Unfolding Orthogonal Polyhedra

Joseph O'Rourke

Abstract. Recent progress is described on the unsolved problem of unfolding the surface of an orthogonal polyhedron to a single non-overlapping planar piece by cutting edges of the polyhedron. Although this is in general not possible, partitioning the faces into the natural vertex-grid may render it always achievable. Advances that have been made on various weakenings of this central problem are summarized here.

1. Introduction

Two unfolding problems have remained unsolved for many years [**DO05b, DO07**]: (1) Can every convex polyhedron be edge-unfolded? (2) Can every polyhedron be unfolded? An *unfolding* of a 3D object is an isometric mapping[1] of its surface to a single, connected planar piece, the "net" for the object, that avoids overlap. An *edge-unfolding* achieves the unfolding by cutting edges of a polyhedron, whereas a *general unfolding* places no restriction on the cuts. A net representation of a polyhedron finds use in a variety of applications [**O'R00**]. For example, in manufacturing parts from sheet metal [**Wan97**], a 3D part is approximated as a polyhedron and its surface is mapped to a collection of 2D flat patterns. Each pattern is cut from a sheet of metal and folded by a bending machine [**KBGK98**], and the resulting pieces assembled to form the final part.

It is known that some nonconvex polyhedra cannot be edge-unfolded without overlap. However, no example is known of a nonconvex polyhedron that cannot be unfolded with unrestricted cuts. Advances on these problems have been made by specializing the class of polyhedra, or easing the stringency of the unfolding criteria. On one hand, it was established in [**BDD$^+$98**] that certain subclasses of *orthogonal polyhedra*—those whose faces meet at angles that are multiples of 90°, and whose edges are parallel to Cartesian axes—have an unfolding. In particular, the class of *orthostacks*, stacks of extruded orthogonal polygons, was proved to have a general unfolding. On the other hand, loosening the criteria of what constitutes a net to permit connection through points/vertices, the so-called *vertex-unfoldings*, led to an algorithm to vertex-unfold any triangulated manifold [**DEE$^+$03**] (and indeed,

1991 *Mathematics Subject Classification.* Primary 52B10; Secondary 52B45, 52B70, 68W05.
Key words and phrases. polyhedra, orthogonal polyhedra, unfolding.
Supported by NSF award DUE-0123154.

[1]An isometric mapping preserves distances as measured within the surface.

any simplicial manifold in higher dimensions). A vertex unfolding maps the surface to a single, connected piece P in the plane, but P may have "cut vertices" whose removal disconnects P. Fig. 1(e) shows a vertex unfolding of the polyhedron in (a).

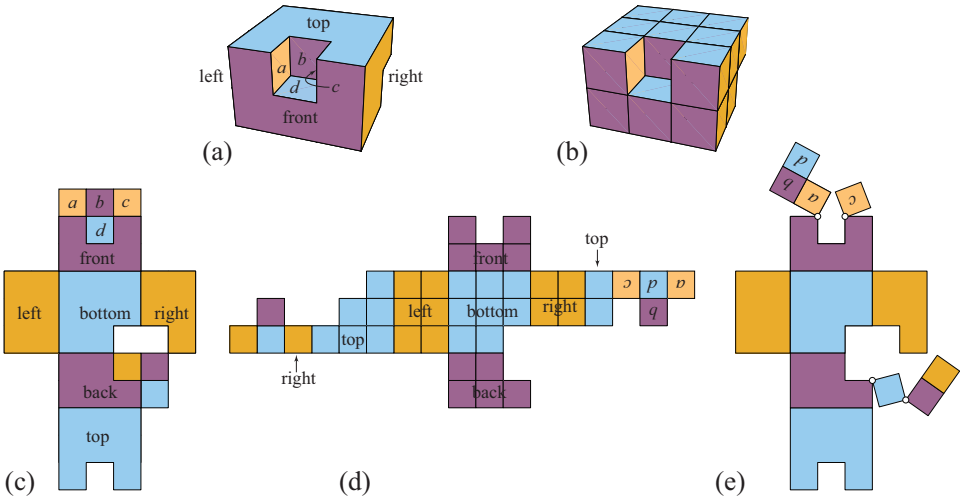

FIGURE 1. (a) A 3×3×2 solid block with two unit cubes removed, front-top center, and back-bottom right. (b) The same shape gridded by coordinate planes through every vertex. (c) An edge unfolding, with touching cuts but no interior overlap. (d) A grid-edge unfolding of (b) with strict non-overlap. (e) A vertex unfolding of (a), with rotations about circled vertices.

Here we survey the known results on unfolding orthogonal polyhedra, a shape class close to those shapes often encountered in manufacturing applications (e.g., see [**TLT04, LT07**]). We explore both general unfoldings and vertex unfoldings. Although edge unfoldings are not sufficient for orthogonal polyhedra, a second loosening of the unfolding criteria is especially natural for orthogonal polyhedra. A *grid unfolding* adds edges (which we will call *grid edges*) to the surface by intersecting the polyhedron with planes parallel to Cartesian coordinate planes through every vertex; see Fig. 1(b). For orthogonal polyhedra, a grid-edge unfolding is a natural median between edge-unfoldings and unrestricted unfoldings. However, even finding a grid-edge unfolding of orthogonal polyhedra is unsolved, so attention has turned to grid refinements. A $k_1 \times k_2$ *refinement* of a surface [**DO05a**] partitions each face into a $k_1 \times k_2$ grid of faces (with the convention that a 1×1 refinement is an unrefined grid unfolding).

Before detailing the results, three further scope distinctions are needed. First, we will confine our attention to orthogonal polyhedra without holes, i.e., genus-zero polyhedra; henceforth the term "orthogonal polyhedron" should be read with this restriction. So little is known that it is premature to venture beyond genus-zero.[2] Second, in general a non-overlapping unfolding permits the boundary of the unfolding to touch as long as no interior points overlap. This corresponds to the physical

[2]The algorithm for orthotubes (Table 1(5)), however, does work for genus 1, simply by cutting the tube cycle.

model of cutting out the net from a sheet of material, with perhaps some cuts touching but not properly crossing other cuts. In general we will only require interior non-overlap, so that the net is a *weakly simple polygon*, as in Fig. 1(c). However, some early work insists on strict non-overlap, so that the unfolding is a *simple polygon*, as in Fig. 1(d), as we will mention later. Third, we focus entirely on finding a "final flat state" rather than the continuous unfolding motion achieving that state. Indeed an example was constructed in [**BLS05**] that has a non-overlapping edge unfolding but cannot reach that flat state without self-intersection if the faces are treated as rigid plates.

2. Results and Status

Before describing the somewhat confusing partial results, we should state at the outset that the goal of proving or disproving that every orthogonal polyhedron has a (1×1) grid-edge unfolding remains unsolved. Indeed, even a slightly finer gridding leaves matters unsolved. Define a *polycube* as an orthogonal polyhedron formed by gluing identical (unit) cubes face-to-face. Polycubes are 3D generalizations of polyominoes, whose edges available for cutting are a (sometimes proper) superset of the grid edges. See Fig. 1(b). Indeed, one way of phrasing the question is this (posed with George Hart in 2004):[3]

> Is there any polycube P that cannot be edge-unfolded, where all cube edges on the surface of P are considered edges available for cutting?

Despite this question and the (1×1) goal being unresolved, a number of specialized results have been achieved, as summarized in Table 1. We describe these results briefly in Sec. 3 before turning to the algorithmic techniques employed.

#	*Polyhedra*	*Edge/Vertex*	*Refinement*	*Reference*
(1)	orthostacks	grid-edge	2×1	[**BDD+98**]
(2)		grid-edge	1×1	—*Open*—
(3)		grid-*vertex*	1×1	[**DIL05**]
(4)	o-convex orthostacks	grid-edge	1×1	[**DM04**]
(5)	orthotubes	grid-edge	1×1	[**BDD+98**]
(6)	orthotrees	grid-edge	$O(1)\times O(1)$	—*Open*—
(7)	well-sep. orthotrees	grid-edge	1×1	[**DFMO05**]
(8)	Manhattan towers	grid-edge	5×4	[**DFO05**]
(9)		grid-edge	1×1	—*Open*—
(10)	general	grid-*vertex*	1×1	[**DFO06**]
(11)		grid-edge	$2^{O(n)}\times 2^{O(n)}$	[**DFO07**]
(12)		grid-edge	$O(1)\times O(1)$	—*Open*—

TABLE 1. Orthogonal polyhedra unfolding results. Abbreviations: "o-convex" = orthogonally convex; "well-sep." = well-separated.

In the table, *grid edge* and *grid vertex* refer to edges/vertices of the refined grid (as opposed to those of the originating polyhedron). Various restricted classes of shapes have been explored to give insight into "general" (unrestricted) orthogonal polyhedra. An *orthostack* P is a stack of extruded orthogonal polygons, with

[3]See also `http://cs.smith.edu/~orourke/TOPP/P64.html#Problem.64`.

each pair of adjacent slabs of the stack intersecting in an orthogonal polygon. An equivalent definition is that every intersection of the solid bound by P with a plane orthogonal to the z-axis is either empty, or a single orthogonal polygon without holes. (See Fig. 2.) Orthostacks were introduced and studied in [**BDD$^+$98**], which presented an algorithm that in the notation above achieves a (2×1) grid-edge unfolding (Line (1) of Table 1). This algorithm includes several key ideas that have found continued use. Although improving this to a (1×1) algorithm remains open (2) (a much restricted version of the main unsolved problem), two even more special cases have been resolved. First, the novel idea of exploring grid-*vertex* unfoldings led to a (1×1) algorithm (3) in [**DIL05**], an algorithm we will discuss further below in Sec. 3.1. Second, restricting the shape of each slab of the orthostack to be orthogonally convex (meeting each horizontal and vertical line in at most one component) led to a (1×1) grid-edge algorithm (4).

A second achievement of the early work [**BDD$^+$98**] was a (1×1) grid-edge algorithm for orthotubes (5). An *orthotube* is composed of rectangular bricks glued face-to-face to form a path or a cycle, with a restriction on the gluing to ensure "clean" corner turns that we will not detail.

These two classes of shapes—orthostacks and orthotubes—share a certain linear structure that facilitated the non-overlapping layout. The remainder of the work listed in Table 1 is aimed at more complex "nonlinear" structures. *Orthotrees* are similar to orthotubes except with dual a tree rather than a path, and already no $O(1)\times O(1)$ refinement algorithm is known (6). Only a special case we will not define, "well-separated orthotrees," has been settled (7). In another direction, a class of shapes with a more controlled tree structure has led to results. A *Manhattan tower* is an orthogonal polyhedron P with an orthogonal polygon base, and such that, for $h_1 < h_2$, the intersection of the solid bound by P with plane $z = h_2$ is nested in (i.e., a subset of) the intersection with $z = h_1$. Thus P can be viewed as a collection of narrowing vertical towers, a natural generalization of a "Manhattan skyline" polygon to 3D. Manhattan towers are monotone terrains in the sense that they meet every z-vertical line in at most one component. For these shapes, a (5×4) algorithm (8) is available [**DFO05**], but (1×1) remains open (9).

Most recently, two results on general polyhedra have been established: a (1×1) grid-*vertex* algorithm (10) [**DFO06**], and a grid-edge algorithm (11) that needs to refine faces with a grid beyond any constant density [**DFO07**]. This latter algorithm is in a sense, then, one that makes arbitrary cuts, but always parallel to polyhedron edges. For general polyhedra, not only is (1×1) open, even a constant $O(1)\times O(1)$ grid-edge algorithm is unavailable (12) at this writing.

3. Algorithmic Techniques

Rather than repeat the algorithmic descriptions contained in the original papers, we highlight here instead the central algorithmic techniques or themes: staircases, recursive staircases, helical peels, and nested helical peels.

3.1. Staircases. The first paper specifically on unfolding orthogonal polyhedra was, as far as we know, that of Biedl et al. [**BDD$^+$98**]. This paper established that orthogonal polyhedra, and even orthostacks, cannot always be edge-unfolded, and so set the stage for refined grid unfoldings. The orthostack algorithm in particular established several techniques employed by all subsequent algorithms. Orient an orthostack polyhedron P so the slabs (extruded polygons) are stacked in the

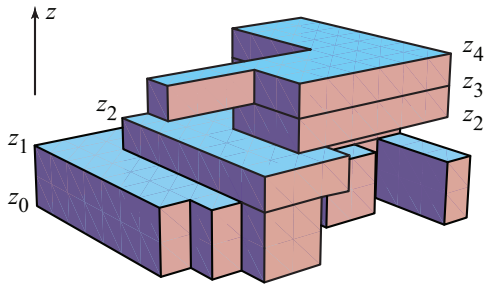

FIGURE 2. An orthostack of four extruded orthogonal polygons, the first extruded from $z = z_0$ to $z = z_1$, etc.

z-direction, as in Fig. 2. Call a face orthogonal to the x-axis an *x-face*, and similarly for y- and z-faces. Each slab is bounded by a *band* B_i composed of x- and y-faces (and so parallel to the z-axis). The algorithm unfolds each band B_i to a horizontal strip, and connects this strip to that for the next band up, B_{i+1}, via a "connecting bridge" C_i^*. These strips form a staircase monotone with respect to the horizontal x. However, to avoid overlap, the algorithm splits each B_i into two halves, L_i and R_i, which are (perhaps) staggered horizontally in the layout. Fig. 3 shows a schematic layout of the unfolding. The z-faces, both front (facing a viewer at $z = -\infty$) and back, are labeled D_i, and will be placed into the regions so marked in the figure. The connecting bridges C_i^* are subpieces of D_i.

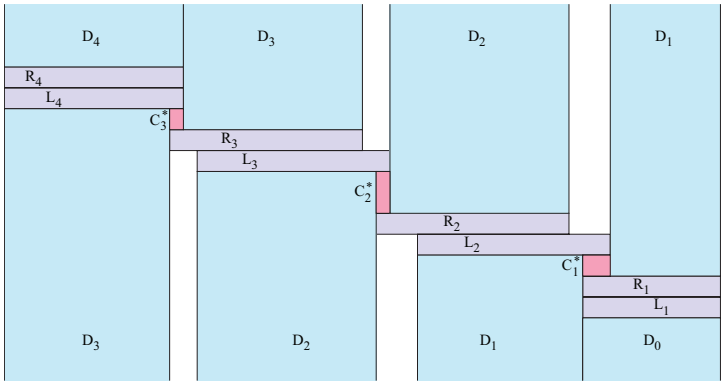

FIGURE 3. Bands $B_i = L_i \cup R_i$ unfold to a staircase. Based on Fig. 3 in [**BDD+98**].

The z-faces D_i are partitioned into pieces P_j, and arranged within the free regions above and below the staircase, as depicted in Fig. 4.

As mentioned (Table 1(1)), this algorithm achieves a (2×1) grid-edge unfolding of orthostacks, indeed a strictly nonoverlapping unfolding. However, the splitting of the bands B_i that is responsible for the "2" in the "(2×1)" is essential to avoid overlap in this algorithm, and no (1×1) grid-edge algorithm is known (Table 1(2)).

In [**DIL05**] this result was improved to a (1×1) unfolding, but using vertex attachments at key junctures. The overall design of the algorithm is the same, resulting in a staircase for the (now unsplit) bands, with pieces of the z-faces arranged

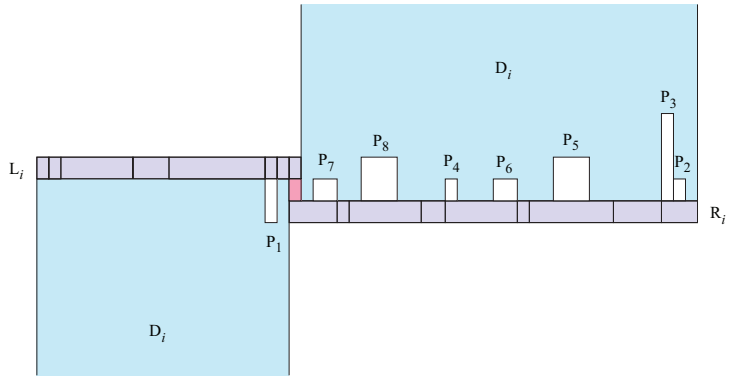

FIGURE 4. The rectangles P_j assemble to form the z-face D_i. Based on Fig. 8 in [**BDD+98**].

above and below. But the freedom to rotate a piece around a vertex permits the layout to avoid overlap without refinement.

3.2. Recursive Staircases. As mentioned, orthostacks lend themselves to the techniques just described because of the linear structure resulting from the stacking. To go beyond this requires a new technique. An idea introduced in [**DFO05**] is best described through the later and more complex work [**DFO06**], which extended the grid-vertex unfolding idea just described to handle general polyhedra. The basic technique is to continue with a staircase built from unfolded bands, connected by bridges, with above-and-below placements of other nonband faces (as in Fig. 3), but to follow a recursive tree structure rather than the path induced by a stacking. This requires completing the unfolding of the parts of the polyhedron represented by a subtree once that subtree is entered by the algorithm.

Fig. 5(a) shows an example, with the bands composed of x- and z-faces (and so parallel to the y-axis) in this different orientation. Although the example is simple, the techniques extend to more complex polyhedra. Here we emphasize the recursive structure leading to the staircase depicted in Fig. 5(b). The unfolding of band A is interrupted at grid face a_0 to access band B, which is recursively unfolded before resuming work on A. The "return path" k_1 encounters band C, which is then unfolded, interrupted by C's child D, whose unfolding is completed before returning to C. The unfolding of C proceeds to face c_4, at which point it is again interrupted to unfold child band G, and then H. Finally the unfolding of C is completed, returning back to A's grid face a_1 via k_8 and k_9. So the unfoldings of D, and G/H, are embedded within the twice-interrupted unfolding of C, while the unfoldings of B/C occur within the unfolding of A. Note the several vertex hinges $x_b, \ldots, x_h, u_3$ at which the freedom of vertex unfolding is exploited. Vertical y-faces are later "hung" above and below the horizontal staircase (according to the "illumination" arrows in Fig. 5(a)) much as before. This algorithm results in substantial cut-edge overlap in the unfolding.

3.3. Helical Peels. Without the hinging afforded by vertex unfolding, maintaining the recursive staircase structure with grid-edge unfolding requires an additional idea. The problem encountered is that, without hinging, it seems impossible

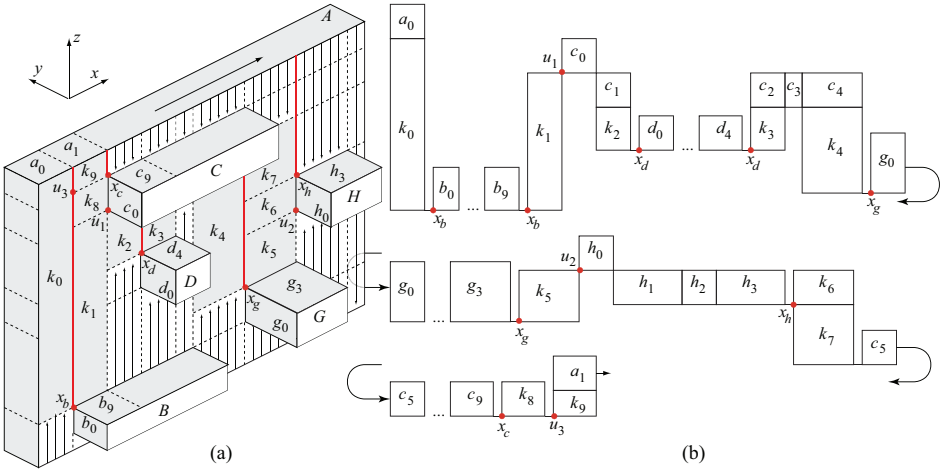

FIGURE 5. (a) An orthogonal polyhedron composed of bands A, B, C, D, G, H. (b) The vertex-unfolding, with elisons indicated by "..." The vertical "illuminated" faces marked by parallel arrows in (a) are not shown hung above and below the staircase in (b). Rotations about marked x_i vertices are shown separated from their left neighboring faces for clarity. Based on Fig. 12 in [**DFO06**].

to maintain the steady rightward unfolding of the staircase: after recursively unfolding a band and its children, the staircase might reverse, which then obstructs the regions hanging below and above the staircase. The technique introduced in [**DFO05**] to surmount this problem is to unfold a subpiece of the polyhedron via a "helical peel," which, together with the Manhattan tower assumption used in that work, allows a choice of *suturing direction* upon emerging from the recursive call. Fig. 6(ab) shows a simple example where the peel starts at s, winds around leaving a gap, turns around on the back face K via strip K_0, and threads back through the gap to return to t, adjacent to the starting point s. This permits reversing the direction of the unfolding of an adjacent band (not shown in the figure) that touches s and t: the unfolding could approach s, interrupt and spiral around the box, finish at t, and resume unfolding in the reverse direction. Fig. 6(c) shows that the staircase structure is maintained, and is poised to be interfaced to adjacent staircases, with s at t extreme left and right respectively. Another, slightly more complicated helical unfolding permits continuing an interrupted adjacent band unfolding in the same direction.

These two unfoldings, together with the special structure of Manhattan-tower polyhedra, permit this technique to unfold the polyhedron by spiraling around the y-direction, and recursing on front and back children, as illustrated in Fig. 7. (As in Figs. 5 and 6, the bands here are parallel to the y axis rather than parallel to the z-axis as in the orthostack algorithms.) Here recursive calls encounter the labeled "blocks" in the order $r_1, \ldots, r_7$.

As mentioned, the helices must be joined or "sutured" at each parent-child arc in the recursion tree. The *reverse-direction suture*, illustrated in Fig. 6, is used at the r_1/r_2 junction and at the r_2/r_5 junction in Fig. 7, where the helix reverses its cw/ccw turning at the join. This is most evident at the latter junction: the helix

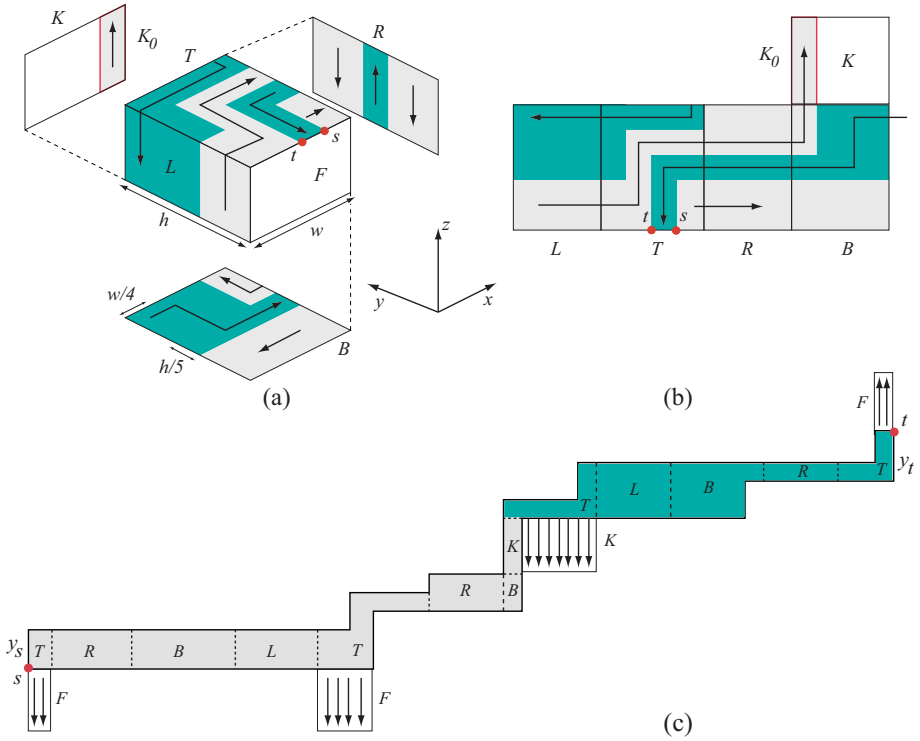

FIGURE 6. Reverse-direction unfolding of single box. (a) Front view of box and mirror view of left, bottom, and back faces. (b) All but front faces of box flattened out. (c) Staircase unfolding of box, with front and back "illuminated" faces (marked by parallel arrows) hung below. Fig. 6 in [**DFO05**].

enters back child r_5 while heading right ($+x$) on the top of r_2 but emerges, after r_5, heading left ($-x$). The *same-direction suture* is employed at all other junctions in Fig. 7. For example, the helix on r_2 enters front child r_3 heading right, and emerges again heading right (from which point it goes underneath r_2 to the front child r_4). Each of the same-direction sutures employs vertical strips (such as I_2 and I_3) on front faces to reach underneath, which is key to enabling same-direction emergence.

The necessity to wind around each band leads to a (4×5) refinement.

3.4. Nested Helical Peels. Although not evident from the sketches presented here, the shape restriction provided by the Manhattan tower permits the vertical strips that allow the use of both same- and reverse-direction sutures. This in turn allows traversal of the recursion tree in a natural order. This seems not possible for general polyhedra.

Consider Fig. 5(a), which is not a Manhattan tower in the orientation shown. For example, a line parallel to the z-axis could intersect P in three components by passing through C, D, and B. This vertical stacking precludes the use of the vertical strips needed for same-direction sutures. Forced to use reverse-direction sutures at every juncture results in the peel sometimes heading "the wrong way"

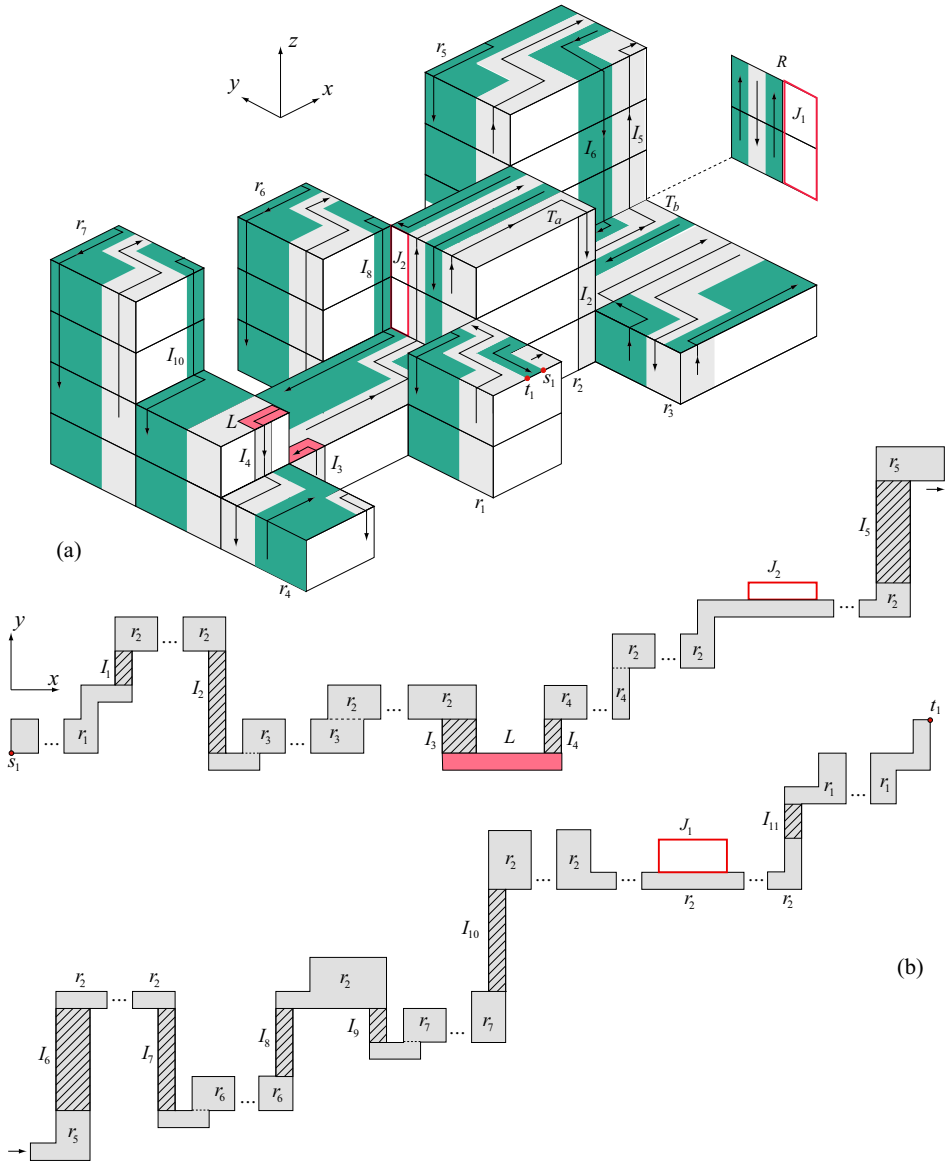

FIGURE 7. (a) The helices around the bands of a Manhattan Tower polyhedron. (b) A sketch of the staircase unfolding. Fig. 12 in [**DFO05**].

after emerging from a recursive call. Accepting this situation as forced by a general recursion leads to the necessity of retracing a previously traversed path, slightly displaced, until one has returned adjacent to the starting point s. Fig. 8 hints at the general idea, just on a one-layer example with root band b and children b_i. The peel starts at s, and then oscillates back and forth between the front children $b_1, \ldots, b_5$, reversing direction at each one and so forming a nested configuration of paths. The path then visits the back children $b_6, \ldots, b_{10}$ with a similar nested oscillation. As

the path emerges from the last recursive call on b_{10} at point t_{10}, it is "trapped" deep inside the convoluted path, and can only reach t, adjacent to s, by retracing. The recursive crisscrossing of a band leads to strips of thickness $\varepsilon = 1/2^{\Theta(n)}$ for a polyhedron of n vertices. To distinguish this from grid-edge unfolding, it was named "epsilon unfolding." It can be viewed as achieving a $2^{O(n)} \times 2^{O(n)}$ refined grid unfolding.

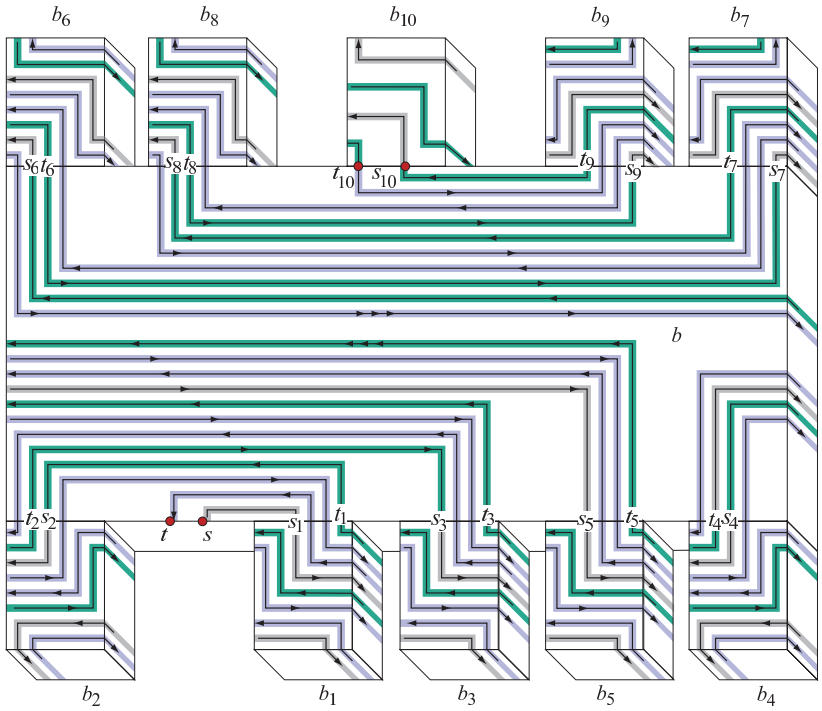

FIGURE 8. Helix enters b at s, visits the child blocks in the order $b, b_1, \ldots, b_{10}$, entering b_{10} at s_{10} and and emerging at t_{10}. To return back to t adjacent to s, the entire path must be retraced in reverse. Based on Fig. 12 in [**DFO07**].

4. Prospects

In the absence of a counterexample to the boxed question in Sec. 2, one might take as working hypothesis that a (1×1) grid-edge unfolding exists for any orthogonal polyhedron. There seems some hope of converting the epsilon-unfolding just sketched to a $O(1)\times O(1)$ grid-edge unfolding, by more carefully organizing the recursive structure.[4] But without either shape restrictions or the flexibility afforded by vertex unfolding, the gap between $O(1)\times O(1)$ and (1×1) seems formidable. What has been glossed over here in our description of the (1×1) algorithms, particularly [**DFO06**], are the complications that arise when a single grid face needs to be employed for several purposes in the algorithm. This sharing of duties often can be resolved by refining the face into subfaces, one per task. But if the goal

[4]Personal communication from Erik Demaine, June 2006.

is a (1×1) algorithm, instead conflicts must be resolved by careful ordering of the recursive calls, and other techniques. Perhaps, then, the resolution of the (1×1) question awaits a new algorithmic idea.

Acknowledgments. Much of this survey is drawn from joint work with Erik Demaine, Mirela Damian, and Robin Flatland. I am grateful also for extensive and useful comments by John Iacono.

References

[BDD+98] T. Biedl, E. D. Demaine, M. L. Demaine, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, and S. Whitesides, *Unfolding some classes of orthogonal polyhedra*, Proc. 10th Canad. Conf. Comput. Geom., 1998, pp. 70–71. Full version in *Elec. Proc.*: `http://cgm.cs.mcgill.ca/cccg98/proceedings/cccg98-biedl-unfolding.ps.gz`.

[BLS05] T. Biedl, A. Lubiw, and J. Sun, *When can a net fold to a polyhedron?*, Comput. Geom. Theory Appl. **31** (2005), no. 3, 207–218.

[DEE+03] E. D. Demaine, D. Eppstein, J. Erickson, G. W. Hart, and J. O'Rourke, *Vertex-unfoldings of simplicial manifolds*, Discrete Geometry (A. Bezdek, ed.), Marcel Dekker, 2003, pp. 215–228.

[DFMO05] M. Damian, R. Flatland, H. Meijer, and J. O'Rourke, *Unfolding well-separated orthotrees*, 15th Annu. Fall Workshop Comput. Geom., November 2005, pp. 23–25.

[DFO05] M. Damian, R. Flatland, and J. O'Rourke, *Unfolding Manhattan towers*, Proc. 17th Canad. Conf. Comput. Geom., 2005, pp. 204–207. Full version: arXiv:0705.1541v1 [cs.CG].

[DFO06] ———, *Grid vertex-unfolding orthogonal polyhedra*, Proc. 23rd Sympos. Theoret. Aspects Comput. Sci., Lecture Notes Comput. Sci., vol. 3884, Springer-Verlag, 2006, pp. 264–276. arXiv:05090.54v2 [cs.CG] supercedes preliminary version.

[DFO07] ———, *Epsilon-unfolding orthogonal polyhedra*, Graphs Combin., Akiyama-Chvatal Festschrift, 2007.

[DIL05] E. D. Demaine, J. Iacono, and S. Langerman, *Grid vertex-unfolding of orthostacks*, Proc. 2004 Japan Conf. Discrete Comput. Geom., Lecture Notes Comput. Sci., vol. 3742, Springer, November 2005, pp. 76–82.

[DM04] M. Damian and H. Meijer, *Grid edge-unfolding orthostacks with orthogonally convex slabs*, 14th Annu. Fall Workshop Comput. Geom., November 2004, pp. 20–21.

[DO05a] E. D. Demaine and J. O'Rourke, *Open problems from CCCG 2004*, Proc. 17th Canad. Conf. Comput. Geom., 2005, pp. 303–306.

[DO05b] E. D. Demaine and J. O'Rourke, *A survey of folding and unfolding in computational geometry*, Combinatorial and Computational Geometry (J. E. Goodman, J. Pach, and E. Welzl, eds.), Mathematics Sciences Research Institute Pub., vol. 52, Cambridge University Press, 2005, pp. 167–211.

[DO07] ———, *Geometric folding algorithms: Linkages, origami, polyhedra*, Cambridge University Press, 2007. `http://www.gfalop.org`.

[KBGK98] K. K. Kim, D. Bourne, S. Gupta, and S. S. Krishna, *Automated process planning for sheet metal bending operations*, J. Manufacturing Syst. **17** (1998), no. 5, 338–360.

[LT07] W. Liu and K. Tai, *Optimal design of flat patterns for 3D folded structures by unfolding with topological validation*, Comput.-Aided Design (2007), to appear.

[O'R00] J. O'Rourke, *Folding and unfolding in computational geometry*, Proc. 1998 Japan Conf. Discrete Comput. Geom., Lecture Notes Comput. Sci., vol. 1763, Springer-Verlag, 2000, pp. 258–266.

[TLT04] K. Tai, W. Liu, and G. Thimm, *Unfolding and flat layout design of non-manifold 3D folded structures*, Comput.-Aided Design & Appl. **1** (2004), no. 1-4, 439–448.

[Wan97] C.-H. Wang, *Manufacturability-driven decomposition of sheet metal products*, Ph.D. thesis, Robotics Inst., Carnegie Mellon Univ., 1997.

Dept. Computer Science, Smith College, Northampton, MA 01063, USA.
E-mail address: `orourke@cs.smith.edu`