Unfolding Manhattan Towers

Mirela Damian* Robin Flatland† Joseph O'Rourke‡

Abstract

We provide an algorithm for unfolding the surface of any orthogonal polyhedron that falls in a particular shape class we call Manhattan Towers, to a planar simple orthogonal polygon. The algorithm cuts along edges of a 5×5 refinement of the vertex grid.

1 Introduction

It is a long standing open problem to decide whether the surface of every convex polyhedron can be *edge unfolded*: cut along edges and unfolded flat to one piece without overlap [DO05]. It is known that some nonconvex polyhedra have no edge unfolding. A simple example is a small box sitting on top of a larger box. However, no example is known of a nonconvex polyhedron that cannot be unfolded with unrestricted cuts, i.e., cuts that may pass through the interior of faces.

The difficulty of these questions led to the exploration of *orthogonal polyhedra*, those whose faces meet at right angles. Progress has been made in two directions. First, by restricting the shapes to subclasses of orthogonal polyhedra, such as the "orthostacks" and "orthotubes" studied in [BDD+98]. And second, by generalizing the cuts beyond edges but with some restrictions. In particular, a *grid-unfolding* partitions the surface of the polyhedron by coordinate planes through every vertex, and then restricts cuts to the resulting grid. The box-on-box example mentioned above can be easily grid-unfolded. Recent work on grid-unfolding of orthostacks is reported in [DM04] and [DIL04].

Because on the one hand no example is known of an orthogonal polyhedron that cannot be grid-unfolded, and on the other hand no algorithm is known for grid-unfolding other than very specialized shapes, the suggestion was made in [DO04] to seek *refined grid-unfoldings*, where every face of the vertex grid is further refined into a regular $k \times k$ grid. It is this line we pursue in this paper, on a class of shapes not previously considered.

*Dept. Comput. Sci., Villanova Univ., Villanova, PA 19085, USA. mirela.damian@villanova.edu.

†Dept. Comput. Sci., Siena College, Loudonville, NY 12211, USA. flatland@siena.edu.

‡Dept. Comput. Sci., Smith College, Northampton, MA 01063, USA. orourke@cs.smith.edu. Supported by NSF Distinguished Teaching Scholars award DUE-0123154.

We define "Manhattan Tower (MT) polyhedra" to be the natural generalization of "Manhattan Skyline polygons." Although we do not know of an unrefined grid-unfolding for this class of shapes, we prove (Theorem 2) that there is a 5×5 grid-unfolding. Our algorithm peels off a spiral strip that winds first forward and then interleaves backward around vertical slices of the polyhedron, recursing as attached slices are encountered. The algorithm extends beyond MT shapes, and holds some promise for wider generalization.

2 Definitions

Let Z_k be the plane $\{z = k\}$, for $k \geq 0$. Define $\mathcal{P}$ to be a *Manhattan Tower* (MT) if it is an orthogonal polyhedron such that:

1. $\mathcal{P}$ lies in the halfspace $z \geq 0$, and its intersection with Z_0 is a simply connected orthogonal polygon;
2. For $k < j$, $\mathcal{P} \cap Z_k \supseteq \mathcal{P} \cap Z_j$: the cross-section at higher levels is nested in that for lower heights.

A Manhattan Tower $\mathcal{P}$ may be viewed as consisting of nested layers, with each layer the extrusion of an orthogonal polygon. The *base* of $\mathcal{P}$ is its bottom layer, which is bounded below by Z_0 and above by the xy-plane passing through the first vertex with $z > 0$. See Fig. 1a for an example of a MT.

We use the following notation to describe the six types of faces, depending on the direction in which the outward normal points: *front*: $-y$; *back*: $+y$; *left*: $-x$; *right*: $+x$; *bottom*: $-z$; *top*: $+z$.

An *x-edge* is an edge that is parallel to the x-axis; *y-edges* and *z-edges* are defined similarly. Clockwise (cw) and counterclockwise (ccw) directions are defined with respect to the viewpoint from $y = -\infty$.

3 Base Partitioning

We start with the partition π of the base layer induced by the xz-planes passing through every vertex of P. Such a partition consists of rectangular boxes only. See Fig. 1b. The dual graph of π has a node for each box and an edge between each pair of nodes corresponding to adjacent boxes. Since the base is simply connected, the dual graph of π is a tree T (Fig. 1c), which we refer to as the *recursion tree*. The root of T is a node corresponding to a box (the *root box*) whose front face has a minimum y-coordinate (breaking ties arbitrarily).

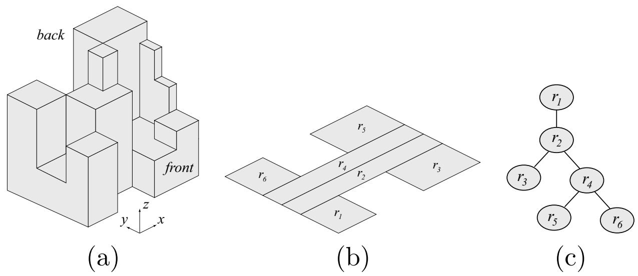

Figure 1: (a) Manhattan Tower $\mathcal{P}$ (b) Partition π of the base of $\mathcal{P}$ (c) Recursion tree T.

It turns out that nearly all issues are present in unfolding single-layer MTs, which we describe in Section 4. The algorithm is then extended to handle towers in Section 5.

4 Single-Layer MTs

We describe the unfolding algorithm recursively, starting with the base case in which the partition π consists of a single rectangular box.

4.1 Single Box Unfolding

Let r be a rectangular box and let A, B, C, D, E and F be the top, right, bottom, left, back and front faces of r, respectively. Let s and t be two points on the same x-edge or opposite x-edges of the front face of r. The unfolding of r starts at s and ends at t. Here we discuss the case where both s and t lie on the top edge of the front face; the other cases are similar and will be illustrated in subsequent sections.

The main unfolding idea is to cut the box into a staircase-like strip that starts at s, spirals cw around side faces A, B, C and D, crosses the back face E and then spirals ccw around back to t. This idea is illustrated in Fig. 2. The resulting strip ξ can be unfolded flat and laid out horizontally in a plane. The front face F and back face E can be flipped up and attached vertically to this strip without overlap, as in Fig. 2c.

4.2 Recursion Structure

In general, a box r has children (adjacent boxes) attached along its front and/or back face. Call the children attached on the front the *front children* and the children attached on the back the *back children.* In unfolding r, we unwind the "side faces" (top, bottom, left, and right faces) into a staircase-like strip just as described for the single box. But when the strip runs alongside the front or back face of r and encounters an adjacent child, the unfolding of r is temporarily suspended, the child is recursively unfolded, then the unfolding of r resumes where it left off.

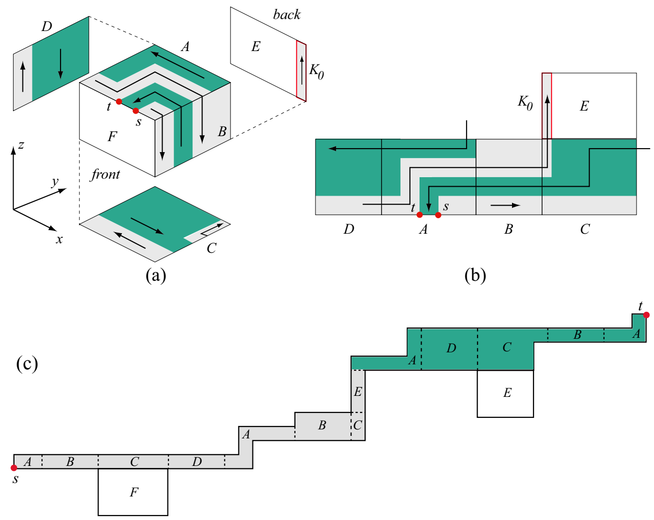

Figure 2: Single box unfolding. (a) Front view of box r and mirror view of left (D), bottom (C) and back (E) faces, marked with unfolding cuts (b) Faces of r flattened out (front face not shown) (c) Spiral unfolding of r; labels identify faces containing the unfolded pieces.

At any time in the recursive algorithm there is a *forward* direction, corresponding to the initial spiraling from front to back (the lighter strip in Fig. 2), and an opposing *backward* direction corresponding to the subsequent reverse spiraling from back to front (the darker strip in Fig. 2). When the recursion processes a front child, the sense of forward/backward is reversed: we view the coordinate system rotated so that the $+y$ axis is aligned with the spiral's forward direction, with all terms tied to the axes altering appropriately. In particular, this means that the start and end unfolding points s', t' of a front child r' lie on the front face of r', as defined in the rotated system.

4.3 Suturing Techniques

We employ two methods to "suture" the unfolding of a child to its parent's unfolding. The first method, *same-direction suture*, is used to suture all front children to their parent. If there are no back children, then the back face of the parent is used to reverse the direction of the spiral to complete the parent's unfolding, as described in Section 4.1 for the single box. However, if the parent has one or more back children, these children cover parts or perhaps the entire back face of the parent, and thus a back face strip (such as K_0 in Fig. 2) may not be available for the reversal. So instead we use a second suturing method, *reverse-direction suture*, for one of the back children. This suture uses the child's unfolding to reverse the direction of the parent's spiral, and does not require a back-face strip. We choose *exactly one* back child for reverse-direction suturing. Although any such child would serve, for definiteness we select the rightmost child. Our suturing rules are as follows:

1. For every front child, use same-direction suturing.
2. For the rightmost back child, use reverse-direction suturing.
3. For remaining back children, use same-direction suturing.

Same-direction suture. We first note that a front child r' never entirely covers the front face of its parent box r. The same-direction suture may only be applied in such a situation of non-complete coverage of the shared face, for it uses an ε-strip of that face.

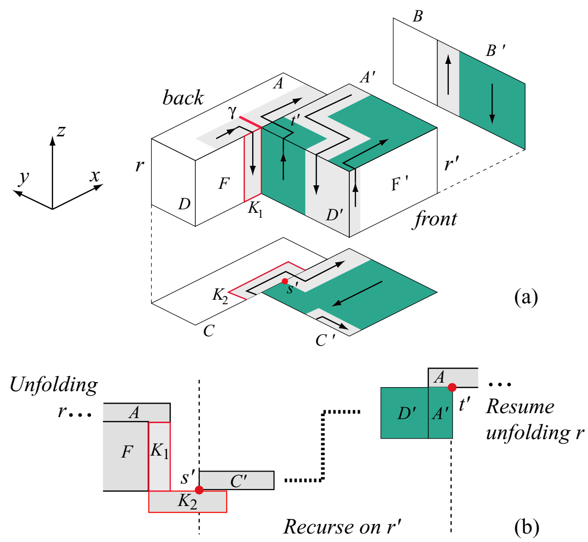

Figure 3: Same-direction suture. (a) Front view of faces root box r and front child r', with mirror bottom and right views. (b) Recursive unfolding.

This suture begins at the point where the parent's spiral meets an adjacent child as it runs alongside its front or back face, as illustrated in Fig. 3 for parent r and front child r'. It begins by cutting an ε-thick strip off the vertical (front or back) face of the parent alongside the child (strip K_1 in Fig. 3), then it takes an ε-thick bite off the opposite (top or bottom) face of the parent (strip K_2 in Fig. 3). This marks the point s' where the spiral unfolding of the child starts. The child's spiral unfolding ends at point t' vertically opposite to s'. When the child's unfolding is complete, the spiral unfolding of the parent resumes at t'. As the name suggests, this suturing technique preserves the unwinding direction (cw or ccw) of the parent's spiral.

Reverse-direction suture. This suture begins after the parent's spiral completes its first cycle around the side faces, as illustrated in Fig. 4 for parent r and back child r'. After a forward move in the $+y$-direction, the spiral starts a second cycle around the side faces, stops when it reaches the rightmost back child, then continues with an ε-thick strip S in the $+y$-direction. Let s' be the left corner of S on the boundary between r and r'. The unfolding of r' begins at point s' and ends at point t' slightly to the left of s'. When the child's unfolding is complete, the unfolding of the parent resumes at t', with the spiral unwinding in reverse direction.

Fig. 5 shows the two sutures used together to unfold a five box H-shaped base.

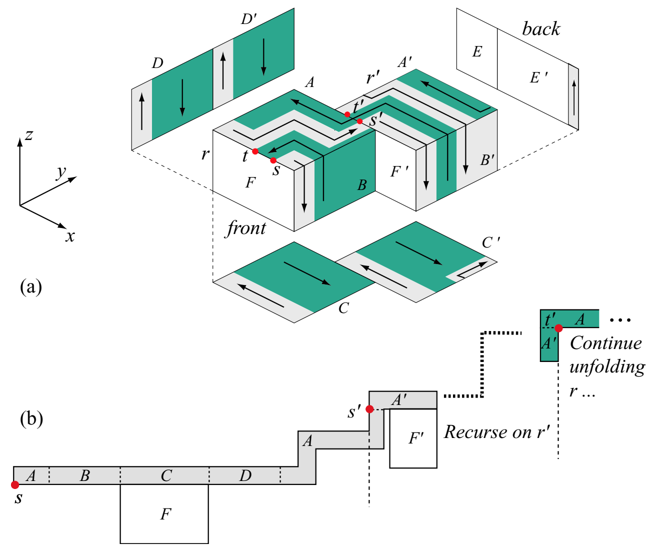

Figure 4: Reverse-direction suture. (a) Front view of faces root box r and back child r', with mirror bottom, left and back views. (b) Recursive unfolding.

4.4 Attaching Front and Back Faces

The spiral strip ξ unfolds all top, bottom, right, and left faces of the base. It also unfolds the ε-thick strips of front and back faces used by the same-direction sutures (K_1 in Fig. 3) and the ε-thick strips of back faces used to reverse the spiral direction in the base cases (K_0 in Fig. 2). The staircase structure of ξ guarantees that no overlap occurs. The following lemma (whose proof is omitted in this abstract) is key to proving nonoverlap of the entire unfolding:

Lemma 1 *For any box b, exposed front and back pieces of b that are not part of ξ can be attached orthogonal to ξ without overlap.*

5 Multiple-Layer MTs

Few changes are necessary to make the single-layer unfolding algorithm handle multiple-layer Manhattan

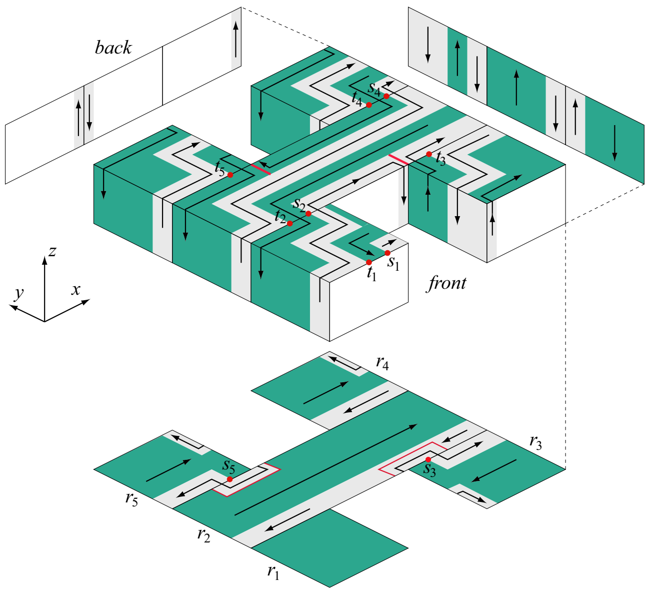

Figure 5: Unfolding Single-Layer MTs.

Towers. When there are multiple-layers, the basic unit to unfold is a vertical *stack* S_r consisting of a box r in the base layer partition π and all the towers that rest on top of r. The unfolding of S_r is similar to that of r but with the spiral cycling up and down over the towers; compare Figs. 2a and 6. The structure of the recursive calls is identical to the single-layer case. What differs is that transitions from one stack to another may move up or down between towers; compare Figs. 5 and 7. These differences lengthen ξ horizontally and vertically.

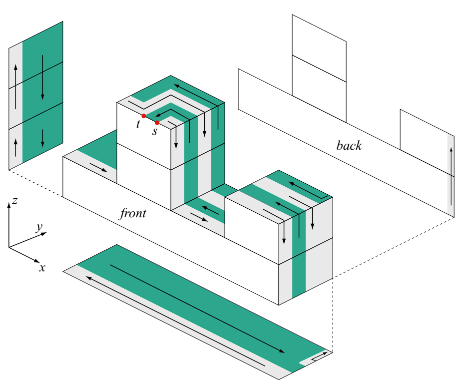

Figure 6: Unfolding a vertical stack.

It may also happen that ξ does not include every left or right face strip (such as K_3 in Fig. 7), but such strips can be easily attached to ξ without overlap.

6 Conclusions

Although we have used ε-strips throughout, it should be evident that there is no need for arbitrarily thin strips:

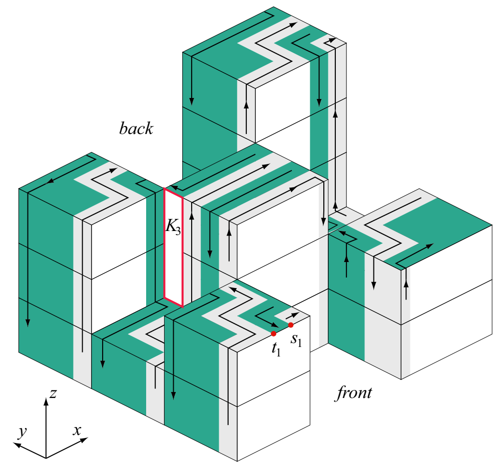

Figure 7: Unfolding Multiple-Layer MTs.

Theorem 2 *Every Manhattan Tower polyhedron can be edge-unfolded with a 5×5 refinement of each face of the vertex grid.*

The algorithm can be easily implemented to run in $O(n^2)$ time.

We know our algorithm as described works unaltered on objects beyond the class of Manhattan Towers, and we are currently extending the algorithm to handle a wider range of orthogonal polyhedra.

Acknowledgments. We thank Ivaylo Ilinkin, Lynn Stauffer, Marcello Sztainberg, and all the participants of the 2004 DIMACS Reconnect workshop held at St. Mary's College. And we thank Henk Meyer for enlightening discussions.

References

[BDD+98] T. Biedl, E. Demaine, M. Demaine, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, and S. Whitesides. Unfolding some classes of orthogonal polyhedra. In *Proc. 10th Canad. Conf. Comput. Geom.*, pages 70–71, 1998.

[DIL04] E. D. Demaine, J. Iacono, and S. Langerman. Grid vertex-unfolding of orthostacks. In *Proc. Japan Conf. Discrete Comp. Geom.*, November 2004. To appear in LNCS, 2005.

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

[DO04] E. D. Demaine and J. O'Rourke. Open problems from CCCG 2004. In *Proc. 16th Canad. Conf. Comput. Geom.*, 2004.

[DO05] 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, *Combinatorial and Computational Geometry*. Cambridge University Press, 2005.