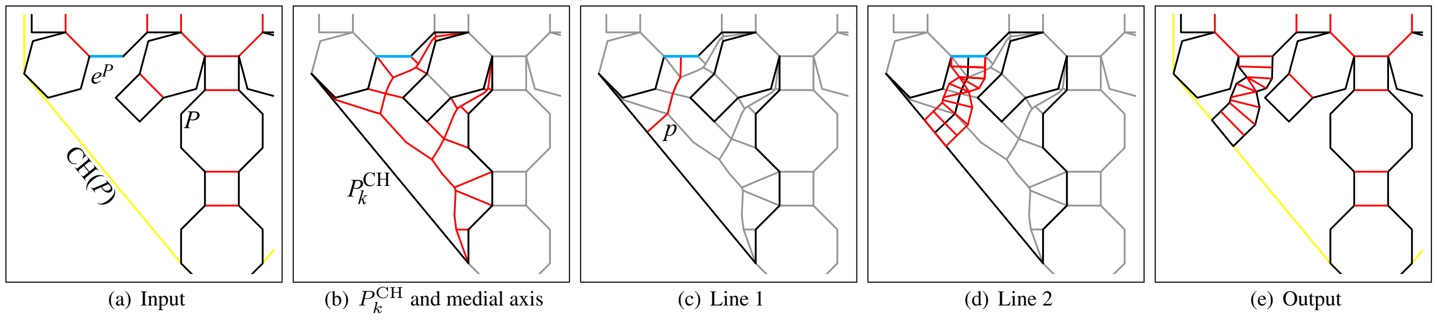


FIGURE 6. ALGORITHM 2: BRIDGING AN EDGE ON THE BOUNDARY OF THE UNFOLDING TO THE BOUNDARY OF THE CONVEX HULL. (a) ORIGINAL FOLD PATTERN. THE EDGE TO JOIN e^P IS HIGHLIGHTED IN BLUE, THE BOUNDARY IN BLACK, AND THE CONVEX HULL IN YELLOW. (b) THE REGION P_k^{CH} (BLACK) AND ITS MEDIAL AXIS (RED). (c) THE PATH p FROM e^P TO THE BOUNDARY OF THE CONVEX HULL. (d) PLEATS TILED ALONG THE PATH. (e) OUTPUT UNFOLDING WITH THE BRIDGE ADDED.

```
Algorithm 2: BridgeFromBoundary($(P,F), e^P$)
    Data: $(P,F)$ = input unfolding
          $e^P$ = boundary edge to bridge
    Result: $(P_{new}, F_{new})$ = unfolding containing $(P,F)$
            $e^P_{new}$ = edge on CH($P_{new}$) that folds onto $e^P$

1   $p \leftarrow$ path from $e^P$ to CH($P$) (Lemma 3);
2   $(P^b, F^b) \leftarrow$CreatePleats($e^P, p$);
3   $P_{new} \leftarrow P \cup P^b$; $F_{new} \leftarrow F \cup F^b \cup \{(e^P, \pi)\}$;
4   Remove intersections;
5   $e^P_{new} \leftarrow e_{n_e}$ created during bridge construction in line 2;
```

vertex p_1 located at the midpoint of e^P, intermediate vertices $p_2, \ldots, p_{n_e-1}$ at vertices in the medial axis, and the final vertex p_{n_e} on the convex hull edge.

Line 2–3: Overlay Pleats. Overlay p with accordion-style pleats using Alg. 1. For the last pleat, rather than following the procedure in lines 3–6 of Alg. 1, the point of intersection between the edge e_{n_e-1} and the convex hull edge is found. Then, e_{n_e-1} is rotated about this point of intersection onto the convex hull edge to create e_{n_e}. This guarantees that the last edge added to the pleated structure lies on the boundary of the unfolding's convex hull and that the new pleat is still an isosceles trapezoid. As in Alg. 1, e_{n_e} is then assigned a fold angle of π, and a fold line is added on the new pleat's axis of symmetry with a fold angle of $-\pi$. The last edge's exact location on the convex hull boundary is not prespecified. Since, however, p_{n_e-1} is a vertex on the medial axis bordering the region containing the convex hull edge, the median of this pleat will lie entirely inside the free space. The result of this step is a bridge that collapses flat onto the face adjacent to e^P.

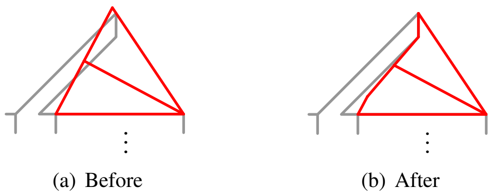


FIGURE 7. (a) A BRIDGE (RED) THAT INTERSECTS WITH (P, F) (GRAY). (b) THE OFFENDING REGION IS REMOVED.

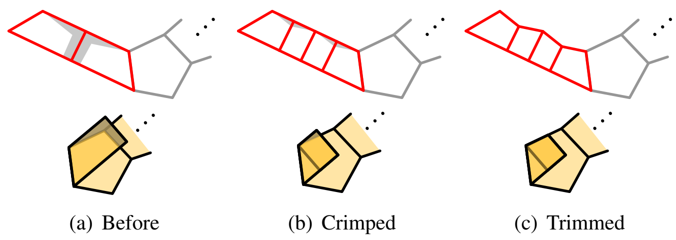
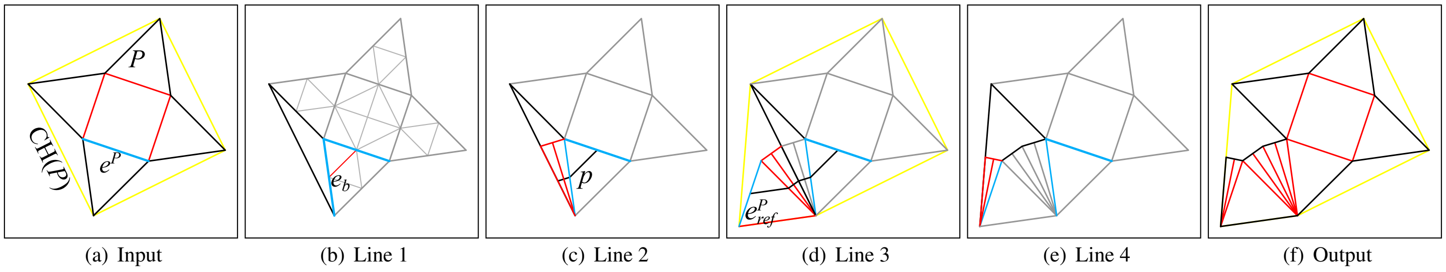


FIGURE 8. A LONG PLEAT. *TOP*: THE UNFOLDING WITH OFFENDING PLEAT IN RED. *BOTTOM*: FOLDED STATE. (a) IN THE FOLDED STATE, THE PLEAT PROTRUDES OUTSIDE THE ADJACENT FACE. (b) IT CAN BE CRIMPED TO NOT INTERFERE WITH OTHER FOLDS AND (c) TRIMMED TO AVOID PROTRUSIONS.

Line 4: Remove Intersections with the Input Unfolding. Although the path p lies entirely in the free space, the pleats following p have a width and may intersect with the input unfolding (Fig. 7). In this case, the overlapping regions can be cut out of the bridge. When this operation causes the bridge to become disconnected, then pieces that are not connected to e^P should also be removed. The bridge will still extend from e^P to the convex hull boundary since p lies entirely in the free space.

Finally, if a pleat is so long that it interferes with the folding of P, it can be trimmed or fold lines can be added to crimp the pleat arbitrarily small (Fig. 8).



(a) Input (b) Line 1 (c) Line 2 (d) Line 3 (e) Line 4 (f) Output

FIGURE 9. ALGORITHM 3: BRIDGING AN EDGE ON THE INTERIOR OF THE UNFOLDING TO THE BOUNDARY OF THE CONVEX HULL. (a) ORIGINAL FOLD PATTERN. THE EDGE TO JOIN e^P IS SHOWN IN BLUE, THE BOUNDARY IN BLACK, FOLD LINES IN RED, AND THE CONVEX HULL IN YELLOW. (b) THE EDGE-ADJACENCY GRAPH WITH THE PATH FROM e^P TO e_b HIGHLIGHTED IN RED. (c) PLEATS ATTACHED TO e_b USING ALG. 2. (d) THE ACCORDION PATH AND INTERIOR FACES REFLECTED OVER THE BOUNDARY OF THE CONVEX HULL. THE CONVEX HULL IS ALSO UPDATED. (e) PLEATS ATTACHED TO e^P_{ref} USING ALG. 2. (f) OUTPUT UNFOLDING WITH THE BRIDGE ADDED.

Algorithm 3: BRIDGEFROMFOLDLINE$((P,F),e^P)$

```
Data: $(P,F)$ = input unfolding
      $e^P$ = fold line edge to bridge
Result: $(P_{new},F_{new})$ = unfolding containing $(P,F)$
        $e^P_{new}$ = edge on CH$(P_{new})$ that folds onto $e^P$

1 $e_b \leftarrow$ a boundary edge on $P$;
2 $\{(P_2,F_2),e_{CH}\} \leftarrow$
      BRIDGEFROMBOUNDARY$((P,F),e_b)$;
3 $\{(P_3,F_3),e^P_{ref}\} \leftarrow$REFLECTFACES$((P_2,F_2),e^P,e_{CH})$;
4 $\{(P_{new},F_{new}),e^P_{new}\} \leftarrow$
      BRIDGEFROMBOUNDARY$((P_3,F_3),e^P_{ref})$;
```

CASE 2: Edges that are Fold Lines

When e^P is not on the boundary of P, then it is not adjacent to any free space and Alg. 2 alone cannot be used. Instead, we must first construct a boundary edge e^P_{ref} that in the folded state will coincide with e^P. This can be achieved by taking a path through the interior of P to an edge e_b on the boundary and reflecting the path out of P. Algorithm 3, illustrated in Fig. 9, gives the procedure for constructing a bridge for this case.

Line 1: Choose A Boundary Edge. Before constructing the bridge, it is necessary to choose where to attach it to the unfolding. Since e^P is not on the boundary of P, it is impossible to connect the bridge at e^P without causing self-intersection of the final unfolding. Any boundary edge e_b can be used here. For our implementation, we chose e_b to minimize the area of the final constructed bridge.

Line 2: Bridge e_b to the Convex Hull Boundary. Unless e_b is on the boundary of P's convex hull, the faces between e^P and e_b cannot simply be reflected over e_b to make e^P a boundary edge, since this operation may result in intersection with P. Instead, the edge e_b must first be bridged to the boundary of the convex hull using Alg. 2. Let e_{CH} be the resulting edge on the convex hull boundary.

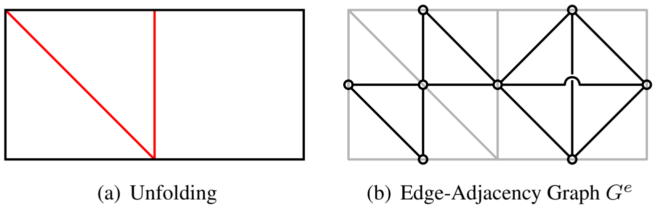
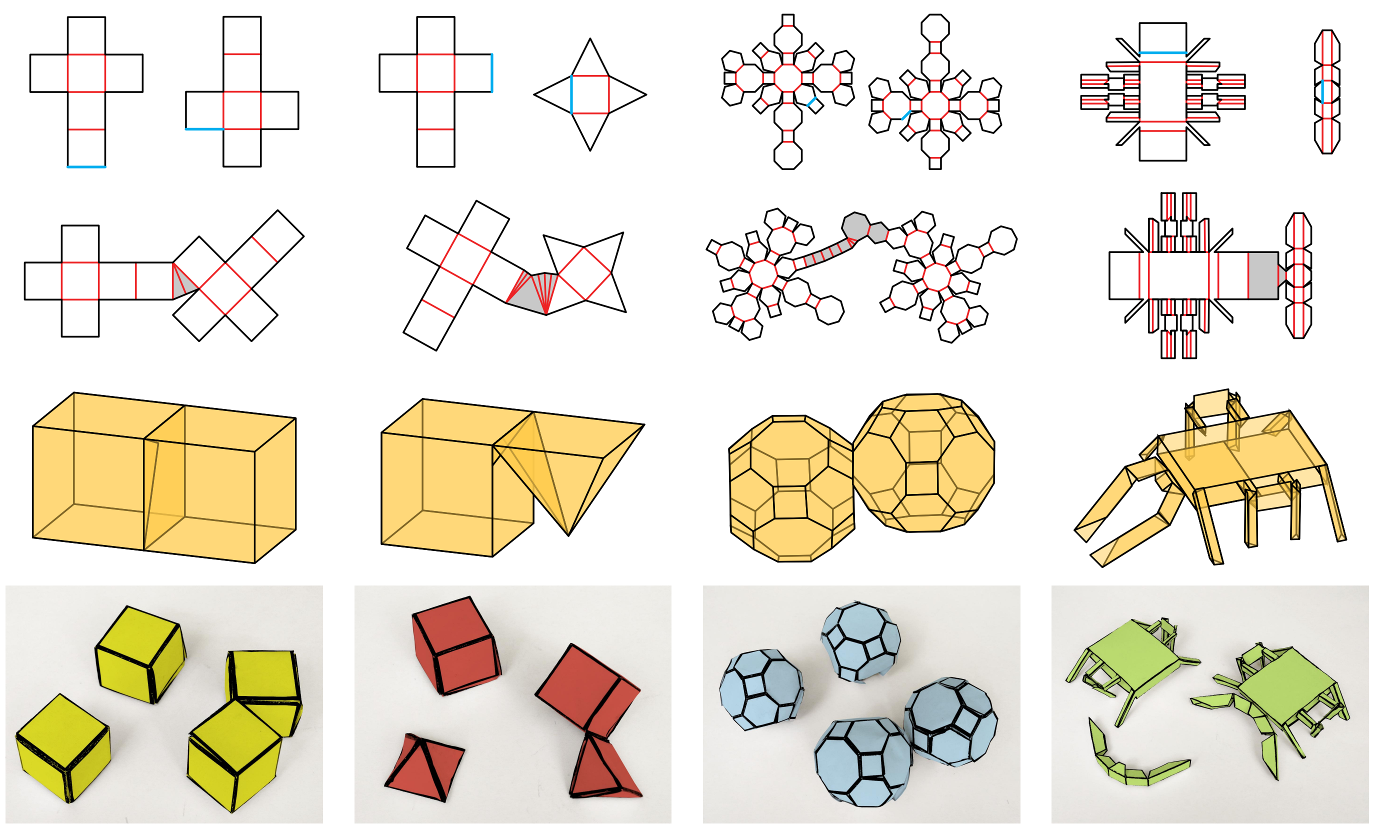

(a) Unfolding (b) Edge-Adjacency Graph G^e

FIGURE 10. EXAMPLE EDGE-ADJACENCY GRAPH

Line 3: Reflect Faces between e^P and e_{CH}. In order to construct a e^P_{ref} on the boundary of the unfolding, reflect all faces between e^P and e_{CH} over e_{CH}. These faces can be found by considering the unfolding's *edge-adjacency graph* (Fig. 10). This is a graph $G^e = (V^e, E^e)$ with vertices V^e representing the edges in P and F, and edges $E^e = \{(v^e_i, v^e_j)|e_i$ and e_j lie on the boundary of the same face in $(P,F)\}$. Every edge in E^e corresponds to a face. A path in G^e from the vertex corresponding to e^P to the vertex corresponding to e_{CH} yields a set of faces that connect e^P and e_{CH}. Since e_{CH} is on the convex hull boundary, the reflected faces will not intersect with the rest of the unfolding.

Line 4: Bridge e^P_{ref} to the Convex Hull Boundary. The result of line 3 is that e^P_{ref} is now on the boundary of P but not necessarily of the convex hull, reducing the situation to Case 1. In the folded state, the reflected path will fold flat along the surface of the input folded structure Q so that the reflected edge is coincident e^P. Thus any material attached to e^P_{ref} is as if it were added at e^P.



(a) Two cubes (b) Cube and square pyramid (c) Two truncated cuboctohedra (d) Walking and gripping robots

FIGURE 11. FOLD PATTERNS GENERATED BY THIS ALGORITHM. *TOP*: INPUT FOLD PATTERNS FOR THE POLYHEDRAL COMPLEXES TO JOIN. THE BLUE EDGES INDICATE THE EDGES TO JOIN. *SECOND ROW*: THE GENERATED COMPOSITE UNFOLDING. BRIDGES CONSTRUCTED BY OUR ALGORITHM ARE SHADED IN GRAY. *THIRD ROW*: THE FOLDED STATE OF THE COMPOSITE UNFOLDING. *BOTTOM*: PHYSICAL MODELS OF THE INPUT SURFACES AND THE COMPOSITION FOLDED FROM POSTER BOARD.

5 EXPERIMENTAL RESULTS

The proposed algorithm was implemented in MATLAB and tested on various compositions. Figure 11 shows the input unfoldings and the composition for each test. Cut lines on the boundary of the unfoldings are shown in black, and folds are shown in red. The edges to join are highlighted on the input unfoldings in blue. The constructed bridges are shaded gray. The expected folded states of each generated unfolding are simulated and verified via physical models folded from poster board.

All final folded states are, as expected, the two inputted surfaces connected along the specified edge. Since the joined edges act as a hinge joint, the angle of the input surfaces relative to each other in the folded state are not fixed. This effect can be clearly seen in the physical models, where the stiffness of the material used prevented faces from resting coincident as they do in the simulated folded states.

The constructed bridges are indeed a series of pleats connecting the edges to join to the boundaries of the convex hulls of their respective unfoldings. Figure 11(a) joining two cubes is an example of Alg. 2. Both edges to join are on the boundaries of their unfoldings. Pleats are added to the unfolding on the right since the edge to join is not already on the boundary of the convex hull. In the folded state, these pleats are flattened between the two cubes.

Figures 11(b)–(d) demonstrate Alg. 3, when an edge to join is on the interior of the unfolding. The square pyramid in Fig. 11(b) is the same one considered in Fig. 9. Not only are pleats added but a face of the unfolding is reflected in the bridge construction. In the folded state, this face lies flat against the surface of the square pyramid so that one side is doubly covered. Similarly, for the insect and gripper in Fig. 11(d), faces between the edges to join and the boundaries of their unfoldings are reflected to create the bridge. Since the boundary edges where the bridges are attached are already on the convex hulls of the unfoldings, no extra pleats are added.

6 DISCUSSION AND FUTURE WORK

This paper demonstrates that a connected composite unfolding satisfying the requirements of Problem 2 exists. Although an unfolding could be found in every case, however, it was not necessarily the most efficient unfolding in terms of material usage. For example, Fig. 11(a) shows a composition of two cubes that could be achieved simply by joining the two input unfoldings along the chosen edges. The unfolding produced by our algorithm uses extra pleats since it seeks to join unfoldings at their convex hulls.

In addition, practically, factors other than connectivity between the two input unfoldings must be taken into account. For example, when choosing the boundary edge in Alg. 3, we minimized the amount of extra material added. Other times, it may be more desirable to minimize the number of layers or to maximize the width of pleats. Each of these optimization problems yields a different choice when performing Alg. 2 or 3. For example, even if e^P is on the boundary, taking a longer path through the interior of the unfolding to an area with a greater amount of free space may yield wider pleats or fewer layers, so we may want to use Alg. 3 even if Alg. 2 is applicable. This was the approach used for the test in Fig. 11(c).

This work considers connecting two unfoldings along an edge, which is the minimum attachment necessary for two surfaces to be joined. This leads to a folded state where the two input surfaces can move relative to each other. Another type of joining is a face-joining, where the two input surfaces would be attached along one or multiple faces and would fixed relative to each other. An extension of the proposed algorithm to face-joinings is straightforward: repeat the algorithm for every boundary edge of the two connecting surfaces, choose the result that minimizes the amount of added material (or optimizes some other metric), and fix the hinge fold angles so that the two surfaces coincide. Of course, while theoretically the folded state would in this case be constrained so that the joined surfaces touch, a physical instantiation would be no stiffer than that produced when seeking a hinge joint alone. We are currently investigating alternative methods for composing unfoldings when joining occur along faces.

Finally, the results of this algorithm are restricted in that the generated unfolding must contain (P_1, F_1) and (P_2, F_2) in their entirety. When humans compose origami designs, they often rearrange the unfoldings and change the shape of the free space to achieve more efficient composite unfoldings (see, for example, Fig. 1). As discussed in Section 1, depending on the application, certain folds should not be cut; however, a 3-D surface often has several equivalent unfoldings that yield the same mechanical strength. In order for this algorithm to be useful practically, we will analyze the mechanical properties of materials that have been folded as compared to cut and develop a model that will allow us to generate equivalent unfoldings.

7 CONCLUSION

This paper addresses automatic composition of unfoldings of 3-D surfaces. We show that given the unfoldings of two 3-D surfaces, it is always possible to construct a bridge between them such that the folded state is the two originals connected along a hinge joint, and we provide an algorithm to generate the composite unfolding. The algorithm was tested on a variety of simple compositions, demonstrating that it can indeed be used to generate one-piece unfoldings of composed surfaces. The algorithm shows promise for automated design of folded structures in the future.

ACKNOWLEDGMENT

We thank Cagdas Onal for helpful discussions. This work was funded in part by NSF grants 1240383 and 1138967. C.S. was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

REFERENCES

[1] Onal, C. D., Wood, R. J., and Rus, D., 2013. "An origami-inspired approach to worm robots". *IEEE/ASME Transactions on Mechatronics,* ***18***(2), pp. 430–438.

[2] Okuzaki, H., Saido, T., Suzuki, H., Hara, Y., and Yan, H., 2008. "A biomorphic origami actuator fabricated by folding a conducting paper". *Journal of Physics: Conference Series,* ***127***(1), p. 012001.

[3] Siegel, A. C., Phillips, S. T., Dickey, M. D., Lu, N., Suo, Z., and Whitesides, G. M., 2009. "Foldable printed circuit boards on paper substrates". *Advanced Functional Materials,* ***20***(1), pp. 28–35.

[4] Hoover, A. M., Steltz, E., and Fearing, R. S., 2008. "RoACH: An autonomous 2.4g crawling hexapod robot". In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 26–33.

[5] Demaine, E. D., and O'Rourke, J., 2008. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press.

[6] Wang, C.-H., 1997. "Manufacturability-driven decomposition of sheet metal products". PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

[7] Bern, M., Demaine, E. D., Eppstein, D., Kuo, E., Mantler, A., and Snoeyink, J., 2003. "Ununfoldable polyhedra with convex faces". *Computational Geometry,* ***24***(2), pp. 51–62.

[8] Tachi, T., 2010. "Origamizing polyhedral surfaces". *IEEE Transactions on Visualization and Computer Graphics,* ***16***(2), pp. 298–311.

[9] Cheng, H. Y., and Cheong, K. H., 2012. "Designing crease patterns for polyhedra by composing right frusta". *Computer-Aided Design,* ***44***(4), pp. 331–342.