GRID VERTEX-UNFOLDING ORTHOSTACKS*

ERIK D. DEMAINE[†]
MIT Computer Science and Artificial Intelligence Laboratory,
32 Vassar St., Cambridge, MA 02139, USA
edemaine@mit.edu

JOHN IACONO[‡]
Department of Computer and Information Science, Polytechnic University,
5 MetroTech Center, Brooklyn, NY 11201, USA
http://john.poly.edu

STEFAN LANGERMAN[§]
Département d'informatique, Université Libre de Bruxelles,
ULB CP212, 50 Av. F.D. Roosevelt, 1050 Brussels, Belgium
Stefan.Langerman@ulb.ac.be

Received June 1, 2005
Revised October 20, 2006
Communicated by Godfried Toussaint

Biedl et al. [1] presented an algorithm for unfolding orthostacks into one piece without overlap by using arbitrary cuts along the surface. They conjectured that orthostacks could be unfolded using cuts that lie in a plane orthogonal to a coordinate axis and containing a vertex of the orthostack. We prove the existence of a vertex unfolding using only such cuts.

Keywords: Edge unfolding; orthogonal polyhedra; cutting; folding.

1. Introduction

A long-standing open question is whether every convex polyhedron can be *edge unfolded*—cut along some of its edges and unfolded into a single planar piece without overlap [12,11,7,10]. A related open question asks whether every polyhedron[a] (not

*A preliminary version of this paper appeared in *Revised Selected Papers from the Japan Conference on Discrete and Computational Geometry*, Tokyo, Oct. 2004, LNCS 3742, 2005, pages 76–82.

[†]Research supported in part by NSF grants CCF-0347776, OISE-0334653, and CCF-0430849, and by DOE grant DE-FG02-04ER25647.
[‡]Research supported in part by NSF grants OISE-0334653 and CCF-0430849.
[§]Chercheur qualifié du FNRS.

[a]A *polyhedron (without boundary)* is an embedded connected polyhedral complex without boundary, i.e., a connected set of polygons in Euclidean 3-space such that (1) every two polygons meet

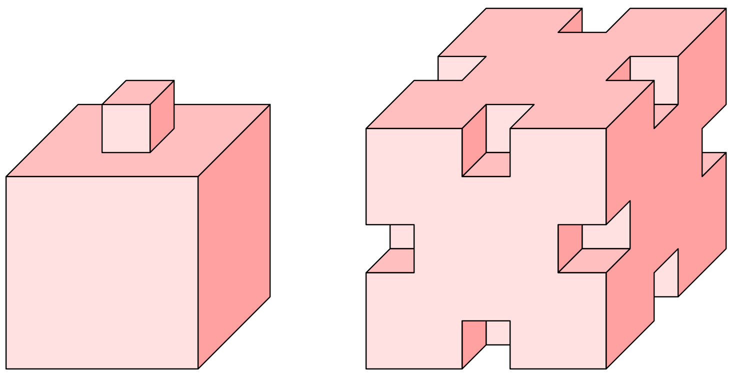

Fig. 1. These orthostacks are not edge-unfoldable [1]. The first one is also not vertex-unfoldable.

necessarily convex but forming a closed surface) can be *generally unfolded*—cut along its surface (not just along edges) and unfolded into a single planar piece without overlap. Biedl et al. [1] made partial progress on both of these problems in the context of *orthostacks*. An orthostack is an orthogonal polyhedron[b] of which every horizontal planar slice not including a horizontal face is a single simple (orthogonal) polygon. Biedl et al. showed that not all orthostacks can be edge unfolded (see Figure 1), but that all orthostacks can be generally unfolded. In their general unfoldings, all cuts are parallel to coordinate axes, but many of the cuts do not lie in coordinate planes that contain polyhedron vertices. Given the lack of pure edge unfoldings, the closest analog we can hope for with (nonconvex) orthostacks is to find *grid unfoldings* in which every cut is in a coordinate plane that contains a polyhedron vertex. In other words, a grid unfolding is an edge unfolding of the refined ("gridded") polyhedron in which we slice along every coordinate plane containing a polyhedron vertex. Biedl et al. [1] asked whether all orthostacks can be grid unfolded.

We make partial progress on this problem by showing that every orthostack can be *grid vertex-unfolded*, i.e., cut along some of the grid lines and unfolded into a vertex-connected planar piece without overlap. Vertex unfoldings were introduced in [8,9]; the difference from edge unfoldings is that faces can remain connected along single points (vertices) instead of having to be connected along whole edges. As before, a vertex unfolding must be a single planar piece without overlap. In fact, our vertex unfoldings consist of a single path of polygons, with consecutive polygons connected together at common vertices. Furthermore, as argued in [8,9], connections

at either a common vertex, a common edge, or not at all; (2) every edge is incident to exactly two polygons; and (3) every vertex is incident to exactly a topological disk of polygons, with only cyclically adjacent polygons sharing an edge. Note that a polyhedron is treated as a surface throughout this paper.

[b]An *orthogonal polyhedron* is a polyhedron (without boundary) in which every face is perpendicular to a coordinate axis. This definition implies that every face is an orthogonal polygon.

through a vertex never need to cross: for four incident faces A, B, C, D in cyclic order around a vertex v if a vertex unfolding connects A to C and B to D both via v, we can uncross the connection and keep the unfolding a single path by making different connections through v. Our unfolding places faces orthogonally into the plane: all edges of the unfolded faces are parallel to a coordinate axis. (This property is not forced by gridness in vertex unfoldings.) Our unfolding may, however, place faces so as to touch along boundary edges; we guarantee nonoverlap only of polygon interiors.

Our use of grid refinement seems to be necessary for vertex-unfolding, because the box-on-box example in Figure 1(left) has no vertex-unfolding if we are allowed to cut only along edges. It remains open whether there is such an example requiring grid cuts for a vertex-unfolding, but where every face has no holes (i.e., is homeomorphic to a disk).

Since the conference version of this paper, Damian et al. [5] generalized our techniques to grid vertex-unfold all orthogonal polyhedra of genus zero. Also, by further axis-parallel refinement of an orthogonal polyhedron beyond the grid, they have shown how to edge-unfold "orthostacks with orthogonally convex slabs" [6], "Manhattan towers" [3], "well-separated orthotrees" [2], and general orthogonal polyhedra [4]. The last case requires an exponential amount of refinement, making the two special cases of interest.

2. Grid Vertex Unfolding

Given an orthostack K, let $z_0 < z_1 < \cdots < z_n$ be the distinct z coordinates of vertices of K. Refer to Figure 2. Subdivide the faces of K by cutting along every plane perpendicular to a coordinate axis that passes through a vertex of K. This subdivision *rectangulates* K We use the term *rectangle* to refer to one element of this facial subdivision, while *face* refers to a maximal edge-connected set of coplanar rectangles. (Thus faces can have holes, but at most one in an orthostack.) We use *up* and *down* to refer to the z dimension, and use *left* and *right* to refer to the x dimension.

2.1. *Rectangle Categorization*

We partition the rectangles of K into several categories. After this categorization, the description of the unfolding layout is not difficult.

For $i = 0, 1, \ldots, n-1$, define the *i-band* to be the set of vertical rectangles (i.e., that lie in an xz plane or in a yz plane) whose z coordinates are between z_i and z_{i+1}. By the definition of rectangles, all of the rectangles of an i-band have the same extent in the z dimension, namely, $[z_i, z_{i+1}]$. By the definition of an orthostack, each i-band is connected, forming the boundary of an extruded simple orthogonal polygon.

For $i = 0, 1, \ldots, n$, we define the *i-faces* to be the faces of K in the horizontal plane $z = z_i$. As we have defined them, an i-face has several properties. It may

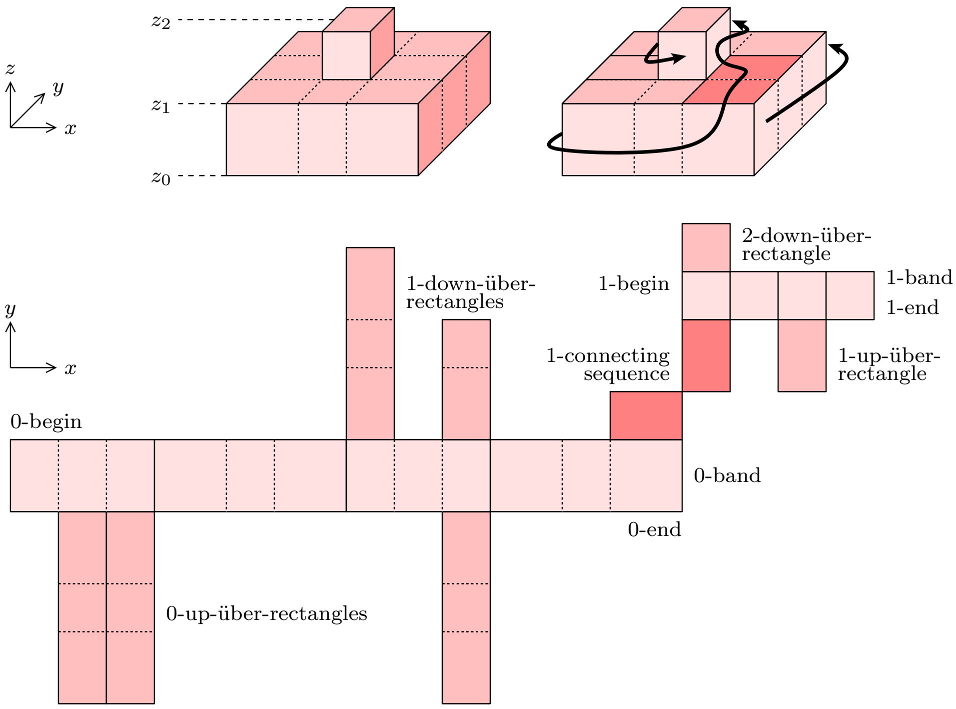

Fig. 2. Top-left: A rectangulated orthostack K with three distinct z coordinates z_0, z_1, z_2. Top-right: Categorization into i-band rectangles (light), i-über rectangles (medium), and i-connecting rectangles (dark); and the tour visiting i-band and i-connecting rectangles. Bottom: The resulting unfolding.

have the interior of K above or below it (but not both). The perimeter of the i-face (both perimeters if the i-face has a hole) has a nonempty intersection with the $(i-1)$-band, provided $i > 0$, and with the i-band, provided $i < n$. (If an i-face f is incident to only the i-band, then all edges of f must be incident to vertical faces above $z = z_i$, which form a cycle of faces in the i-band, so by connectivity of the i-band no other i-face can be incident to the i-band; also, by connectivity of the polyhedron, there cannot be another i-face meeting only the $(i-1)$-band; so f must be the bottom face of the polyhedron. Similarly, an i-face incident to only the $(i-1)$-band must be the top face of the polyhedron.)

We also need the notions of the "begin rectangle" and "end rectangle" of the i-band. Choose the 0-*band begin rectangle* to be an arbitrary rectangle of the 0-band. For $i \geq 0$, define the i-*band end rectangle* to be the rectangle of the i-band that is adjacent to the i-band begin rectangle in the clockwise direction as viewed from $+z$. For $i \geq 1$, define the i-*connecting face* to be the i-face that shares an edge with the $(i-1)$-band end rectangle, if such a face exists. Thus, the i-connecting face does not exist if and only if the $(i-1)$-band end rectangle shares an edge with the i-band. For $i \geq 1$, choose the i-*band begin rectangle* to be one of the rectangles of the i-band that shares an edge with the i-connecting face, if it exists, or else the rectangle of the i-band that shares an edge with the $(i-1)$-band end rectangle. The *i-band interior rectangles* are rectangles of the i-band that are neither the begin rectangle nor the end rectangle.

Define the *i-connecting sequence* to be an arbitrarily chosen edge-connected sequence of rectangles in the i-connecting face, if it exists, starting at the rectangle that shares an edge with the $(i-1)$-band end rectangle and ending at the rectangle that shares an edge with the i-band begin rectangle. This sequence is chosen to contain the fewest rectangles possible (a shortest path in the dual graph on the rectangles in the i-connecting face), in order to prevent the path from looping around an island and thereby isolating interior portions of the i-face. If the i-connecting face does not exist, the i-connecting sequence is the empty sequence. The rectangles in the i-connecting sequence are called *i-connecting rectangles*; all other rectangles of the i-faces are called *normal rectangles*.

We now merge all normal rectangles with their normal neighbors in the x dimension. Call the resultant rectangular regions *über-rectangles*. Thus i-faces are partitioned into the i-connecting rectangles and the i-über-rectangles. Every i-über-rectangle is connected to the perimeter of an i-face; otherwise, the rectangles that compose it could be used to construct a shorter i-connecting path. Thus, every i-über-rectangle shares an edge with either the $(i-1)$-band or the i-band (or both). Define an *i-up-über-rectangle* to be an über-rectangle that is incident to the i-band and an *i-down-über-rectangle* to be an über-rectangle that is incident to the $(i-1)$-band. If an über-rectangle is incident to both, we classify it arbitrarily.

Thus we have partitioned K into i-band begin rectangles, i-band end rectangles, i-band interior rectangles, i-up-über-rectangles, i-down-über-rectangles, and i-connecting rectangles. We now proceed to a description of the unfolding.

2.2. *Unfolding Algorithm*

Our unfolding of an orthostack consists of several components strung together at distinguished rectangles called *anchors*. Specifically, there are two types of components, i-main components and i-connecting components, both of which are anchored at two rectangles, a begin rectangle and an end rectangle. The *i-main component* consists of the entire i-band (the i-band begin rectangle, the i-band interior rectangles, and the i-band end rectangle), the $(i+1)$-down-über-rectangles, and the i-up-über-rectangles. The *i-connecting component* consists of the $(i-1)$-band end rectangle, the i-connecting rectangles (if any), and the i-band begin rectangle. It serves to connect the $(i-1)$-main component and the i-main component (at the $(i-1)$-band end rectangle and the i-band begin rectangle, respectively).

To ensure that components do not overlap each other, we enforce that the components are *anchored* in the following sense. A component is anchored at anchor rectangles R and S if, in the unfolded layout of the component, no rectangles are in the hatched region of Figure 3. More precisely, every rectangle is strictly right of

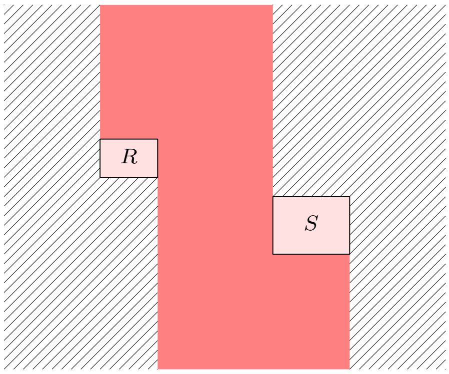

Fig. 3. A component anchored at R and S must avoid the hatched regions, remaining within the shaded region.

R and strictly left of S, or directly above R, or directly below S.

We can combine two anchored components with a common anchor while avoiding overlap. More precisely, given a component C anchored at anchors R and S, and another component C' anchored at S and T with the same orientation of S, we can combine the two unfolded layouts by translating C' so that the two copies of S coincide (with matching orientations). The conditions on the rectangles in the two components C and C' guarantee nonoverlap of the combined unfolded layout. To guarantee the matching orientations of anchors, we enforce that the positive z direction of every vertical (i-band) rectangle becomes the positive y direction in the planar unfolding.

We edge-unfold the i-main component by leaving one edge attached between the über-rectangles of the component (arbitrarily, if there is a choice), and cutting along all of the other edges of the über-rectangles. As shown in Figure 4, the layout induced by this edge unfolding consists of a central horizontal rectangular strip, which contains all i-band rectangles, and has the $(i+1)$-down-über-rectangles connected to the top of this strip, and the i-up-über-rectangles connected to the bottom of this strip. The leftmost rectangle of this strip is the i-band begin rectangle, and the rightmost rectangle of the strip is the i-band end rectangle. There is nothing below the leftmost rectangle or above the rightmost rectangle because these vacant locations are where the connecting rectangles are attached, and connecting rectangles are not über-rectangles. (In the special cases $i = 0$ and $i = n$, there can be an über-rectangle below the leftmost rectangle and above the rightmost rectangle, respectively, but in these cases, we can choose to attach the über-rectangle at its opposite edge.) Therefore the edge unfolding of the i-main component is anchored at the i-band begin and end rectangles.

We vertex-unfold the i-connecting component by a sequence of modifications to the edge-unfolding of the rectangles in the component. Let $R_0, R_1, \ldots, R_k$ denote these rectangles in connected order, where R_0 is the $(i-1)$-band end rectangle and

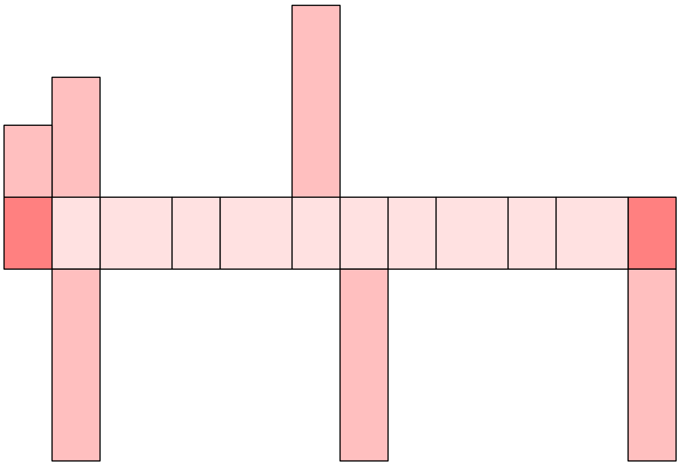

Fig. 4. An example of an unfolded i-main component. The dark rectangles are the i-band begin rectangle (left) and i-band end rectangle (right). They are connected by the remainder of the i-band (light). Above the i-band are the $(i+1)$-down-über-rectangles and below are the i-up-über-rectangles (medium). This example is a possible outcome for the 0-main component of Figure 2.

R_k is the i-band begin rectangle. The i-connecting rectangles $R_1, R_2, \ldots, R_{k-1}$ all come from an i-face, so they were planar even before the edge unfolding. The $(i-1)$-band end rectangle R_0 is adjacent to R_1 along the edge originally in the positive z direction; we rotate the edge-unfolding so that this edge is the top edge of R_0, with R_1 stacked above. Now for $2 \leq j < k$, assume that $R_0, R_1, \ldots, R_{j-1}$ have been placed, and R_{j-1} and R_j remain connected at a common edge which is not the left edge of R_{j-1}. There are three cases, depending on whether R_j shares the top, bottom, or right edge of R_{j-1}; see Figure 5. In the third case, we do nothing; in the first two cases, we vertex-unfold R_j by 90° around the right endpoint of the shared edge. After this step, R_{j+1} lies in one of the dark shaded squares, sharing R_j's top, bottom, or right edge, so the induction proceeds. We handle the i-band begin rectangle R_k differently to guarantee the proper orientation. Again there are three cases, depending on whether R_k shares the top, bottom, or right edge of R_{k-1}; see Figure 6. The shared edge corresponds the edge of R_k in the negative z direction, so in each case we vertex-unfold if necessary to make that edge the bottom edge in the unfolding. In the end, each rectangle R_j is strictly right of the previous rectangles, except R_k which might be on top of R_{k-1}. Thus, the anchored unfolding of the i-connecting component does not self-intersect.

By combining the anchored unfoldings of the 0-main component, the 1-connecting component, the 1-main component, etc., the $(n-1)$-main component, the $(n-1)$-connecting component, and the n-main component, we obtain the desired vertex unfolding:

Theorem 1. *Every orthostack can be grid vertex-unfolded.*

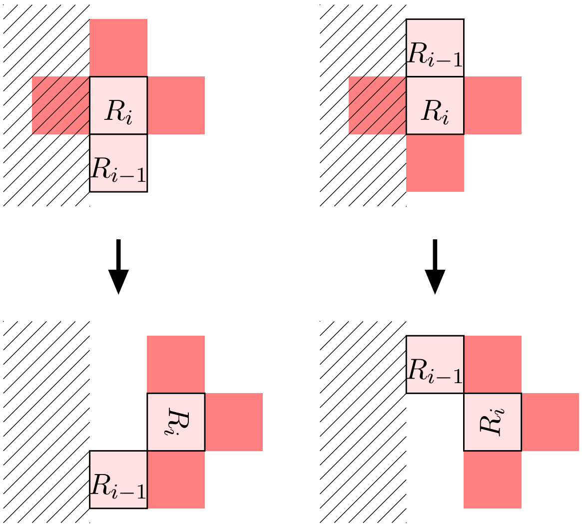

Fig. 5. How to vertex-unfold R_i after $R_0, R_1, \ldots, R_{i-1}$ have been placed (all but the last of which are in the hatched region). There are three cases, from left to right: R_i above, R_i below, and R_i to the right. In all cases, R_{i+1} is in one of the dark shaded regions, which is never left of R_i after vertex-unfolding. The illustrated unfoldings work no matter what are the sizes of the rectangles.

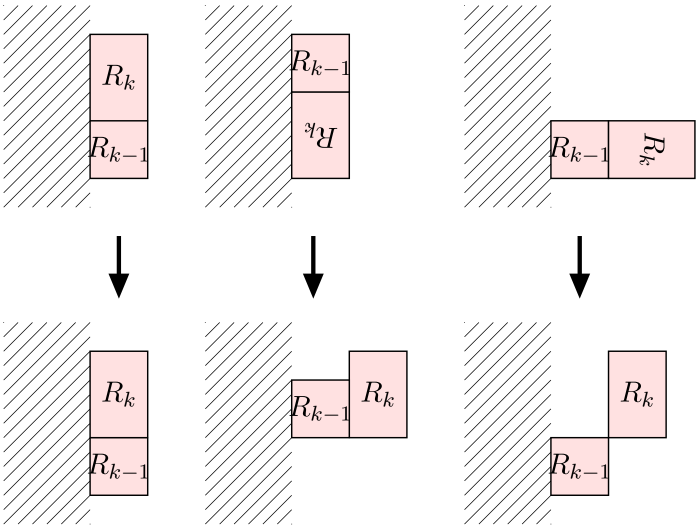

Fig. 6. How to vertex-unfold the last rectangle R_k after $R_0, R_1, \ldots, R_{k-1}$ have been placed (all but the last of which are in the hatched region). There are three cases, from left to right: R_k above, R_k below, and R_k to the right. In all cases, we must orient R_k so that the edge opposite R_{k-1} is on top. The illustrated unfoldings work no matter what are the sizes of the rectangles.

The construction leads to an algorithm whose running time is linear in the number of rectangles, which is at most quadratic in the combinatorial complexity of the polyhedron.

Acknowledgments

This work was initated while the authors visited McGill University's Computational Geometry Lab. We thank Mirela Damian and Joseph O'Rourke for helpful discussions. We also thank Koichi Hirata, Joseph O'Rourke, and an anonymous referee for helpful comments on the paper.

References

1. T. Biedl, E. Demaine, M. Demaine, A. Lubiw, M. Overmars, J. O'Rourke, S. Robbins, and S. Whitesides. Unfolding some classes of orthogonal polyhedra. In *Proceedings of the 10th Canadian Conference on Computational Geometry*, Montréal, Canada, August 1998.
2. M. Damian, R. Flatland, H. Meijer, and J. O'Rourke. Unfolding well-separated orthotrees. In *Abstracts from the 15th Annual Fall Workshop on Computational Geometry*, Philadelphia, PA, November 2005.
3. M. Damian, R. Flatland, and J. O'Rourke. Unfolding manhattan towers. In *Proceedings of the 17th Canadian Conference on Computational Geometry*, pages 211–214, Windsor, Canada, August 2005.
4. M. Damian, R. Flatland, and J. O'Rourke. Epsilon-unfolding orthogonal polyhedra. Technical Report 082, Smith College, February 2006. arXiv:cs.CG/0602095.
5. M. Damian, R. Flatland, and J. O'Rourke. Grid vertex-unfolding orthogonal polyhedra. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science*, volume 3884 of *Lecture Notes in Computer Science*, pages 264–276, Marseille, France, February 2006. arXiv:cs.3013175.
6. M. Damian and H. Meijer. Edge-unfolding orthostacks with orthogonally convex slabs. In *Abstracts from the 14th Annual Fall Workshop on Computational Geometry*, pages 20–21, Cambridge, MA, November 2004.
7. E. D. Demaine. Folding and unfolding linkages, paper, and polyhedra. In *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 2098 of *Lecture Notes in Computer Science*, pages 113–124, Tokyo, Japan, November 2000.
8. E. D. Demaine, D. Eppstein, J. Erickson, G. W. Hart, and J. O'Rourke. Vertex-unfolding of simplicial manifolds. In *Proceedings of the 18th Annual ACM Symposium on Computational Geometry*, pages 237–243, Barcelona, Spain, June 2002.
9. E. D. Demaine, D. Eppstein, J. Erickson, G. W. Hart, and J. O'Rourke. Vertex-unfolding of simplicial manifolds. In *Discrete Geometry: In Honor of W. Kuperberg's 60th Birthday*, pages 215–228. Marcer Dekker Inc., 2003.
10. E. D. Demaine and J. O'Rourke. A survey of folding and unfolding in computational geometry. In J. E. Goodman, J. Pach, and E. Welzl, editors, *Discrete and Computational Geometry*, Mathematical Sciences Research Institute Publications. Cambridge University Press, 2005. To appear.
11. J. O'Rourke. Folding and unfolding in computational geometry. In *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 1763 of *Lecture Notes in Computer Science*, pages 258–266, Tokyo, Japan, December 1998.

12. G. C. Shephard. Convex polytopes with convex nets. *Mathematical Proceedings of the Cambridge Philosophical Society*, 78:389–403, 1975.