

Unfolding Lattice Polygons on Some Lattice Polyhedra*

Sheung-Hung Poon[†]

Abstract

We consider the problem of unfolding lattice polygons embedded on the surface of some classes of lattice polyhedra. We show that an unknotted lattice polygon embedded on a lattice orthotube or orthotree can be convexified in $O(n)$ moves and time, and a lattice polygon embedded on a lattice Tower of Hanoi or Manhattan Tower can be convexified in $O(n^2)$ moves and time.

1 Introduction

Graph reconfiguration problems have wide applications in contexts including robotics, molecular conformation, animation, wire bending, rigidity and knot theory. The motivation for reconfiguration problems of lattice graphs arises in applications in molecular biology and robotics. For instance, the bonding-lengths in molecules are often similar [8, 13, 14], as are the segments of some types of robot arms.

A *unit tree* (resp. *unit polygon*) is a tree (resp. polygon) containing only edges of unit length. An *orthogonal tree* (resp. *orthogonal polygon*) is a tree (resp. polygon) containing only edges parallel to coordinate-axes. A *lattice tree* (resp. *lattice polygon*) is a tree (resp. polygon) containing only edges from a square or cubic lattice. Note that a lattice tree or polygon is basically a unit orthogonal tree or polygon. A *lattice polyhedron* is a polyhedron whose surface is the union of lattice faces from a cubic lattice. A graph is *simple* if non-adjacent edges do not intersect. We consider the problem about the reconfiguration of a simple chain, polygon, or tree through a series of continuous motions such that the lengths of all graph edges are preserved and no edge crossings are allowed. A tree can be *straightened* if all its edges can be aligned along a common straight line such that each edge points “away” from a designated root node. In particular, a chain can be straightened if it can be stretched out to lie on a straight line. A polygon can be *convexified* if it can be reconfigured to a convex polygon. We say a chain or tree is *locked* if it cannot be straightened.

We say a polygon is *locked* if it cannot be convexified. We consider one *move* in the reconfiguration as a continuous monotonic change for the joint angle at some vertex, or a continuous axial rotation of one of its angle side around its another angle side in 3D, during which no edge crossings occur.

In four dimensions or higher, a polygonal tree can always be straightened, and a polygon can always be convexified [9]. In two dimensions, a polygonal chain can always be straightened and a polygon can always be convexified [11, 17, 6]. However, there are some trees in two dimensions that can lock [3, 10, 15]. In three dimensions, even a 5-chain can lock [4]. Alt et al. [2] showed that deciding the reconfigurability for trees in two dimensions and for chains in three dimensions is PSPACE-complete. However the problem of deciding straightenability for trees in two dimensions and for chains in three dimensions remains open. Due to the complexity of the problems in two and three dimensions, some special classes of trees and polygons have been considered. Cantarella and Johnston [7] showed that a unit 5-chain in three dimensions can always be straightened. Demaine et al. [12] even studied interlocked configurations of several short chains in three dimensions. In particular, they showed that two 3-chains cannot interlock, but three of them can. They also showed that a 3-chain and a 4-chain can interlock. Poon [15] showed that a unit tree of diameter 4 in two dimensions can always be straightened. In their paper, they posed a challenging open question whether a unit tree in either two or three dimensions can always be straightened.

Biedl et al. [4] proved that an open chain on the surface of a convex polyhedron can always be straightened. In this paper, we show that an unknotted lattice polygon embedded on a lattice orthotube, orthotree, Tower of Hanoi, and Manhattan Tower can always be convexified.

2 Preliminaries

A *near-lattice edge* is a unit-length edge within distance $\epsilon \ll 1$ from some lattice edge. The particular lattice edge is called the *core edge* of the corresponding near-lattice edge. A *core vertex* is a lattice vertex of some core edge. A *near-lattice tree* (resp. *near-lattice polygon*) is a tree (resp. polygon) that contains only near-lattice edges. Suppose P is a near-lattice

*The author was supported by the Netherlands’ Organization for Scientific Research (NWO) under project no. 612.065.307.

[†]Department of Mathematics & Computer Science, Technical University of Eindhoven, spoon@win.tue.nl

tree or polygon. The *core* of P , denoted by $K(P)$, is the union of core edges for all edges in P . A *spring* in P is the set of edges in P converging to a common lattice edge. A spring with only one edge is called a *singleton*. A near-lattice edge or spring is called *embedded or lying on* a lattice polyhedron if its core is embedded on the lattice polyhedron.

3 Lattice Orthotube

A *lattice orthotube* is a lattice polyhedron made out of boxes that are glued face-to-face such that its face-to-face contact graph is a path or cycle. A lattice orthotube is called *open* if its face-to-face contact graph is a path; otherwise it is called *closed*. In an open lattice orthotube, the two blocks whose degrees in the face-to-face contact graph are one are called the *end blocks* of the given orthotube. An *end face* of an open orthotube is a face of its end block such that it is opposite to the face which is the intersection of the end block and the second last end block.

Remark that there are some orthogonal polygons embedded on some orthogonal polyhedra that can lock as shown in Figure 1(a), and there are some lattice polygons embedded on some closed lattice orthotubes can knot as shown in Figure 1(b). This moti-

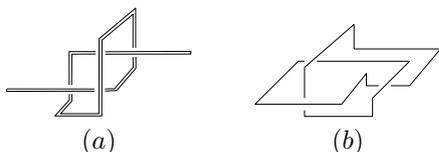


Figure 1: (a) A 3D locked orthogonal polygon. (b) A 3D knotted lattice polygon.

vates that we consider the lattice polygons embedded on open lattice orthotubes, and the *unknotted* lattice polygons embedded on closed lattice orthotubes.

3.1 Open Lattice Orthotube

In this subsection, we will show that lattice polygons embedded on open lattice orthotubes can always be convexified.

Consider a near-lattice polygon embedded on open lattice orthotube. The end block of the orthotube is called *free* if its end face does not contain any edge from the core of the given embedded near-lattice polygon. It is clear that the free end blocks of an open orthotube do not help in our unfolding process and can be truncated away. We thus assume the end block of any orthotube mentioned below is not free. Our algorithm proceeds by folding up the polygon from the non-free end blocks of the orthotube successively. Suppose we are given a near-lattice polygon embedded on a lattice orthotube at the beginning of each

folding step. We fold up the part of the given near-lattice polygon lying on the end block onto the springs of the second last end block using constant number of moves. After one folding step, again we obtain back a near-lattice polygon. We repeat this step until the remained orthotube contains only one lattice cell. Now it is clear that the near-lattice polygon embedded on one lattice cell can be unfolded to a convex polygon straightforwardly. We first need the following lemma on how to perform a folding step. Then we summarize our result in Theorem 2.

Lemma 1 *Given a near-lattice polygon P embedded on an open lattice orthotube \mathcal{Q} such that both end blocks of \mathcal{Q} are not free, and \mathcal{Q} contains more than one lattice cells. Then the part of P lying on an end block of \mathcal{Q} can be folded onto some springs on the second last end block so that the current end block becomes free.*

Proof. Suppose the end face of the orthotube \mathcal{Q} is facing to the right. We divide into three cases depending on how many core edges of P lie on the end face.

Case 1: The end face contains one core edge. Then the end block can be folded up as shown in Figure 2.

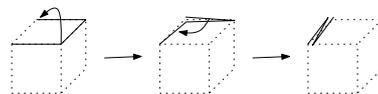


Figure 2: *Case 1* of folding up an end block.

Case 2: The end face contains two core edges. Then for the two subcases in Figure 3(a) or (b), the end block can be transformed into *Case 1* as shown in the figures; for the subcase in Figure 3(c), it can be treated as two occurrences of *Case 1*. Consequently, the resulting end block can be folded up by applying once or twice the operation of *Case 1*.

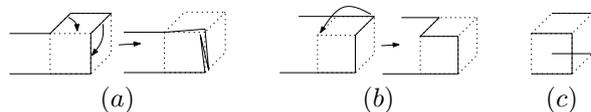


Figure 3: *Case 2* of folding up an end block.

Case 3: The end face contains three core edges. Then the end block can be transformed into *Case 1* as shown in Figure 4.

Note that for any of the operations above, only the joint angles at the end vertices of the end edges of a constant number of related springs are changed. Thus folding up the end block takes a constant number of moves and time. \square

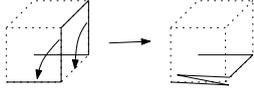


Figure 4: *Case 3* of folding up an end block.

Theorem 2 *A lattice polygon embedded on an open lattice orthotube can be convexified in $O(n)$ moves and time.*

3.2 Closed Lattice Orthotube

Given an unknotted lattice polygon embedded on a closed lattice orthotube. First it is clear that at any cross section, the intersection of the given lattice polygon and the cross-section cutting plane contains either zero, two or four corner vertices. If there is a point on lattice orthotube where the cross section does not intersect the given lattice polygon, then after cutting the closed orthotube open, we can use the algorithm for open lattice orthotube to unfold the polygon. Otherwise there is no such point where the cross section does not intersect the given lattice polygon. Then it is clear that there must exist some cross section at some lattice points where all the four corner vertices lie in the intersection of the given polygon and the current cross-section cutting plane, and the structure of its neighborhood on the polygon is in one of the cases in Figure 5.

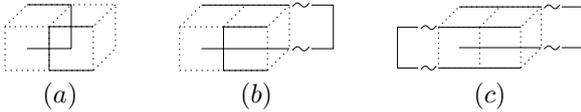


Figure 5: Cases for folding a closed lattice orthocube.

For case (a) in the figure, the closed orthocube can still be cut along the cross section to obtain an open lattice orthocube. In both cases (b) and (c), by successively applying the folding operation of *Case 1* to fold the end block of an open lattice orthocube, we can transform them to case (a) by eliminating the long U-turns. Therefore, we have the following theorem.

Theorem 3 *An unknotted lattice polygon embedded on a closed lattice orthotube can be convexified in $O(n)$ moves and time.*

4 Lattice Orthotree

A *lattice orthotree* is a lattice polyhedron made out of boxes that are glued face-to-face such that its face-to-face contact graph is a tree. In a lattice orthotree, those blocks whose degrees in the face-to-face contact graph are one are called the *end blocks* of the given

orthotree. To convexify a lattice polygon embedded on a lattice orthotree, the algorithm runs in the same fashion as that for an open lattice orthotube. We fold up the polygon from the end blocks successively.

Theorem 4 *A lattice polygon embedded on a lattice orthotree can be convexified in $O(n)$ moves and time.*

5 Lattice Towers

Let Z_k be the plane $z = k$ for $k \geq 0$. A *Manhattan Tower* \mathcal{Q} is an orthogonal polyhedron such that

1. \mathcal{Q} lies in the halfspace $z \geq 0$ and its intersection with Z_0 is a simply connected orthogonal polygon;
2. For $j > k \geq 0$, $\mathcal{Q} \cap Z_j \subset \mathcal{Q} \cap Z_k$: the cross section at a higher level is nested in that at a lower level.

A *Tower of Hanoi* \mathcal{Q} is a Manhattan Tower such that its intersection with Z_k for $k \geq 0$ is either empty or a simply connected orthogonal polygon.

5.1 Lattice Tower of Hanoi

Given a lattice polygon embedded on a lattice Tower of Hanoi. The overall intuition of the unfolding algorithm is to press level by level vertically downwards from the highest level. Let's first consider the detail for pressing the highest level L down to the second highest level L' under the condition that L' is not the lowest level. Notice that between L and L' , there are vertically lattice polygon edges connecting them, which we call *legs*. And we also call the end vertex of the leg at L' the *foot* of the leg. To press level L to level L' , we press the maximal polygon path on level L one by one onto the level L' . More precisely, each maximal polygon path α on level L has two legs connecting to level L' . We will collapse one leg and pull one edge of α towards one of the collapsed leg. On the other end, the other leg is pulled to replace the position of one end edge of α , and the end edge of α is pulled to replace the position second last end edge of α , and so on so forth. See the operation (a) in Figure 6. Remark that at the end of the pressing step, we

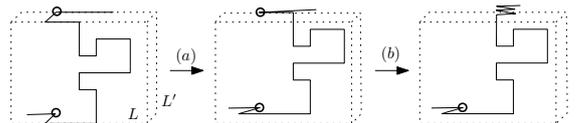


Figure 6: Pressing a path from level L down to L' .

don't really press α down to the level L' ; but we keep a level higher but very close the level L' to prevent edge crossing with polygon edges in level L' . Later on, this treatment also gives us some convenience to recognize which springs are "inside" and which springs are

“outside” for those springs with the same core edges. We then have the convention that the highest spring is the “most inside”. After the pressing step, if the core edge of some spring has degree one at its one end, we need to collapse those dangling springs. See the operation (b) in Figure 6. It is clear that this pressing step takes at most $O(n)$ moves and time. Notice that each time we press a path down to one level lower, two vertical legs are collapsed. There can be at most $O(n)$ vertical legs. All the pressing operations take $O(n^2)$ moves and time.

After all the pressing steps, we obtain near-lattice polyhedron of height one. At this stage, using a generalized end-block collapsing similar to what we did for orthotubes and orthotrees, we can fold up the current near-lattice polygon to become a near-lattice unit square, which can then be convexified straightforwardly. This end-block collapsing process can be realized such that it takes $O(n^2)$ moves and time. However, its detail is eliminated in this abstract. Hence, we have the following theorem.

Theorem 5 *A lattice polygon embedded on a lattice Tower of Hanoi can be convexified in $O(n^2)$ moves and time.*

5.2 Lattice Manhattan Tower

Given a lattice polygon embedded on a lattice Manhattan Tower. The algorithm is the same as that for lattice Tower of Hanoi. The only difference is that when we press the highest level L to the second highest level L' , we need to press several separate orthogonal polygonal regions on L instead of only one for lattice Tower of Hanoi. Thus we have the following theorem.

Theorem 6 *A lattice polygon embedded on a lattice Manhattan Tower can be convexified in $O(n^2)$ moves and time.*

6 Open Problems

We conjecture that a lattice polygon embedded on a general lattice polyhedron can always be convexified. The conjecture [16] that any unknotted lattice polygon in 3D can always be convexified is still open.

References

- [1] J. Alberto-Calvo, D. Krizanc, P. Morin, M. Soss, G. Toussaint. Convexifying polygons with simple projections. *Information Processing Letters*, 80(2):81-86, 2001.
- [2] H. Alt, C. Knauer, G. Rote, and S. Whitesides. The Complexity of (Un)folding. In *Proc. 19th ACM Symposium on Computational Geometry (SOCG)*, 164–170, 2003.
- [3] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O’Rourke, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides. A Note on Reconfiguring Tree Linkages: Trees can Lock. *Discrete Applied Mathematics*, volume 117, number 1-3, pages 293–297, 2002.
- [4] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O’Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides. Locked and Unlocked Polygonal Chains in Three Dimensions. *Discrete & Computational Geometry*, volume 26, number 3, pages 269-281, October 2001.
- [5] 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 *Proc. of 10th Canadian Conference on Computational Geometry (CCCG)*, 1998.
- [6] J. Cantarella, E.D. Demaine, H. Iben, and J. O’Brien. An Energy-Driven Approach to Linkage Unfolding. In *Proceedings of the 20th Annual ACM Symposium on Computational Geometry (SoCG)*, 134–143, 2004.
- [7] J. Cantarella and H. Johnston. Nontrivial embedding of polygonal intervals and unknots in 3-space. *J. Knot Theory Ramifications*, 7, 1027–1039, 1998.
- [8] H.S. Chan and K.A. Dill. The protein folding problem. *Physics Today*, pages 24–32, February 1993.
- [9] R. Cocan and J. O’Rourke. Polygonal Chains Cannot Lock in 4D. *Computational Geometry: Theory & Applications*, 20, 105–129, 2001.
- [10] R. Connelly, E. Demaine, and G. Rote. Infinitesimally Locked Self-Touching Linkages with Applications to Locked Trees. In *Physical Knots: Knotting, Linking, and Folding of Geometric Objects in R^3* , edited by J. Calvo, K. Millett, and E. Rawdon (editors), 2002, pages 287–311, American Mathematical Society.
- [11] R. Connelly, E.D. Demaine, and G. Rote. Straightening Polygonal Arcs and Convexifying Polygonal Cycles. *Discrete & Computational Geometry*, volume 30, number 2, 205–239, 2003.
- [12] E. D. Demaine, S. Langerman, J. O’Rourke, and J. Snoeyink. Interlocked Open Linkages with Few Joints. In *Proc. 18th Annual ACM Symposium on Computational Geometry (SoCG)*, Barcelona, Spain, June 5-7, pages 189–198, 2002.
- [13] K.A. Dill. Dominant forces in protein folding. *Biochemistry*, 29(31), 7133–7155, August 1990.
- [14] B. Hayes. Prototeins. *American Scientist*, 86, 216–221, 1998.
- [15] S.-H. Poon. On Straightening Low-Diameter Unit Trees. In *Proc. 13th International Symposium on Graph Drawing (GD)*, 519–521, 2005.
- [16] S.-H. Poon. On Unfolding Lattice Polygons/Trees and Diameter-4 Trees. In *Proc. 12th Annual International Computing and Combinatorics Conference (COCOON)*, 186–195, 2006.
- [17] I. Streinu. A combinatorial approach for planar non-colliding robot arm motion planning. In *Proc. 41st ACM Annual Symposium on Foundations of Computer Science (FOCS)*, 443–453, 2000.