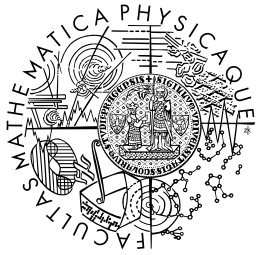

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Klára Pernicová

Grid-edge unfolding orthostacks

Department of Applied Mathematics

Supervisor of the bachelor thesis: doc. Mgr. Jan Kynčl, Ph.D.
Study programme: Computer Science

Prague 2024

I declare that I carried out this bachelor thesis on my own, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I want to thank my supervisor, doc. Mgr. Jan Kynčl, Ph.D., for his guidance throughout this thesis. I also want to thank the three anonymous reviewers from CCCG for their feedback on unzipping box towers. I am very grateful to Marian Poljak for his proofreading and constructive feedback. Lastly, I am very thankful to my family, friends, and neighbors for their support and for providing me with meals during this time.

Title: Grid-edge unfolding orthostacks

Author: Klára Pernicová

Department: Department of Applied Mathematics

Supervisor: doc. Mgr. Jan Kynčl, Ph.D., Department of Applied Mathematics

Abstract: An orthostack is an orthogonal polyhedron obtained by stacking orthogonal prisms (slabs) on top of each other. An unfolding is the process of cutting the surface of the polyhedron and flattening it to the plane. We describe a known algorithm for unfolding a subclass of orthostacks with orthogonally convex slabs, and we indicate why it is unsuitable for slabs of arbitrary height. Next, we describe a known algorithm for unfolding a subclass of orthostacks with rectangular faces, and we present a modification of this algorithm for unfolding box towers (orthostacks with rectangular slabs). Unzipping is a special type of unfolding whose cutting segments form a single path. As our main result, we show that every box tower has an unzipping. Finally, we introduce a subclass of orthogonal polyhedra named box towers streets and present an algorithm for unfolding.

Keywords: orthogonal polyhedron, orthostack, planar net, unfolding polyhedra

Název práce: Mřížkové rozkládání ortostacků

Autor: Klára Pernicová

Katedra: Katedra aplikované matematiky

Vedoucí bakalářské práce: doc. Mgr. Jan Kynčl, Ph.D., Katedra aplikované matematiky

Abstrakt: Ortostack je ortogonální mnohostěn vytvořený stavěním ortogonálních hranolů (pater) na sebe. Rozklad do sítě je proces, ve kterém rozřežeme povrch mnohostěnu a rozložíme ho do roviny. Popíšeme známý algoritmus pro rozklad třídy ortostacků s ortogonálně konvexními patry a naznačíme, proč není dostatečný pro patra libovolné výšky. Poté popíšeme známý algoritmus pro rozklad třídy ortostacků s obdélníkovými stěnami a představíme modifikaci tohoto algoritmu pro rozklad věží z kvádrů (ortostacky s obdélníkovými party). Rozzipování je speciálním typem rozkládání, kde řezné hrany tvoří jedinou cestu. Naším hlavním výsledkem je algoritmus pro rozzipování věží z kvádrů. Nakonec představíme třídu ortogonálních mnohostěnů pojmenovanou ulice s věžemi z kvádrů a ukážeme algoritmus pro rozklad do sítě.

Klíčová slova: ortogonální mnohostěn, ortostack, rovinná síť, rozkládání mnohostěnů

Contents

Chapter 1

Introduction

1.1 Overview of known results and types of unfolding

A long-standing open question by Dürer asks whether every convex polyhedron can be edge-unfolded into a single simple polygon [13]. We use nets for creating 3D objects from paper.

A polyhedron $P \subseteq \mathbb{R}^3$ is an *orthogonal polyhedron* if each face is parallel to an xy, yz, or xz plane.

As it is common in the literature we redefine the notion of a face as follows. We subdivide the surface of P with axis-parallel planes passing through the vertices of P. We will call the parts of the subdivision the *faces* of P. Note that in this definition each face of P is a rectangle. See Figure 1.1 for an illustration.

An *edge unfolding* of P is the process of cutting along the edges of the original polyhedron and isometrically mapping the cut surface into the plane with no interior overlap. A *grid-edge unfolding* is a generalization of an edge unfolding that allows cutting along the edges of the redefined faces of P. A grid-edge unfolding is called a *grid-edge unzipping* if the cutting segments altogether form a path on the surface of P. The result of the unfolding is called a *net*; we also refer to it as the *unfolding*. A grid-edge unfolding is *simple* if no vertices (and therefore no edges) overlap; in other words, the images of the cutting segments form a simple closed curve after flattening to the net; see Figure 1.2. A net created by a simple unfolding is called a *simple net*. Simple nets are more practical for constructing polyhedra from paper as they leave some space for glue tabs.

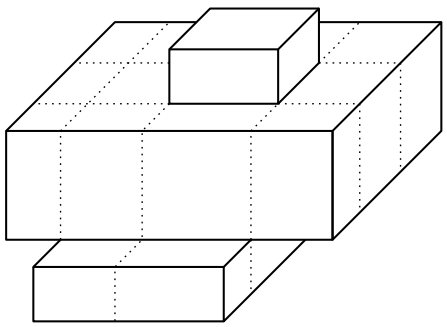

Figure 1.1: The new definition of faces for the purpose of grid-edge unfolding.

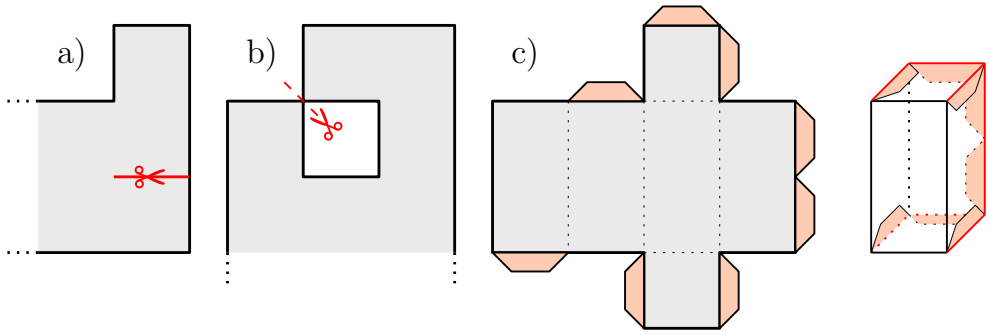

Figure 1.2: a) A part of a net with an overlapping edge. The segment with scissors is supposed to be cut. b) A part of a net with an overlapping vertex. c) A simple net. There is enough space for glue tabs. And the polyhedron folded from the net.

The edge-unfolding does not exist for some non-convex orthogonal polyhedra, for example, a cube with a small hole in the middle of one face. If we allow arbitrary cuts on the surface of the polyhedron, then an unfolding exists for all convex polyhedra [12, Theorem 24.1.2]. We refer to survey papers by O'Rourke [14, 16, 15] for a broader overview.

Grid-edge unzipping is also called a *Hamiltonian unfolding* [17] or an *edge-unzipping* [15]. All Platonic and Archimedean solids have edge unzipping [11]. There exist orthogonal polyhedra with no grid-edge unzipping [10]. The characterization of orthogonal polyhedra that have a grid-edge unzipping remains open.

One could further subdivide each face of the polyhedron using an $a \times b$ orthogonal grid, allowing cuts along these grid lines. This process is termed a *refinement* and is characterized by the parameters a and b, which may also depend on the number of vertices of the polyhedron P. It has been demonstrated that all orthostacks can be unfolded using 1×2 refinement [2] and all genus-0 orthogonal polyhedra can be unfolded using various levels of refinement: exponential refinement [7], quadratic refinement [6], and linear refinement [4]. More recently, Damian, Demaine, Flatland, and O'Rourke have developed an unfolding method for all genus-2 orthogonal polyhedra using only linear refinement [5].

1.2 Thesis outline

An orthostack is an orthogonal polyhedron formed by stacking orthogonal prisms (slabs) on top of each other. In Chapter 2 we explore known results for grid-edge unfolding orthostacks. We describe an algorithm for grid-edge unfolding orthostacks with orthogonally convex slabs by Damian and Meijer [8]. We add a few clear pictures illustrating the notation and the steps and indicate why it is not sufficient for slabs of arbitrary height. Then we show an algorithm to grid-edge unfold rectangle-faced orthostacks (orthostacks whose every component of faces parallel to the xy plane is a rectangle) by Chambers, Sykes, and Traub [3]. We present a modification for grid-edge unfolding box towers (orthostacks with rectangular slabs).

In Chapter 3 we present our main result.

Theorem 1.1. *Every box tower has a simple grid-edge unzipping.*

To prove Theorem 1.1 we provide an algorithm for the unzipping and prove its correctness in Chapter 3.

In Section 3.6 we present a modification of the unzipping for grid-edge unfolding a class of orthogonal polyhedra obtained by gluing two or more box towers on top of another box tower.

Chapter 2

Grid-edge unfolding orthostacks

Let $P \subseteq \mathbb{R}^3$ be an orthogonal polyhedron. Let $z_0, z_1, \ldots, z_n$ be all distinct z-coordinates of its vertices and assume that $z_0 < z_1 < z_2 < \cdots < z_n$. The *slab* S_i is the part of the polyhedron with z-coordinates between z_i and z_{i+1}. The orthogonal polyhedron P is called an *orthostack* if each slab is a prism whose base is a simple polygon. An orthostack is called a *box tower* if its slabs are axis-parallel boxes.

Biedl et al. [2] proved that orthostacks could be unfolded while allowing the cuts along segments from the grid-edge unfolding and also along segments in horizontal planes with z-coordinates $(z_i + z_{i+1})/2$.

2.1 Orthostacks with orthogonally convex slabs

Damian and Meijer [8] studied orthostacks with orthogonally convex slabs. An *orthogonally convex slab* is a slab of an orthogonal polyhedron P where the intersection with any line parallel to the x or y-axis is either empty or a single line segment. They presented an algorithm for grid-edge unfolding of such orthostacks with an additional restriction stating that the boundary of each part of the top boundary of S_i has two orthogonally incident edges that belong to the bottom boundary of the slab S_{i+1}; see Figure 2.1. We describe their algorithm and indicate why it is unsuitable for slabs of arbitrary height.

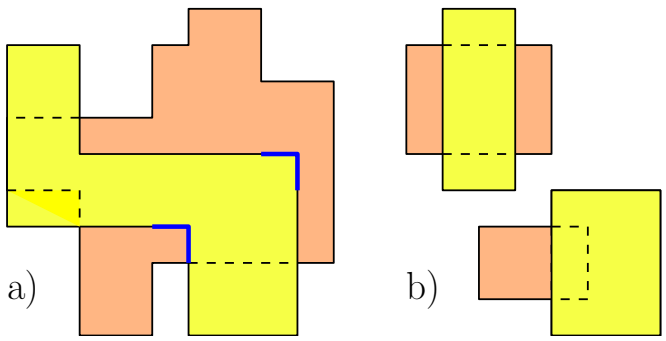

Figure 2.1: a) An orthostack with orthogonally convex slabs considered by Damian and Meijer [8]. The blue lines show the orthogonally incident edges mentioned in their restriction. b) Box towers that do not satisfy the requirements of Damian and Meijer [8].

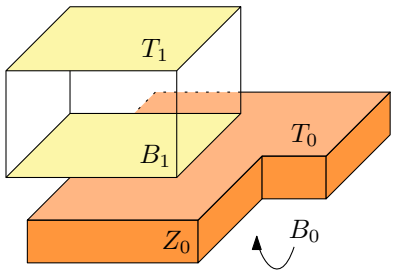

Figure 2.2: An overview of notation.

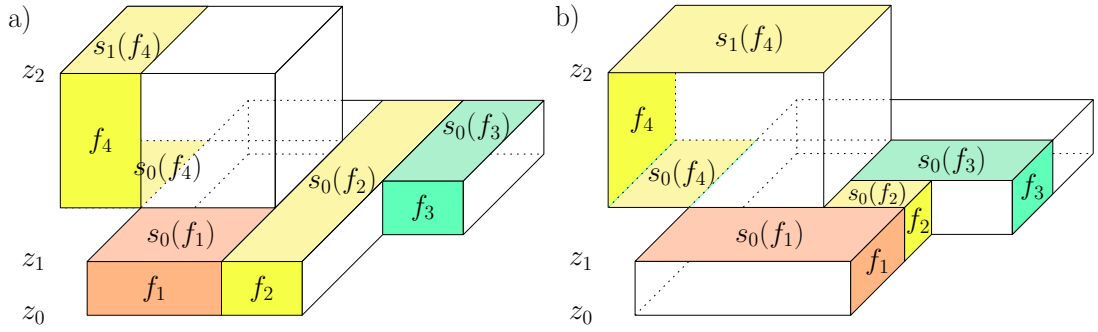

Figure 2.3: a) The definition of strips for the face f parallel to the xz plane. b) The definition of strips for the face f parallel to the yz plane.

In a recently published full version [9] of the extended abstract [8], Damian and Meijer present an algorithm for edge-unfolding polycubes with orthogonally convex layers with no additional restriction. A *polycube* is an orthogonal polyhedron obtained by gluing unit cubes face-to-face. The edges of the cubes are available for cutting for an edge-unfolding. The authors also state that the algorithm works for orthostacks with orthoconvex layers composed of boxes whose height does not exceed the width or depth, but still is not sufficient for slabs of arbitrary height; thus not covering general box towers.

2.1.1 Notation

See Figure 2.2, Figure 2.3, and Figure 2.4 for a summary of this section.

We define T_i to be the top boundary and B_i to be the bottom boundary of the slab S_i (including the interior parts). Faces parallel to the xy plane are called *horizontal*, and faces parallel to the xz plane or the yz plane are called *vertical*. We define Z_i to be the union of vertical faces of S_i.

Assume that a vertical face $f \in Z_i$ is parallel to the xz plane. We define a *strip* $s_i(f)$ to be the set of faces conforming to the following conditions, see Figure 2.3:

(i) the faces are horizontal and have $z = z_{i+1}$

(ii) the projections of the faces on the x-axis are the same as of the face f

(iii) the faces form a connected component adjacent to the face f.

We analogously define $s_{i-1}(f)$ for the faces with $z = z_i$; and for the faces parallel to the yz plane (the condition (ii) uses the y-axis).

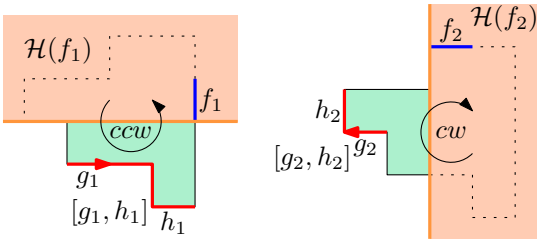

Figure 2.4: A top view of a slab. Half-space $\mathcal{H}(f)$ in its clockwise and counterclockwise variants. The sequence $[g, h]$ is highlighted by the red color.

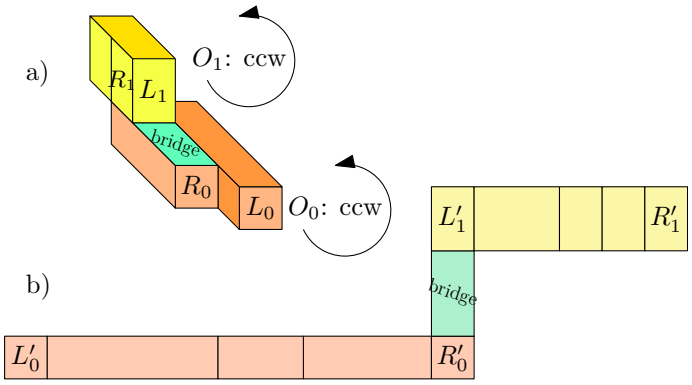

Figure 2.5: a) An orthostack with highlighted important faces. b) The first part of the unfolding of the orthostack from a).

Looking at slab S_i from $z = \infty$ we see Z_i as a simple polygon. We will assign each slab an orientation O_i, either clockwise or counterclockwise. From a given point on the perimeter of the polygon, we create a walk W_i along the whole perimeter in the given orientation O_i (the walk is actually composed of the faces of Z_i). For a vertical face $f \in Z_i$ we define an open half-space $\mathcal{H}(f)$, see Figure 2.4, such that:

(i) it contains the interior of f

(ii) its bounding plane is perpendicular to f

(iii) its bounding plane contains the vertical edge of f that is first encountered on the walk W_i.

For two vertical faces $g, h \in Z_i$ we denote by $[g, h]$ the sequence of all vertical faces of Z_i that are encountered on the walk W_i from g to h, including g and h; see Figure 2.4.

For a given face f, we will denote by f' the corresponding polygon in the constructed planar net.

2.1.2 Algorithm

We arbitrarily choose a vertical face $L_0 \in Z_0$ and an orientation O_0, clockwise or counterclockwise. Next, we proceed inductively.

For every slab S_i (except the topmost one) we find a vertical face $R_i \in Z_i$, a vertical face $L_{i+1} \in Z_{i+1}$, and a bridge. A bridge is a (possibly empty) set of

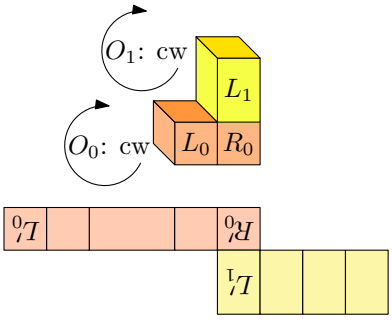

Figure 2.6: An orthostack and the first part of the unfolding. The bridge between R_0 and L_1 is empty. The orientation for the next iteration remains the same. Note that the net is upside down because of the clockwise orientation.

connected horizontal faces with $z = z_{i+1}$ connecting R_i and L_{i+1}. The type of the bridge will determine the orientation for the next step. We place $[L_i, R_i]$ on the net so that it forms a rectangle whose left edge corresponds to a vertical edge of L_i. We place the bridge above/below R'_i and attach L'_{i+1} to it. See Figure 2.5 for an illustration. We will handle the remaining faces of the slab S_i later.

Finding R_i, L_{i+1}, and the bridge between them

Every face f of Z_i except L_i is considered a candidate for R_i. The walk W_i starts at L_i and contains all the candidate faces. For each candidate f, we test for the existence of a suitable bridge and corresponding L_{i+1}. If we cannot find such a bridge and L_{i+1}, f is discarded as a candidate. From the remaining candidates, we select R_i as the last face on W_i.

If a candidate f is adjacent to the band Z_{i+1}, then we choose L_{i+1} as the face adjacent to f, see Figure 2.6. The bridge between f and L_{i+1} is empty in this case. The orientation for the next iteration remains the same.

If a candidate f is not adjacent to the band Z_{i+1}, we try to find a connected component of horizontal faces with $z = z_{i+1}$ that connects f with $L_{i+1} \in Z_{i+1}$. We need L_{i+1} to be parallel to f so that the band Z_{i+1} can also be unfolded in the increasing x direction. We will need almost all the space straight above/below $[L_i, R_i]'$ for the remaining faces of S_i, so the bridge in the net should go only in the increasing x direction. This is fulfilled if the bridge belongs to $\mathcal{H}(f)$. Briefly, we search for a face L_{i+1} that is parallel to f and belongs to $\mathcal{H}(f)$, and a component of horizontal faces with $z = z_{i+1}$ that connects f with L_{i+1} and also belongs to $\mathcal{H}(f)$; see Figure 2.7 for some examples.

There could be several L_{i+1} and corresponding bridges. We introduce two parameters for the choice of L_{i+1}; see Figure 2.7. The first parameter is the distance to the plane bounding $\mathcal{H}(R_i)$; we choose the closest one. The second parameter is the distance to R_i; again we choose the closest one. However, there might be two best ones according to these parameters; see Figure 2.8. We can choose either of them.

We will now choose the bridge. Assume that R_i is parallel to the xz plane and its projection to the x-axis is $[x_{R,1}, x_{R,2}]$ and the projection of L_{i+1} to the x-axis is $[x_{L,1}, x_{L,2}]$, $x_{R,1} \leq x_{L,1}$. As their bridge, we choose the union of the strip $s_i(R_i)$ and the strip $s_i(L_{i+1})$. If $x_{R,2} < x_{L,1}$, then we add also every horizontal face with

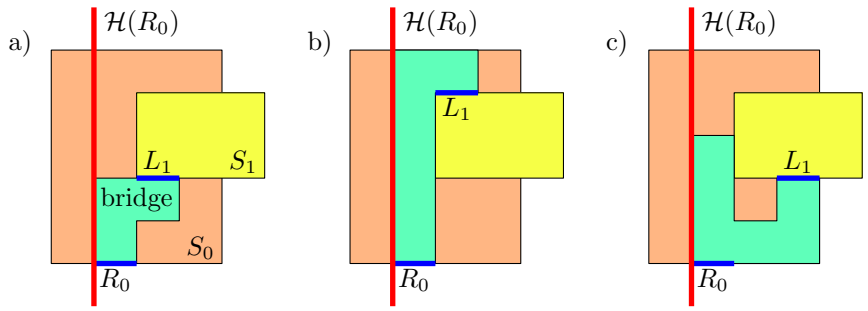

Figure 2.7: Pictures show only some possibilities of L_1 and the bridge, not the final choice. The first parameter is the distance to the plane bounding $\mathcal{H}(R_0)$; we choose a) or b) over c). The second parameter is the distance to R_0; we choose a) over b).

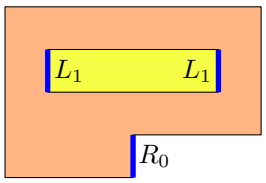

Figure 2.8: Two best options of L_1 according to the parameters.

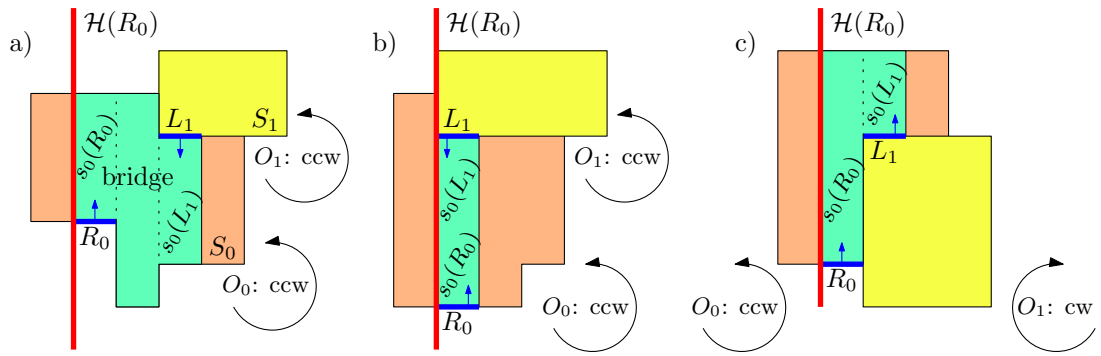

Figure 2.9: The chosen R_0, L_1, and the bridge between them. a) and b) The strips have opposite directions, so the orientation O_1 remains the same as O_0. c) The strips have the same direction, so the orientation O_1 changes from O_0.

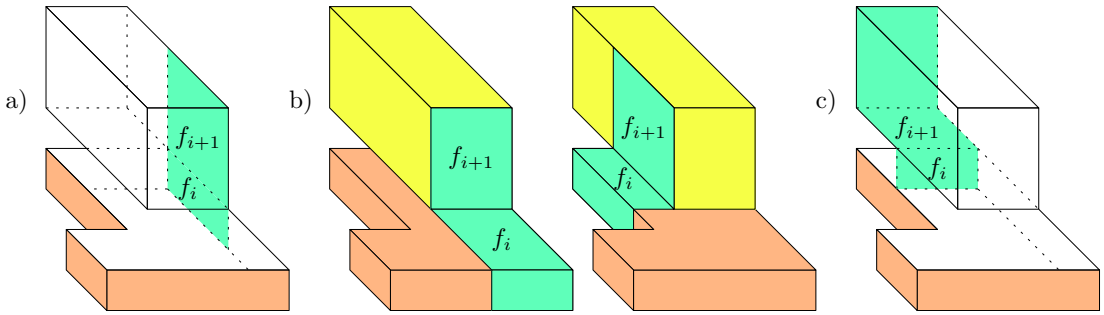

Figure 2.10: An orthostack; faces R_0, L_1, and the bridge between them are green. We show the four situations where the bridge always exists. a) The bridge is empty. b) The bridge is not empty. c) The bridge is at the bottom of the upper slab.

$z = z_{i+1}$ in their component whose projection to the x-axis is in $[x_{R,2}, x_{L,1}]$; see Figure 2.9. The continuity of such a union of faces is ensured by the choice of L_{i+1} based on the previously mentioned parameters.

If the two strips have the same direction from their corresponding vertical faces (for example, both go in the increasing x direction), the orientation O_{i+1} changes from O_i. If the strips have different directions (for example, one goes in the increasing y direction and the other goes in the decreasing y direction), the orientation O_{i+1} remains the same as O_i.

Note that the bridge can also be in the bottom B_{i+1} of the slab S_{i+1}.

Observe that R_i, L_{i+1}, and a bridge exist for every pair of adjacent slabs S_i and S_{i+1}. Because the slabs are adjacent, there exists a face f_i in S_i adjacent to a face f_{i+1} in S_{i+1}. See Figure 2.10 for an illustration. If both faces are vertical, the situation is simple and with an empty bridge. Otherwise one of the faces is vertical and the other is horizontal, let f_{i+1} be vertical. The strip $s_i(f_{i+1})$, containing f_i, will be the bridge. Its one end is $f_{i+1} \in Z_{i+1}$ and its other end surely belongs to Z_i because the slabs are orthogonally convex and it contains f_i. So we can choose the first end as L_{i+1} and the other as R_i. There are four different directions in which such R_i and L_{i+1} can face: rightward, forward, leftward, and backward. This gives us four unique suitable R_i and L_{i+1}. We cannot choose R_i to be L_i, so we might exclude one of the four pairs. We are left with at least three possibilities. That concludes the proof of the existence of a suitable R_i, L_{i+1}, and a bridge for every pair of adjacent slabs.

Unfolding the remaining faces of S_i

We have already placed $[L_i, R_i]$, L_{i+1}, and the bridge between them on the net. There is space above and below $[L_i, R_i]'$, we will place the remaining faces of S_i there.

If $Z_i = [L_i, R_i]$, the unfolding is straightforward; see Figure 2.11 for the unfolding. We divide T_i into strips parallel to $s_i(R_i)$. Every strip is exactly $s_i(f)$ for some $f \in Z_i$ (if a strip was not $s_i(f)$ for some $f \in Z_i$, its both ends would be adjacent to Z_{i+1}; that is not possible as S_{i+1} is orthogonally convex). Note that the bridge is a union of strips within T_i parallel to $s_i(R_i)$, so it cannot interfere with other strips or be adjacent to the ends of the other strips. We can attach

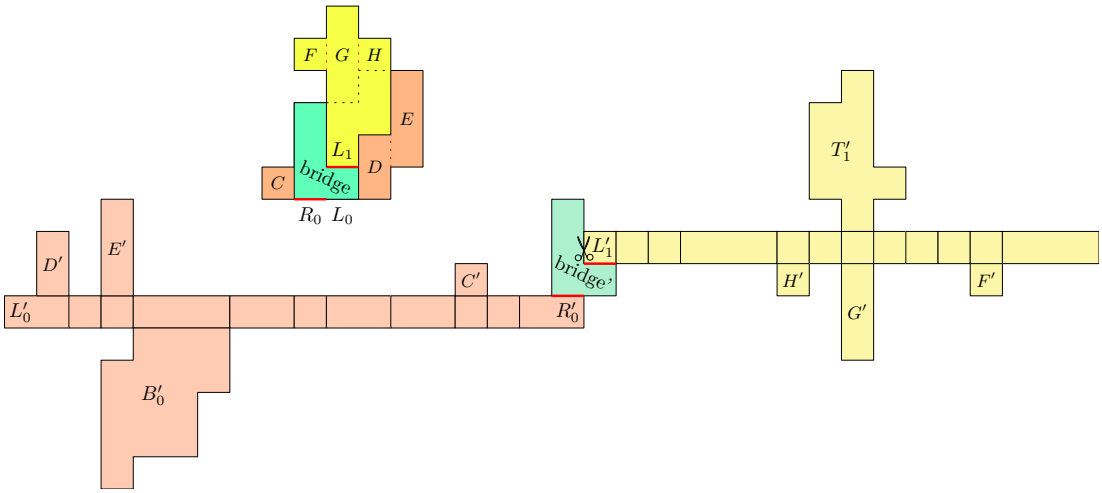

Figure 2.11: An orthostack; faces R_0 and L_1 are highlighted with thick red segments, and the bridge between them is green. The unfolding for $Z_0 = [L_0, R_0]$.

the strip to f', there is enough space for it. We analogously divide B_i into strips parallel to $s_{i-1}(L_i)$ and attach them to the band $Z'_i = [L_i, R_i]'$. Note that B_0 and T_{n-1} can be attached arbitrarily to the bands Z'_0 and Z'_{n-1}.

If $Z_i \neq [L_i, R_i]$, some band faces are still not unfolded. We cannot place them next to $[L_i, R_i]'$ as it will interfere with the space above or below an adjacent unfolded band. Note that for every band face $f \in Z_i$ adjacent to a bottom face in B_{i+1} there exists a bridge, it is $s_i(f)$ (the second end of the strip is adjacent to the upper band because the slabs are orthogonally convex). Also for band faces adjacent to the upper band there exists an empty bridge. Since we chose R_i to be the last face on the walk W_i for which a bridge exists, and because a bridge exists for every band face adjacent to B_{i+1} or Z_{i+1}, all the band faces not in $[L_i, R_i]$ are adjacent to T_i. Thanks to the restriction stating that the boundary of each component of T_i has two orthogonally incident edges that belong to B_{i+1}, it is easier to find a bridge in T_i.

We divide the faces of T_i into two sets, χ and $T_i \setminus \chi$. The set χ consists of all faces in T_i between a plane containing L_i and a plane parallel to it containing a vertical edge of R_i closer to L_i, see Figure 2.12. We divide χ into strips I perpendicular to $s_i(L_i)$. For each face $f \in Z_i \setminus [L_i, R_i]$ perpendicular to L_i there is a strip $s_i(f) \in I$, let g be the vertical face adjacent to the other end of $s_i(f)$, so that $s_i(f) = s_i(g)$. The strip $s_i(f)$ cannot interfere with the bridge from the choice of I and the bridge. The face g cannot lie in Z_{i+1} as it would form a valid bridge with $R_i = f$ further on the walk W_i than the chosen R_i. The face g cannot lie in $Z_i \setminus [L_i, R_i]$ as $\mathcal{H}(f) \cup \mathcal{H}(g)$ covers the whole T_i thus $\mathcal{H}(f)$ or $\mathcal{H}(g)$ contains a valid bridge. Then $g \in [L_i, R_i]$. We place $s_i(g)$ next to g' and f next to $s_i(g)'$.

All faces in $Z_i \setminus [L_i, R_i]$ perpendicular to L_i are placed next to the strips I'. All remaining faces are parallel to L_i.

Let $f \in Z_i \setminus [L_i, R_i]$ be a remaining face. Consider the case where f is adjacent to two already unfolded faces $g_1, h_1 \in Z_i \setminus [L_i, R_i]$; see Figure 2.12. Assume g_1 is earlier on W_i then h_1. Let $g_2, h_2 \in [L_i, R_i]$ be the other ends of $s_i(g_1), s_i(h_1)$. We see that h_2 is earlier on W_i than g_2. The strips $s_i(g), s_i(h)$ are adjacent strips in T_i, so between $s_i(h_2)'$ and $s_i(g_2)'$ according to the x-coordinate there are only

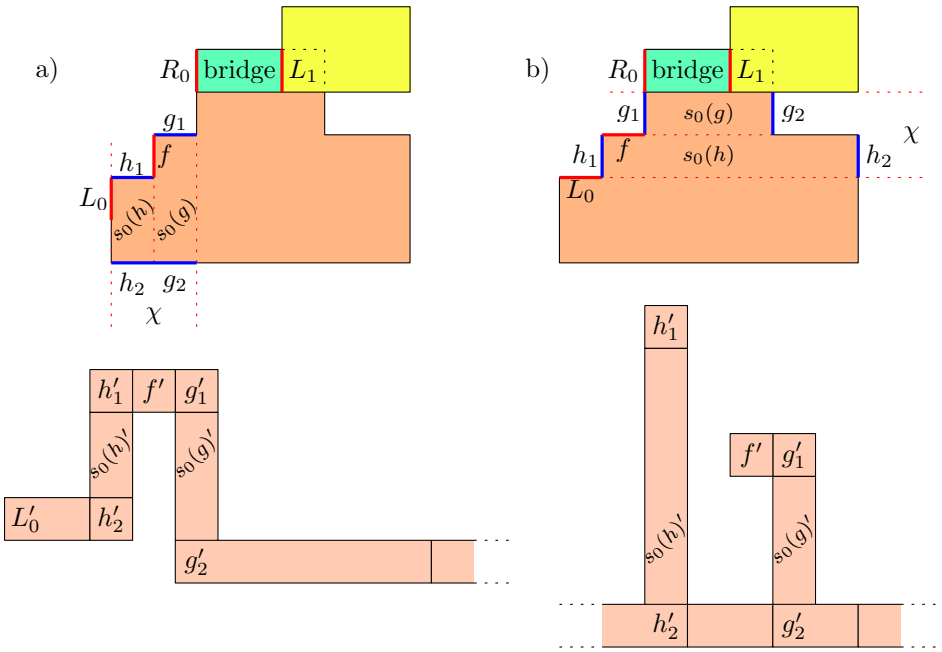

Figure 2.12: a) We cut the band and reconnect the parts with h_1', f', g_1'. b) We place f' next to g_1'.

band faces $A' = [h_2, g_2]' \setminus \{h_2, g_2\}'$. If the projection of A' to the x-axis is longer than (or of equal length as) f when viewed from $z = \infty$, we place f next to g_1'. Otherwise we cut $[L_i, R_i]$ between h_2 and g_2 and reconnect the parts via h_1', f', g_1'. The remaining faces in T_i and B_i are handled similarly as earlier.

A problem arises when a remaining face $f \in Z_i$ is adjacent to L_i or R_i; see Figure 2.13. We indicate, not prove, why we cannot add the face f to the net.

- We cannot easily place f next to another face or do the cutting and reconnecting as in the previous case.
- Changing the set χ does not help for some symmetrical configurations.
- Since each slab has an arbitrary height, face f has one unlimited dimension. If we position f on the net such that the unlimited dimension extends in the x direction, it might overlap with the space reserved for other slabs.
- The images of slabs S_{i-1} and S_{i+1} on the net can extend arbitrarily in both the positive and negative y directions.
- The direction of the strips in B_i is parallel to the direction of $s_{i-1}(L_i)$. There might be a strip $s_{i-1}(f)$ that we need to attach to f' if the end of the strip is adjacent to Z_{i-1}. In this case, we do not want to place f and $s_{i-1}(f)$ in the space on the net reserved for S_{i-1}. We cannot use the faces from B_i to unfold the remaining faces of Z_i. Instead, we need to first unfold the remaining faces of Z_i before unfolding the faces from B_i.

The full version [9] of this article [8] solves the problem with remaining band faces in all cases, provided the height of the faces does not exceed their width or depth.

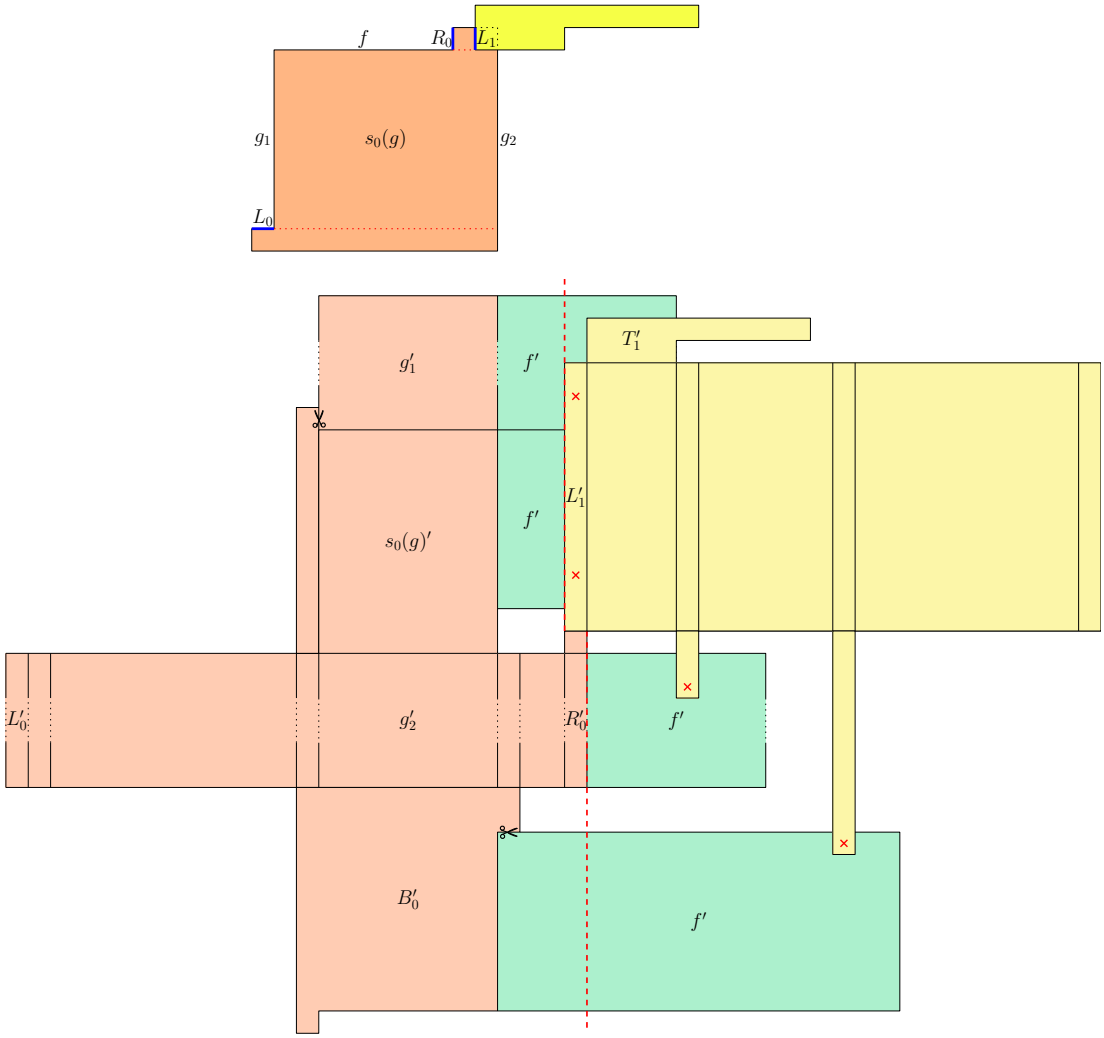

Figure 2.13: We cannot place the face f anywhere. The red crosses indicate overlaps. The green rectangles are possibilities for placing the face f. The red dashed line separates space for slab S_0 and S_1. Faces of S_0 should be placed to the left of the red dashed line. Face f cannot interfere with space for S_1.

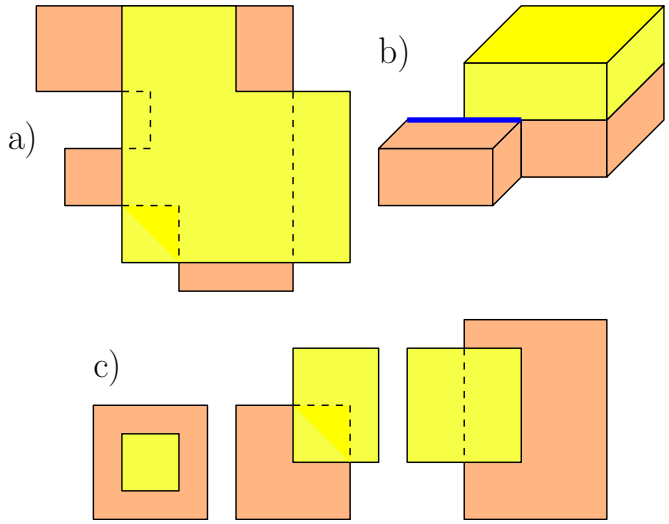

Figure 2.14: a) A top view of a rectangle-faced orthostack satisfying both restrictions (i) and (ii) considered by Chambers, Sykes, and Traub [3]. b) An orthostack violating restriction (ii); the thick (blue) line highlights an edge that partially lies in the side boundary of the top slab and partially in the side boundary of the bottom slab. c) Top views of box towers violating restriction (i).

2.2 Rectangle-faced orthostacks

In this section, we consider the original definition of horizontal faces, without the subdivision. We only subdivide vertical faces by horizontal planes passing through the vertices of the polyhedron.

Chambers, Sykes, and Traub [3] showed that a grid-edge unfolding exists for a special class of orthostacks, named *rectangle-faced orthostacks*, satisfying the following conditions; see Figure 2.14.

(i) All horizontal faces, except the top face of the topmost slab and the bottom face of the bottommost slab, are rectangles

(ii) Every edge of every rectangular horizontal face lies completely within a side boundary (left, front, right, or back) of an adjacent slab.

We describe their algorithm and modify it for box towers.

2.2.1 Preliminaries

We denote by B_i the band around the slab S_i, that is the union of all faces of S_i parallel to the xz plane or the yz plane. The faces parallel to the xy plane are called *z-faces*. By *z-faces of* z_i, we mean the union of all z-faces where $z = z_i$.

The authors provide some important lemmas that will be used in the algorithm.

- For every z-face f with $z = z_i$ (excluding z_0 and z_n) there exists a pair of its opposite sides, such that one side is contained in the band B_{i-1} and the other side is contained in the band of B_i.

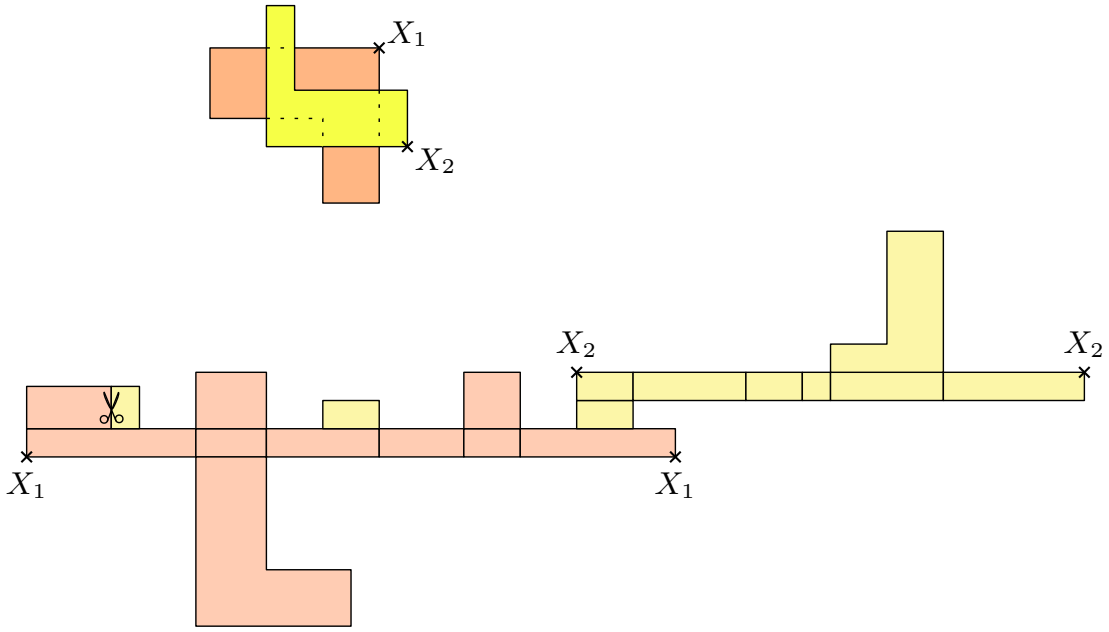

Figure 2.15: The top view of an orthostack and its unfolding. The points X lie on the band cuts. The scissors show a segment where two faces meet but are not adjacent.

- We can order the z-faces of z_i based on adjacency to the band B_{i-1} in counterclockwise order when viewed from $z = \infty$. The ordering is the same for the band B_i.

2.2.2 Algorithm

See Figure 2.15 for an illustration.

We cut the band B_0 arbitrarily and unfold it in a counterclockwise order. We place it on the net so that it goes straight in the increasing x direction. We place the z-face with $z = z_0$ below the unfolded band B_0. We proceed inductively.

For every z-face of z_i, we find a pair of its opposite sides, s_1 and s_2, such that s_1 is contained in the band B_{i-1} and s_2 is contained in the band B_i. We place the z-face on the net above the unfolded band B_{i-1} to match the side s_1. Now s_1 is oriented in the decreasing y direction and s_2 is oriented in the increasing y direction. At least one z-face is present, so we choose the right-most one in the net. Its side s_2 will be matched with the band B_i (its top left point will be matched with the bottom left point of the unfolded band B_i). We unfold the band B_i in a counterclockwise order and place it on the net so that it goes straight in the increasing x direction.

At the end, we place the z-face of z_n above the unfolded band B_{n-1}.

This approach ensures the non-overlap of the faces. It is primarily based on the increasing x direction and secondarily on the increasing y direction.

2.2.3 Modification for box towers

The attachment of the band B_i was chosen to be the right-most unfolded z-face of z_i. It also defined the cut of the band B_i. But we can choose the highest unfolded

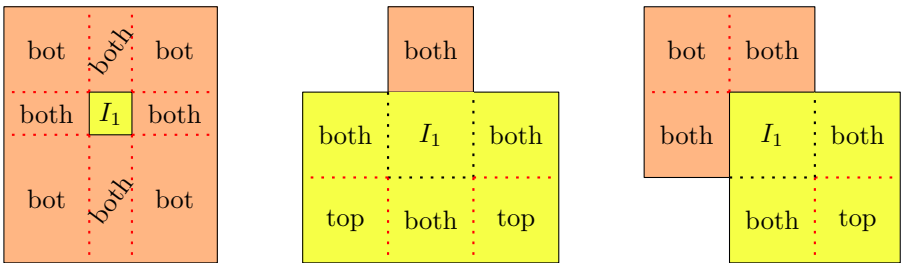

Figure 2.16: Examples of subdivision of z-faces of z_1.

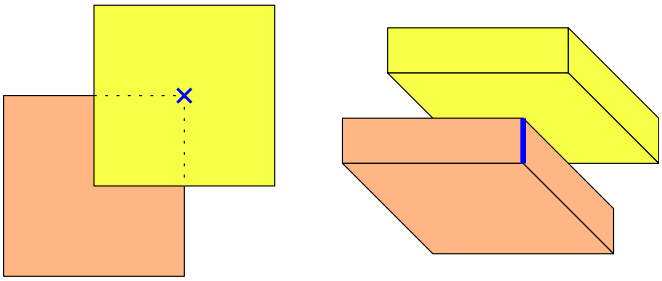

Figure 2.17: The only configuration of two adjacent layers that need a specifically chosen cut. The position of the cut is highlighted by a thick segment (blue). Note that only the bottom slab needs the specifically chosen cut so it does not interfere with other steps.

z-face of z_i (by highest we mean the one with the highest bounding y-coordinate in the net) as the attachment and then cut the band B_i arbitrarily.

We need to split the non-rectangular z-faces into rectangles and find attachment sides for them. We denote by I_i the intersection of the slabs S_{i-1} and S_i; it is a rectangle with $z = z_i$. We divide the z-faces of z_i $(1 \leq i \leq n-1)$ by the lines containing the sides of I_i; see Figure 2.16. From now on, by z-faces, we mean the subdivisions. All the z-faces are rectangles. There are three types of z-faces:

- the *both-band* z-faces, where for one pair of opposite sides, the *bottom-band* side is contained in the band B_{i-1} and the *top-band* side is contained in the band B_i
- the *bottom-band* z-faces, whose two adjacent sides are contained in the band B_{i-1} and the other two adjacent sides are adjacent to both-band z-faces
- the *top-band* z-faces, whose two adjacent sides are contained in the band B_i and the other two adjacent sides are adjacent to both-band z-faces

For every band, we choose a cut. There is only one configuration of two adjacent slabs that needs a specifically chosen cut; see Figure 2.17; the rest can be arbitrary.

We start by unfolding the band B_0 in the counterclockwise order. We attach the z-face of z_0 below the unfolded band B_0. We proceed inductively. In each step, we unfold the z-faces of z_i and the band B_i. There are three cases:

- If there is no top-band z-face, we attach all the bottom-band z-faces above the unfolded band B_{i-1} arbitrarily and all the both-band z-faces above the

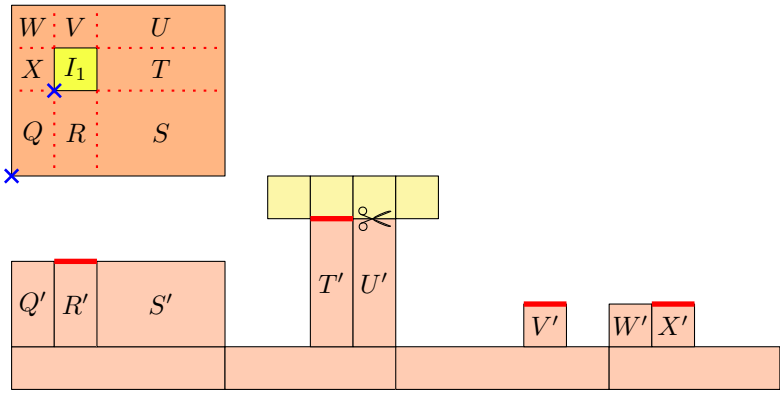

Figure 2.18: Unfolding of two adjacent layers with no top-band z-face. Blue crosses show the cuts of the bands. Thick (red) lines show the possible attachment sides to the upper band.

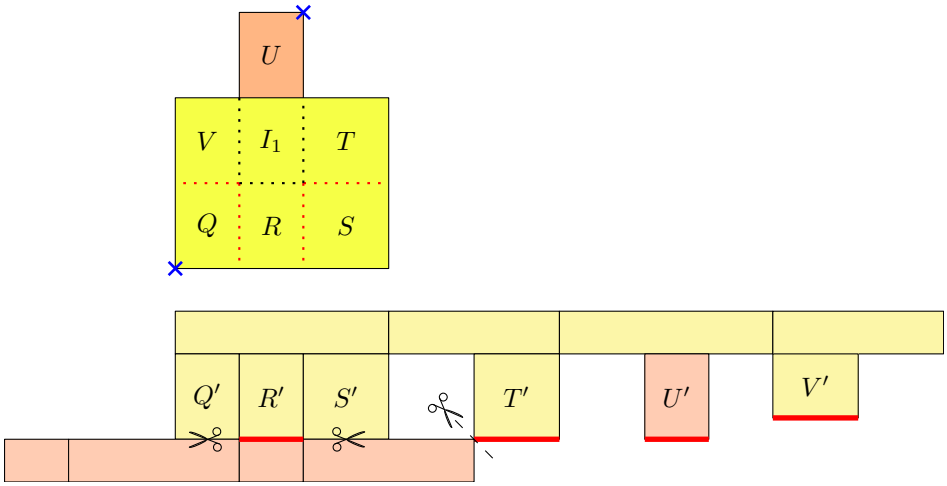

Figure 2.19: Unfolding of two adjacent layers with no bottom-band z-face. Blue crosses show the cuts of the bands. Thick (red) lines show the possible attachment sides to the bottom band.

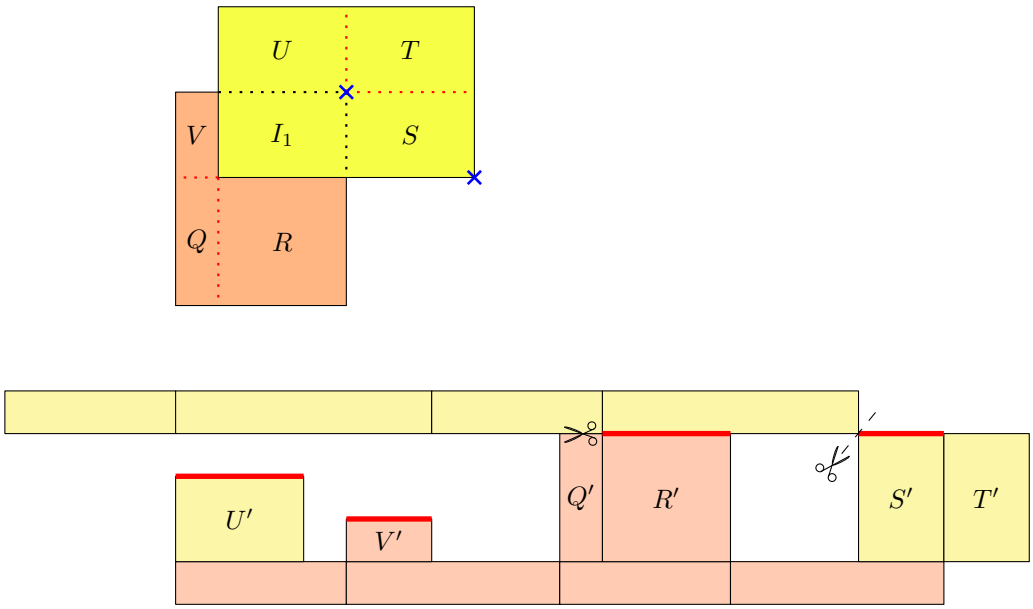

Figure 2.20: The only configuration of two adjacent layers with a bottom-band z-face and a top-band z-face. Blue crosses show the cuts of the bands. Thick (red) lines show the possible attachment sides to the upper band.

unfolded band B_{i-1} by the bottom-band side. For every bottom-band z-face there is a both-band z-face of the same height. One of the highest rectangles of all the unfolded z-faces of z_i must be a both-band z-face. We unfold and attach the band B_i to this rectangle. See Figure 2.18 for an example.

- If there is no bottom-band z-face, we attach all the z-faces below the band B_{i+1} in the same way we attached all the z-faces above the band B_i in the previous step. We attach this by the highest rectangle to the band B_i. See Figure 2.19 for an example.

- If there is a bottom-band z-face and a top-band z-face, it is a special case. We attach the z-faces as shown in Figure 2.20. Again, one of the both-band z-faces must be one of the highest rectangles of all the unfolded z-faces of $z = z_i$. We unfold and attach the band B_{i+1} to this rectangle.

At the end, we attach the z-faces of z_n above the unfolded band B_n.

The algorithm is based on the increasing y direction and thus no overlap of the faces is ensured.

2.2.4 H-convex Manhattan towers

Andres, Largeteau-Skapin, Richaume, and Zrour [1] presented an algorithm for grid-edge unfolding H-convex Manhattan towers; see Figure 2.21 for an example. We create an orthogonal prism by gluing unit cubes face-to-face so that the prism is convex in the x-direction, that is the base. We then glue a stack of cubes on top of every base cube.

They also presented an extension for grid-edge unfolding Up-and-Down orthoterrains; see Figure 2.21 for an example. We have some stacks of cubes par-

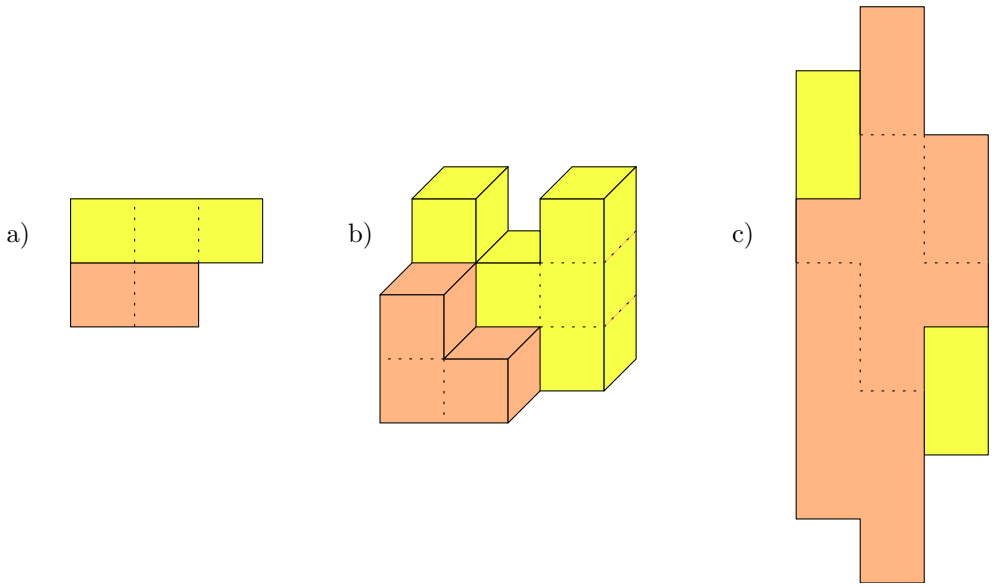

Figure 2.21: a) A base, convex in y-direction. b) H-convex Manhattan towers, convex in z-direction. c) A front view of an Up-and-Down orthoterrain, convex in z-direction.

allel to the z-axis and glue them together to form a single polyhedron with the following restrictions. From $z = \infty$ the polyhedron can be seen as a rectangle. Every two neighboring cube stacks have to be face-connected.

Note that H-convex Manhattan towers and Up-and-Down orthoterrains are subclasses of polycubes and orthostacks. Each slab is formed by cubes with the same y-coordinate.

The algorithms are similar to unfolding rectangle-faced orthostacks. We create bands from faces parallel to the xy and yz planes and unfold them into the plane. Faces parallel to the xz are placed between the unfolded bands.

Chapter 3

Grid-edge unzipping box towers

3.1 Notation

Faces parallel to the xy plane are called *horizontal*, faces parallel to the xz plane are called *front-back*, and faces parallel to the yz plane are called *left-right*.

Since each slab S_i is an axis-parallel box, we can express it as a Cartesian product

$$S_i = [x_{i,1}, x_{i,2}] \times [y_{i,1}, y_{i,2}] \times [z_i, z_{i+1}] .$$

Let E_i be the union of all the horizontal faces of P with z-coordinate z_i (it consists of the top faces of the slab S_{i-1} and the bottom faces of the slab S_i). Let L_i be the union of the left-right faces in S_i with x-coordinate $x_{i,1}$ (it is the left rectangular boundary of S_i). Similarly, let R_i be the union of the left-right faces in S_i with x-coordinate $x_{i,2}$, let F_i be the union of the front-back faces in S_i with y-coordinate $y_{i,1}$, and let B_i be the union of the front-back faces in S_i with y-coordinate $y_{i,2}$. The surface of P is exactly the union of E_i for $0 \leq i \leq n$ and of L_i, R_i, F_i and B_i for $0 \leq i < n$.

We subdivide E_i for $i \in \{1, 2, \ldots, n-1\}$, only E_0 and E_n remain untouched. We denote by $E_{i,L}$ the following subset of E_i (see Figure 3.1):

$$\begin{aligned} E_{i,L} = \{&(x, y, z_i) \in E_i; \\ &x \in [\min(x_{i-1,1}, x_{i,1}), \max(x_{i-1,1}, x_{i,1})] \\ &\wedge\ y \in [y_{i-1,1}, y_{i-1,2}] \cap [y_{i,1}, y_{i,2}]\}. \end{aligned}$$

Similarly we define $E_{i,R}$, $E_{i,F}$ and $E_{i,B}$:

$$\begin{aligned} E_{i,R} = \{&(x, y, z_i) \in E_i; \\ &x \in [\min(x_{i-1,2}, x_{i,2}), \max(x_{i-1,2}, x_{i,2})] \\ &\wedge\ y \in [y_{i-1,1}, y_{i-1,2}] \cap [y_{i,1}, y_{i,2}]\}, \end{aligned}$$

$$\begin{aligned} E_{i,F} = \{&(x, y, z_i) \in E_i; \\ &y \in [\min(y_{i-1,1}, y_{i,1}), \max(y_{i-1,1}, y_{i,1})]\}, \end{aligned}$$

$$\begin{aligned} E_{i,B} = \{&(x, y, z_i) \in E_i; \\ &y \in [\min(y_{i-1,2}, y_{i,2}), \max(y_{i-1,2}, y_{i,2})]\}. \end{aligned}$$

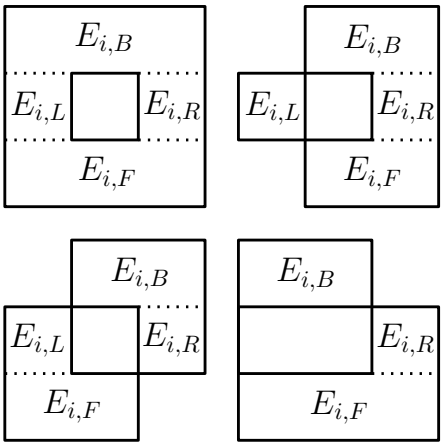

Figure 3.1: The rectangles are the projections to the xy plane of two adjacent slabs; the pictures do not distinguish which slab is above the other one. Dotted lines show the subdivision of E_i into $E_{i,F}$, $E_{i,R}$, $E_{i,B}$, $E_{i,L}$ and are the only places where we cut across an original face of the polyhedron. Pictures show only some cases of how two consecutive slabs can interact.

Note that $E_{i,L}, E_{i,R}, E_{i,F}, E_{i,B}$ are pairwise internally disjoint. Observe that each of these sets is either empty or a rectangle contained either in the top boundary of S_{i-1} or in the bottom boundary of S_i. The lines between $E_{i,L}, E_{i,R}, E_{i,F}, E_{i,B}$ are the only places where we cut across an original face of the polyhedron.

3.2 Algorithm

Let P be a box tower. For a given subset A of the surface of P, we will denote by A' the corresponding subset in the constructed planar net.

We divide the algorithm into three phases, see Figure 3.2 for the division of faces into phases. We start with projecting E_0 orthogonally to the xy-plane. Each phase unfolds a part of the box tower. See Figure 3.3 for the resulting net. The cutting segments will be clear from the process and described in detail in Section 3.3.

3.2.1 Right-left phase

In this phase we unfold all the rectangles R_i, $E_{i,R}$, L_i, $E_{i,L}$ and E_n. Let RE be the union of all the rectangles R_i and $E_{i,R}$, and let LE be the union of E_n and all the rectangles L_i and $E_{i,L}$.

We start with placing all rectangles from RE. We place R_0 to the right of E'_0, then $E_{1,R}$ to the right of R'_0, then R_1 to the right of $E'_{1,R}$, and we continue placing R_i and $E_{i,R}$, for $i = 2, 3, \ldots, n-1$, always to the right of the previous one. We then place to the right the whole rectangle E_n.

We continue placing the remaining rectangles from LE. We place L_{n-1}, $E_{n-1,L}, L_{n-2}, E_{n-2,L}, \ldots, L_0$ in this order, always to the right. Clearly, in the resulting net no vertices, edges, or faces overlap so far.

Note that the y-coordinates of the rectangles are preserved in this phase; they are the same in the box tower and the net.

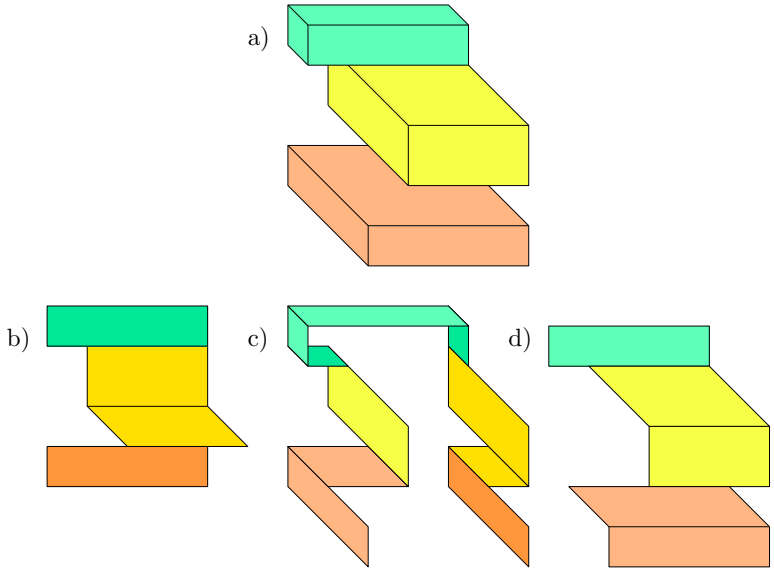

Figure 3.2: a) A box tower with three slabs. b) All the faces unfolded during the back phase. c) All the faces unfolded during the right-left phase. d) All the faces unfolded during the front phase.

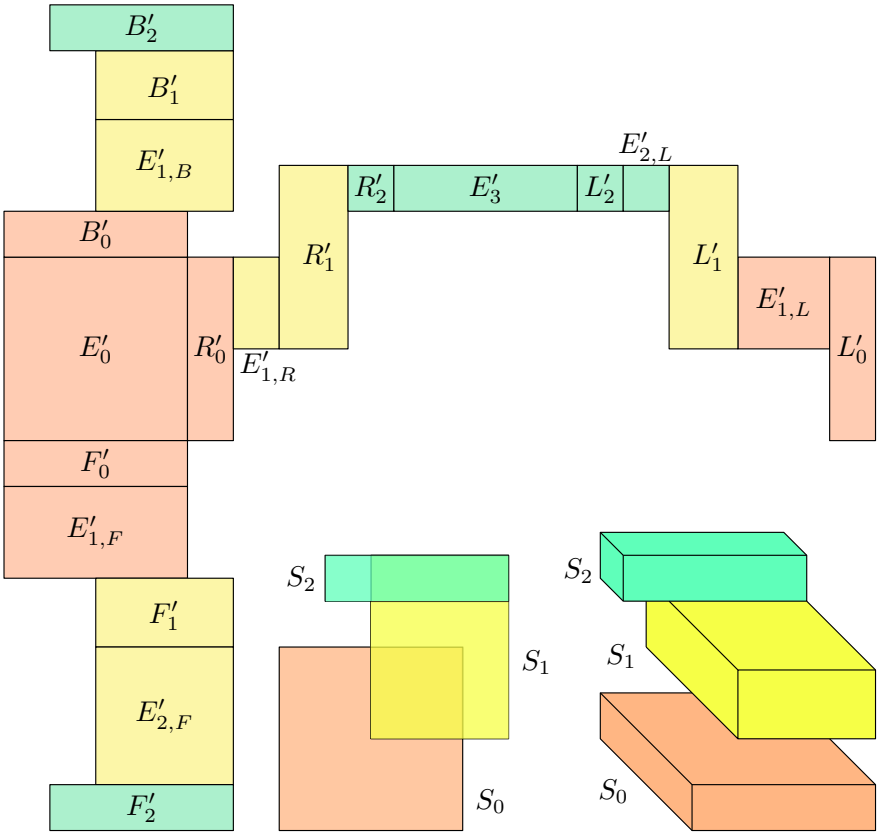

Figure 3.3: A box tower with three slabs and its net resulting from the unfolding algorithm.

3.2.2 Back phase and front phase

Now we describe the second and the third phases. We denote the union of all the rectangles B_i and $E_{i,B}$ by BE and the union of all the rectangles F_i and $E_{i,F}$ by FE.

In the second phase, the back phase, we will be placing the rectangles of BE in the direction of increasing y-coordinate. We place B_0 above E'_0. Then we proceed with $E_{1,B}, B_1, E_{2,B}, B_2, \ldots, E_{n-1,B}, B_{n-1}$ in this order, placing each rectangle always above the previous one. The rectangle E_n has already been placed in the right-left phase.

In the third phase, the front phase, we will be placing the rectangles of FE in the direction of decreasing y-coordinate. We place F_0 below E'_0. Then we proceed with $E_{1,F}, F_1, E_{2,F}, F_2, \ldots, E_{n-1,F}, F_{n-1}$ in this order, placing each rectangle always below the previous one.

In the second and third phases, the x-coordinates of the rectangles are preserved.

3.3 Proof of non-overlap

Each phase on its own creates a simple non-overlapping polygon because of the continuous one-directional process. For a similar reason, rectangles from the back phase and the front phase cannot overlap. We will prove that no rectangle from the back phase can overlap or touch with a rectangle from the right-left phase. The proof for the rectangles from the front phase and the right-left phase would be analogous.

We will define a piecewise linear curve C formed by a subset of the cutting segments on the surface of P, which will partially separate rectangles from the right-left phase and the back phase, see Figure 3.4. The curve C starts at $(x_{0,2}, y_{0,2}, z_0)$, which is the bottom right back corner of S_0, continues in the z direction to $(x_{0,2}, y_{0,2}, z_1)$, which is the top right back corner of S_0, then separates $B_0 \cup E_{1,B} \cup B_1$ from $R_0 \cup E_{1,R} \cup R_1$ in the x and y directions, reaching the point $(x_{1,2}, y_{1,2}, z_1)$, which is the bottom right back corner of S_1. The curve proceeds analogously until C reaches $(x_{n-1,2}, y_{n-1,2}, z_n)$, which is the top right back corner of the topmost slab S_{n-1}.

The curve C maps onto two piecewise linear curves in the net. One curve, denoted by Q, forms the right part of the perimeter of BE', and the second curve, denoted by R, forms the top part of the perimeter of RE'.

Our goal is to prove that the curves Q and R do not intersect.

When the curve C cuts between B_i and R_i, the corresponding part of the curve R moves in the increasing x-direction by $z_{i+1} - z_i > 0$ and the corresponding part of the curve Q moves in the increasing y-direction by $z_{i+1} - z_i > 0$. If C moves in the x-direction by d, then R moves to the right by d and Q moves to the left or right (preserving the former direction of C) by d. If C moves in the y-direction by d, then R moves up or down (preserving the former direction of C) by d and Q moves up by d.

The curves R and Q start diverging just after their common starting point when C cuts between B_0 and R_0: the curve R moves by $z_1 - z_0$ in the increasing x-direction and the curve Q moves by $z_1 - z_0$ in the increasing y-direction.

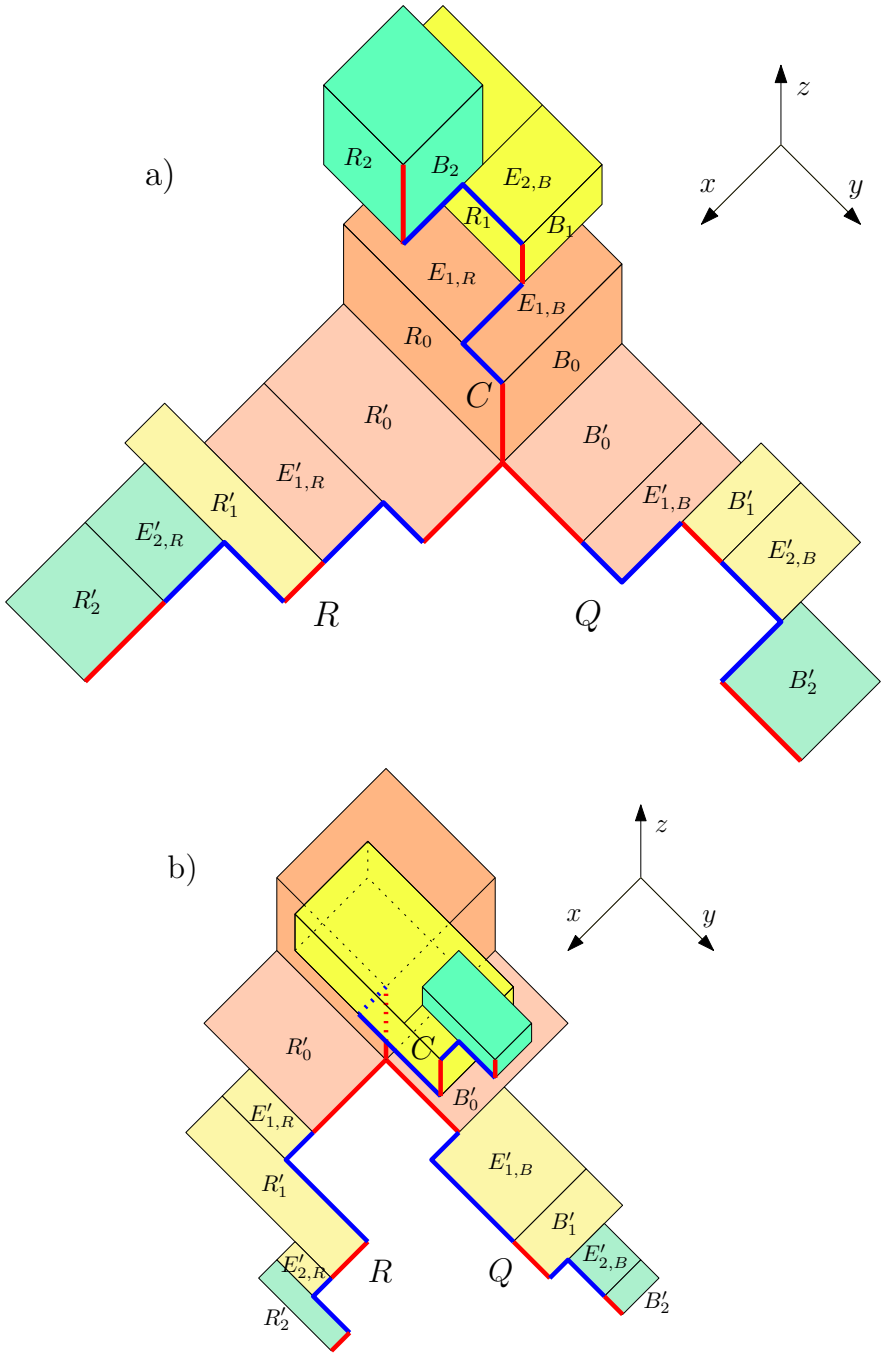

Figure 3.4: Box towers with a clear view of the right and the back part, the cut C and its images Q and R. In both pictures red lines separate B_i from R_i in the z direction and blue lines separate $B_i \cup E_{i+1,B} \cup B_{i+1}$ from $R_i \cup E_{i+1,R} \cup R_{i+1}$ in the x and y directions. a) Blue lines separate $E_{1,B}$ from R_0 in the y direction, $E_{1,B}$ from $E_{1,R}$ in the x direction, and then separate $E_{2,B}$ from R_1 in the y direction and B_2 from $E_{2,R}$ in the x direction. b) Blue lines separate $E_{1,B}$ from $E_{1,R}$ in the x direction, $E_{1,B}$ from R_1 in the y direction, and then separate B_1 from $E_{2,R}$ in the x direction and $E_{2,B}$ from R_2 in the y direction.

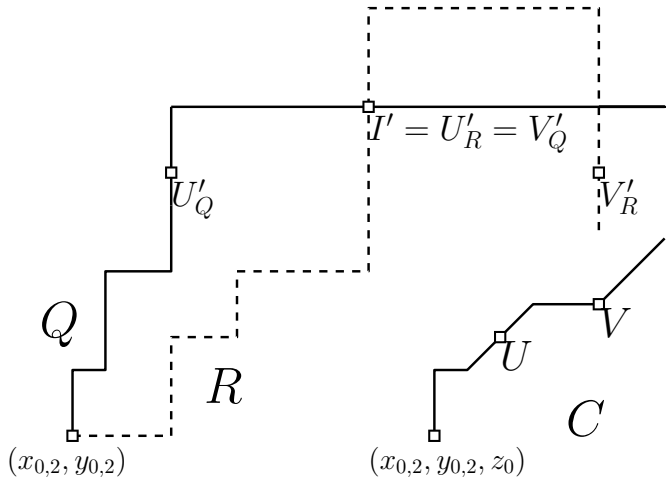

Figure 3.5: The cut C and vertices for proof.

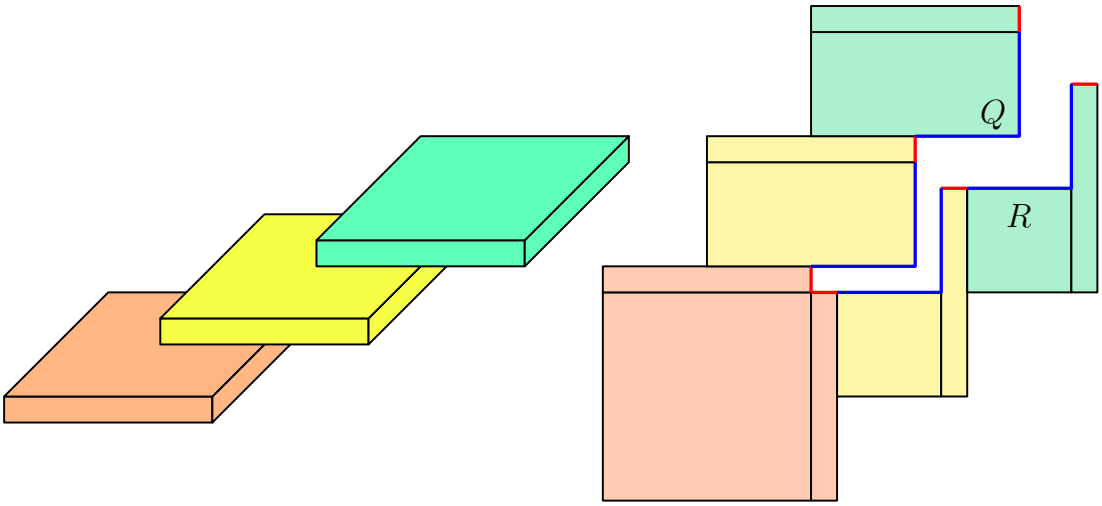

Figure 3.6: The curves R and Q might be very close.

For every point A on the curve C we denote by A'_R the image of A on the curve R and we denote by A'_Q the image of A on the curve Q.

From the iterative process of constructing Q and R, we deduce the following.

Lemma 3.1. *Let A be a point on the curve C except the starting point and assume that $A'_R = (x_R, y_R)$ and $A'_Q = (x_Q, y_Q)$. Then $x_R > x_Q$ and $y_R < y_Q$.* □

Lemma 3.2. *The curves R and Q do not intersect nor touch, except at the starting point $(x_{0,2}, y_{0,2})$.*

Proof. For a proof by contradiction suppose the curves Q and R intersect, see Figure 3.5. Denote by I' the point of the first intersection of Q and R (except the starting point). The point I' corresponds to a point U on C from the perspective of R and also corresponds to a point V on C from the perspective of Q (that is, $U'_R = V'_Q = I'$). By Lemma 3.1 we have $U \neq V$. Assume, without loss of generality, that U appears on C before V. Then U'_Q is before V'_Q on Q. From Lemma 3.1 the y-coordinate of U'_Q is greater than that of U'_R. Since Q is y-monotone and U is before V on C, the y-coordinate of U'_Q is smaller than or equal

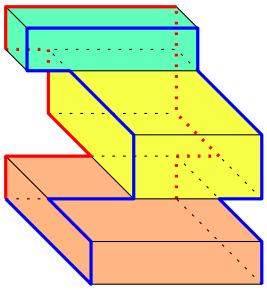

Figure 3.7: The cutting path of the grid-edge unzipping produced by the algorithm. The red line shows the first part and the blue line shows the second part.

to the y-coordinate of $U'_R = V'_Q$. These two inequalities imply a contradiction. Note that R and Q might be very close, see Figure 3.6. □

The rectangles of BE' are to the left of the curve Q and the rectangles of RE' are below the curve R. Because the curve Q is to the left and upwards of the curve R and the curves Q and R do not intersect or touch, the rectangles of BE' and RE' cannot overlap.

Lemma 3.3. *All points in BE' are to the left from the rightmost point of RE'. The set of the rightmost points of RE' forms the rightmost edge of RE'.*

Proof. Let S be a point on the curve C such that S'_Q is the rightmost point of Q. Then by Lemma 3.1 the point S'_R is to the right of S'_Q. The second part is clear from the algorithm. □

All the rectangles in LE' are to the right of S'_R, so the rectangles in BE' cannot overlap with the rectangles in LE'. This concludes the proof that rectangles placed in the back phase cannot overlap with the rectangles from the right-left phase.

3.4 Unzipping

If we closely inspect the cutting segments of our algorithm, we can see that they form a single path, see Figure 3.7. The path starts with the cut C separating the right part RE from the back part BE, followed by the back edge of E_n, a cut separating BE from LE, the left edge of E_0, a cut separating LE from FE, the front edge of E_n, and a cut separating FE from RE.

We can take inspiration from alternatives of unzipping a cube [11], see Figure 3.8, to try to rearrange the parts of the net. There is a problem though; we need all four cuts in the z-direction. The alternative unzippings are then unusable for our algorithm and the distribution of rectangles into the phases. Turning the only valid cube unzipping around will not give new nets as we can also turn the original box tower similarly and obtain the corresponding nets from our algorithm.

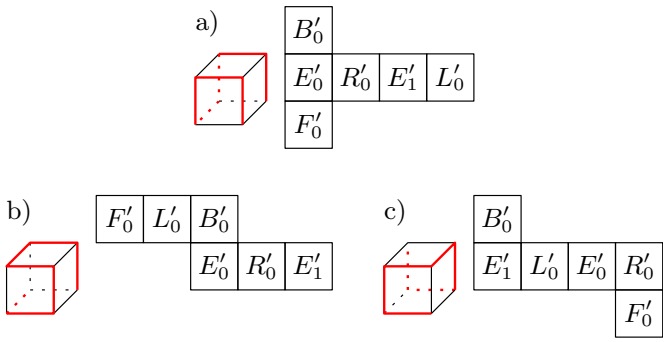

Figure 3.8: a) The unzipping of a cube produced by the algorithm. b) and c) Alternative unzippings of a cube [11].

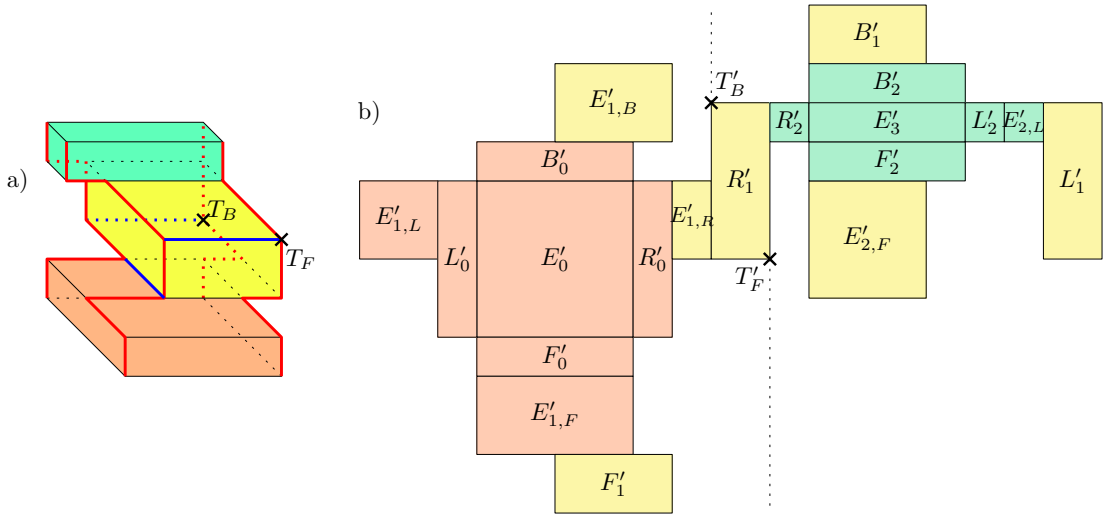

Figure 3.9: a) Red lines show the needed cuts separating the rectangles of different phases. Blue lines show the new cuts. T_F is the point T for the front part, T_B is the point T for the back part. b) The resulting net and the points T'_F and T'_B separating the corresponding parts with their x-coordinates.

3.5 Slight net modifications

We obtain multiple new nets if we consider a grid-edge unfolding and not only an unzipping. We can split the front part FE' horizontally to FE'_1 and FE'_2 and place FE'_2 next to the front side of E'_n. It does not create an overlap as there are still simple cuts separating RE, E_n, and LE from the parts of FE. The parts of FE' cannot overlap each other either. There is a point T in the intersection of the cut separating FE_1 from FE_2 and the cut separating $FE_1 \cup FE_2$ from RE. One of the images of the point T in the net lies in the front boundary of RE'. Its x-coordinate divides the plane into two half-planes; the left contains the whole FE'_1 and the right contains the whole FE'_2. It is based on Lemma 3.1 and the process of unfolding RE from left to right. We can analogously split BE' and LE', see Figure 3.9. More of these slight modifications cannot be done; E'_0 and E'_n have to be connected and without loss of generality we choose RE' to be the connector. More nets can be obtained by rotating the box tower before unfolding.

3.6 Unfolding box towers street

Let $T_0, \ldots, T_m$ be box towers, and let $D_0, \ldots, D_m$ be their respective bottom boundaries. We will unfold an orthogonal polyhedron P obtained by gluing $T_1, \ldots, T_m$ on top of T_0 so that the projections of $D_1, \ldots, D_m$ onto the x-axis are pairwise disjoint and if $i < j$ then the projection of D_i to the x-axis is to the right from the projection of D_j.

We leave it open whether box towers streets with more branching are unfoldable.

3.6.1 Notation

Without loss of generality, assume that P is the union of $T_0, T_1, \ldots, T_m$.

The notation is analogous to a single box tower. We only add the first index indicating the index of the box tower.

Let n_i be the number of slabs of the box tower T_i, and let $S_{i,0}, \ldots, S_{i,n_i-1}$ be the slabs of T_i. Let $z_{i,0} < z_{i,1} < \cdots < z_{i,n_i}$ be the z-coordinates of the vertices of T_i; note that $z_{0,n_0} = z_{1,0} = \cdots = z_{m,0}$. Then

$$S_{i,j} = [x_{i,j,1}, x_{i,j,2}] \times [y_{i,j,1}, y_{i,j,2}] \times [z_{i,j}, z_{i,j+1}]\,.$$

Let $R_{i,j}, L_{i,j}, F_{i,j}, B_{i,j}$ be the right, left, front, and back face of $S_{i,j}$. Let $E_{i,j}$ be the union of the horizontal faces of T_i with $z = z_{i,j}$. For every i, j satisfying $0 < j < n_i$ we subdivide $E_{i,j}$ into $E_{i,j,R}, E_{i,j,L}, E_{i,j,F}, E_{i,j,B}$ analogously as in the algorithm for unfolding a single box tower.

We denote by E_g the union of the horizontal faces of P with $z = z_{0,n_0}$ and we divide it as follows; see Figure 3.10. Let the connector C_0 be the following subset of E_g:

$$\begin{aligned} C_0 = \{(x, y, z_{0,n_0}) \in E_g;\ & \\ x \in [\min(x_{0,n_0-1,2}, x_{1,0,2}), \max(x_{0,n_0-1,2}, x_{1,0,2})] & \\ \wedge\ y \in [[y_{0,n_0-1,1}, y_{0,n_0-1,2}] \cup [y_{1,0,1}, y_{1,0,2}]])\}. & \end{aligned}$$

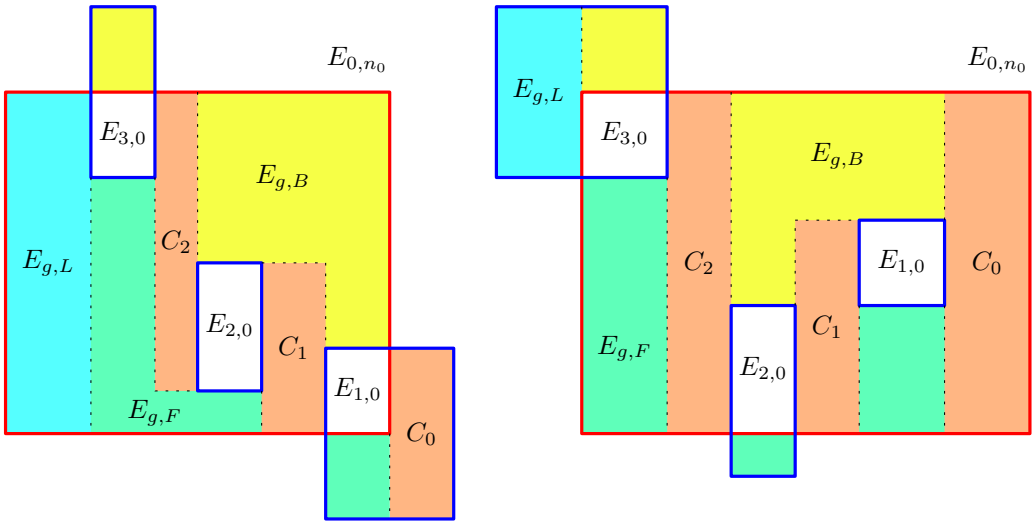

Figure 3.10: Two examples of the subdivision of E_g. The red rectangle is E_{0,n_0}, the blue rectangles are $E_{i,0}$. The union of the green faces is $E_{g,F}$, the union of the yellow faces is $E_{g,B}$, the union of the light blue faces is $E_{g,L}$ and the components of the orange faces are C_i.

(that is, the x-coordinate lies between the right edge of S_{0,n_0-1} and the right edge of $S_{1,0}$, and the y-coordinate lies in the union of the projections of the slabs S_{0,n_0-1} and $S_{1,0}$ to the y-axis). Next, let the connector C_i for $i \in \{1, \dots, m-1\}$ be the following subset of E_g:

$$\begin{aligned} C_i = \{(x, y, z_{0,n_0}) \in E_g;& \\ x \in [x_{i+1,0,2}, x_{i,0,1}]& \\ \wedge\ y \in [[y_{i,0,1}, y_{i,0,2}] \cup [y_{i+1,0,1}, y_{i+1,0,2}]]\}.& \end{aligned}$$

(that is, the x-coordinate lies between the right edge of $S_{i+1,0}$ and the left edge of $S_{i,0}$, and the y-coordinate lies in the union of the projection of the slabs $S_{i,0}, S_{i+1,0}$ to the y-axis). Let $E_{g,L}$ be the following subset of E_g:

$$\begin{aligned} E_{g,L} = \{(x, y, z_{0,n_0}) \in E_g;& \\ x \in [\min(x_{0,n_0-1,1}, x_{m,0,1}), \max(x_{0,n_0-1,1}, x_{m,0,1})]\}.& \end{aligned}$$

(that is, the x-coordinate lies between the left edge of S_{0,n_0-1} and the left edge of $S_{m,0}$).

The union of $E_{g,L}$, C_i and $E_{i,0} \cap E_g$ for $i \in \{1, \dots, m\}$ forms a contiguous chain between the leftmost and the rightmost edge of E_g. We denote by $E_{g,F}$ the union of the faces in front of the chain, and we denote by $E_{g,B}$ the union of the faces behind the chain.

Finally, for $i \in \{1, \dots, m\}$ let

$$LE_i = \bigcup\{L_{i,j}; 0 \le j < n_i\} \cup \bigcup\{E_{i,j,L}; 0 < j < n_i\}.$$

We analogously define RE_i, FE_i, BE_i, and RE_0. Let

$$LE_0 = E_{g,L} \cup \bigcup\{L_{0,j}; 0 \le j < n_0\} \cup \bigcup\{E_{0,j,L}; 0 < j < n_0\}.$$

We analogously define FE_0 and BE_0.

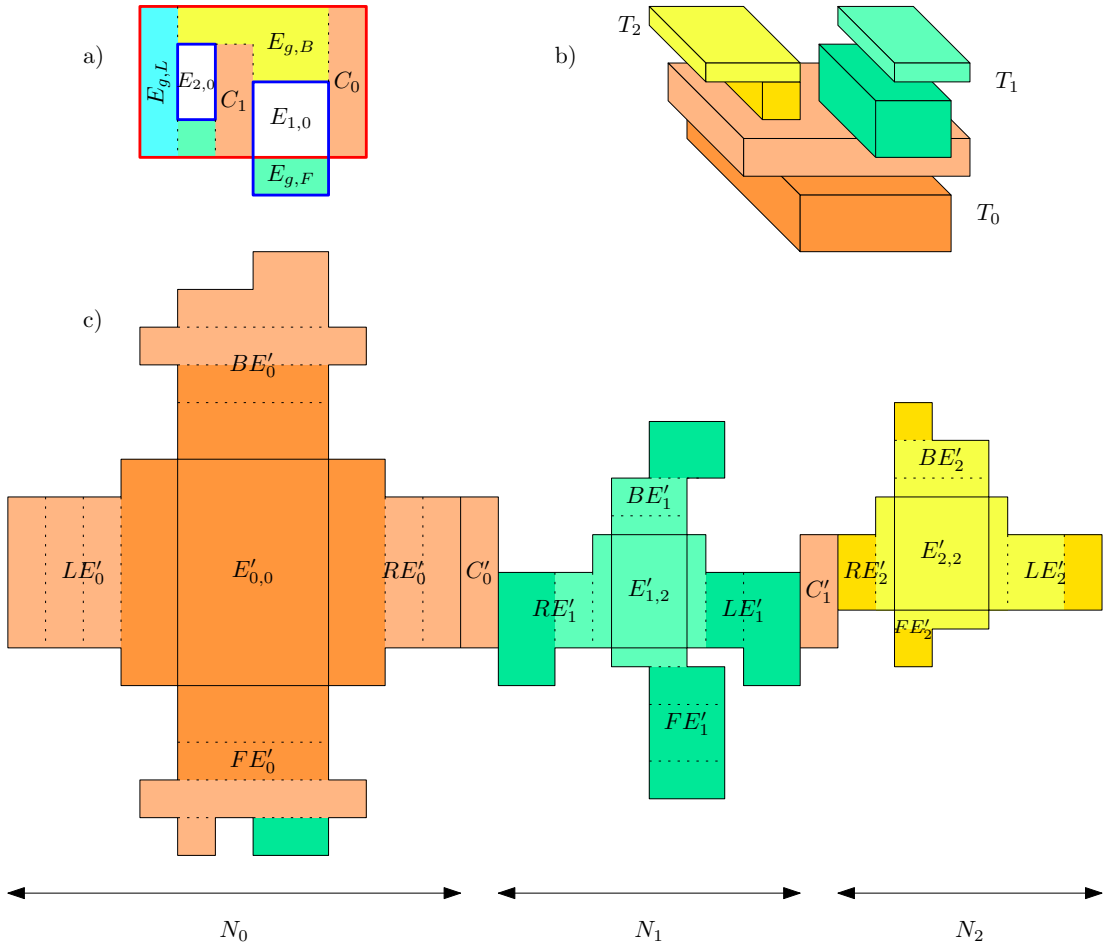

Figure 3.11: a) The subdivision of E_g. b) A box towers street. c) The resulting unfolding. The projections of N_0, N_1, and N_2 to the x-axis are pairwise disjoint.

3.6.2 Algorithm

We create a partial net N_i from each box tower T_i, $i \in \{1, \ldots, m\}$, as follows (it is a special slight net modification mentioned in Section 3.5). The partial nets show the inner surface of the box towers. We place E_{i,n_i} (the union of the topmost horizontal faces of T_i) to the net. We proceed by placing LE_i to the right, RE_i to the left, FE_i to the front, and BE_i behind E'_{i,n_i}. We analogously create a net T'_0; we place $E_{0,0}$ (the union of the bottommost horizontal faces of T_0) to the net. We place LE_0 to the left, RE_0 to the right, FE_0 to the front, and BE_0 behind $E'_{0,0}$.

Note that by Lemma 3.3, the leftmost point of N_0 lies in the left boundary of LE'_0, and the rightmost point of N_0 lies in the right boundary of RE'_0 (not in FE'_0 nor BE'_0). Also for $i > 0$ the leftmost point of N_i lies in the left boundary of RE'_i and the rightmost point of N_i lies in the right boundary of LE'_i.

We place the partial net N_0 arbitrarily into the final net. We proceed by placing to the right: $C_0, N_1, C_1, N_2, C_2, \ldots, C_{m-1}, N_m$. See Figure 3.11 for an example.

All the connecting C'_i are rectangles and connect to the leftmost and the rightmost edges of N_i. Then, the final net is simple.

Conclusion

We have presented an algorithm for grid-edge unzipping box towers, which shows promise for unfolding orthostacks with more general slabs. A modification of the algorithm successfully unfolds some orthostacks with 'L'-shaped slabs, see Figure 3.12. However, certain configurations of slabs require further investigation. Our method relies on ensuring that FE, RE, LE, and BE are contiguous, with simple cuts between BE and RE, and between RE and FE.

Additionally, we introduced an algorithm for grid-edge unfolding box towers street. We can unfold most of the faces of the polyhedron from Figure 3.13 with the algorithm for unfolding box towers street. However, the unfolding of BE_2 and FE_2 remains unresolved.

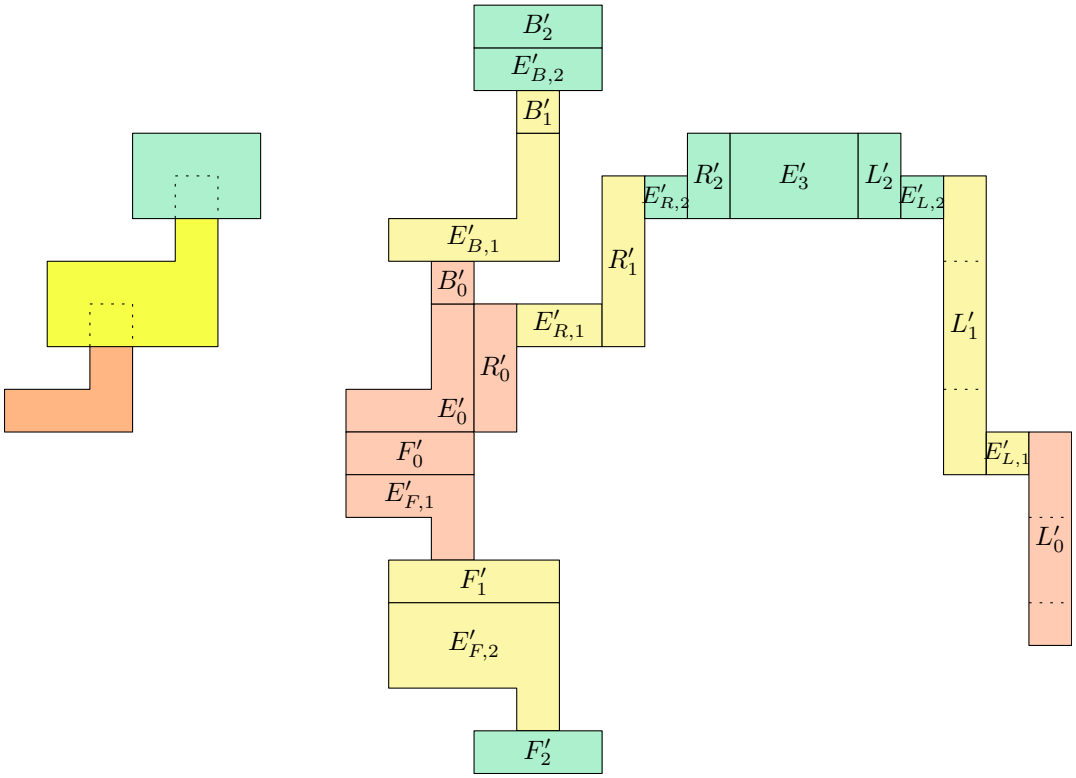

Figure 3.12: Modification of the algorithm for unzipping box towers, used for unfolding 'L'-shaped slabs.

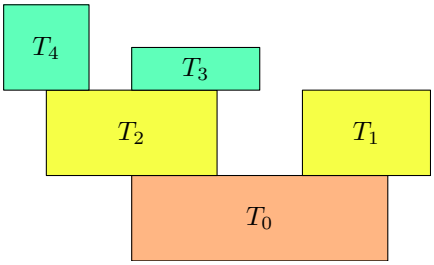

Figure 3.13: A multilayered box towers street.

Bibliography

[1] E. Andres, G. Largeteau-Skapin, L. Richaume and R. Zrour, Unfolding H-convex Manhattan towers, *J. Comb. Optim.* **44**(4) (2022), 3023–3037, URL https://doi.org/10.1007/s10878-021-00829-8.

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

[3] E. W. Chambers, K. A. Sykes and C. M. Traub, Unfolding rectangle-faced orthostacks, *Proc. 24th Canad. Conf. Comput. Geom.* (2012) 23–27.

[4] Y.-J. Chang and H.-C. Yen, Improved algorithms for grid-unfolding orthogonal polyhedra, *Internat. J. Comput. Geom. Appl.* **27**(1-2) (2017), 33–56.

[5] M. Damian, E. Demaine, R. Flatland and J. O'Rourke, Unfolding genus-2 orthogonal polyhedra with linear refinement, *Graphs Combin.* **33**(5) (2017), 1357–1379.

[6] M. Damian, E. D. Demaine and R. Flatland, Unfolding orthogonal polyhedra with quadratic refinement: the delta-unfolding algorithm, *Graphs Combin.* **30**(1) (2014), 125–140.

[7] M. Damian, R. Flatland and J. O'Rourke, Epsilon-unfolding orthogonal polyhedra, *Graphs Combin.* **23** (2007), 179–194.

[8] M. Damian and H. Meijer, Grid edge-unfolding orthostacks with orthogonally convex slabs, *14th Annual Fall Workshop on Computational Geometry* (2004) 20–21.

[9] M. Damian and H. Meijer, Edge-unfolding polycubes with orthogonally convex layers (2024), URL https://arxiv.org/abs/2407.01326.

[10] E. D. Demaine, M. L. Demaine, D. Eppstein and J. O'Rourke, Some polycubes have no edge zipper unfolding, *Proc. 32nd Canad. Conf. Comput. Geom.* (2020) 101–105.

[11] E. D. Demaine, M. L. Demaine, A. Lubiw, A. Shallit and J. L. Shallit, Zipper unfoldings of polyhedral complexes, *Proc. 22nd Canad. Conf. Comput. Geom.* (2010) 219–222.

[12] E. D. Demaine and J. O'Rourke, *Geometric folding algorithms: linkages, origami, polyhedra*, Cambridge University Press (2007).

[13] M. Ghomi, Dürer's unfolding problem for convex polyhedra, *Notices Amer. Math. Soc.* **65**(1) (2018), 25–27.

[14] J. O'Rourke, Unfolding orthogonal polyhedra, *Surveys on discrete and computational geometry: Twenty years later*, vol. 453 of *Contemp. Math.*, Amer. Math. Soc., Providence, RI, ISBN 978-0-8218-4239-3 (2008) 307–317.

[15] J. O'Rourke, Unfolding polyhedra, *Proc. 31st Canad. Conf. Comput. Geom.* (2019) 85–86.

[16] J. O'Rourke, Unfolding polyhedra (2008), URL https://www.science.smith.edu/~jorourke/Papers/PolyUnf0.pdf.

[17] G. C. Shephard, Convex polytopes with convex nets, *Mathematical Proceedings of the Cambridge Philosophical Society* **78**(3) (1975), 389–403.