



HAL
open science

Unfolding H-convex Manhattan Towers

Lydie Richaume, Eric Andres, Gaëlle Largeteau-Skapin, Rita Zrour

► **To cite this version:**

Lydie Richaume, Eric Andres, Gaëlle Largeteau-Skapin, Rita Zrour. Unfolding H-convex Manhattan Towers. 2018. hal-01963257

HAL Id: hal-01963257

<https://hal.science/hal-01963257>

Preprint submitted on 21 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Unfolding H-convex Manhattan Towers

Lydie Richaume, Eric Andres, Gaëlle Largeteau-Skapin, Rita Zrour

*Laboratory XLIM, axe ASALI, UMR CNRS 7252, University of Poitiers, bat. H1,
Teleport 2, Bld. Marie et Pierre Curie BP 30179, 86962 Futuroscope-Chasseneuil Cedex,
France*

Abstract

We propose a new grid unfolding without refinement of a new sub-class of polycubes that we call Manhattan towers with an H-convex base. Such polycubes can be seen both as a Manhattan tower and as an orthostack. At the end of the paper we show that an extension of this algorithm to Up-and-Down Orthoterrains is also possible.

Keywords: Grid Unfolding, polycubes, orthogonal polyhedra, Manhattan tower, orthostacks, orthoterrains

1. Introduction

”An unfolding of a polyhedron is produced by cutting its surface in such a way that it can be flattened to a single, connected piece without overlap” [4]. An *edge unfolding* is obtained by cutting along the edges of the polyhedron, keeping the faces whole. It is still an open question to know whether every convex polyhedron has an edge unfolding, however there are several examples of non-convex polyhedra that cannot be edge unfolded [1, 2]. *General unfolding* is less restrictive since cuts are allowed anywhere on the surface. We know that every convex polyhedron has a general unfolding [7] but there are no certitudes on non-convex polyhedra.

Since the problem of unfolding general non-convex polyhedra is complicated, there has been a focus on *orthogonal polyhedra* in general and *polycubes* in particular. An orthogonal polyhedron has all of its faces parallel to Cartesian coordinate planes. Since there are simple examples of orthogonal polyhedra that cannot be edge unfolded, *grid unfolding of orthogonal polyhedra* has been introduced. Grid unfolding is a special case of general unfolding where the surface can be cut along additional edges obtained by ”intersecting

the polyhedron with planes parallel to Cartesian coordinate planes through every vertex” [9].

There are algorithms for grid unfolding of a number of orthogonal polyhedra sub-classes, such as orthotubes [2], well-separated orthotrees [5], orthoterrains [8], etc. For other sub-classes, the known methods require the grid to be refined further by adding cutting planes in a constant amount of times: 4×5 grid-unfolding for Manhattan towers [6] and 1×2 grid-unfolding for orthostacks [2]. In more general cases, linear refinement are sometimes considered [4].

In this paper we are interested in a particular class of orthogonal polyhedra, the *polycubes*, which are composed of a set of face-connected cubes. A new grid unfolding without refinement (edge unfolding) of a new sub-class of polycubes that can be seen both as a Manhattan tower and as an orthostack is proposed: Manhattan towers with an H-convex base. A Manhattan tower can be defined by a matrix with integer value where each value corresponds to the height of a stack of cubes. Previous papers have focused on Manhattan towers with a rectangular base: each value of the matrix is a strictly positive integer. In our case, we allow zeros at the end or start of each row in the matrix, thus the name of H-convex Manhattan towers. Of course the final polycube has to be formed of one connected component. This algorithm extends the types of Manhattan towers that can be grid unfolded without refinement [9]). Our technique is somewhat similar to the one of [3].

The paper is organized as follows: in Section 2 we present some notations and definitions that will be needed all along the paper. Section 3 details our unfolding algorithm with its three distinct cases and presents some results and examples. Section 4 explains how the algorithm can be extended to Up-and-Down Orthoterrains.

2. Notations and Definitions

2.1. Notations and Definitions

In this paper we are considering cubes of side one with a center located on a 3D grid point (integer coordinates), formed of six square faces with a normal parallel to the axes, pointing to the exterior of the cube. Two cubes are said to be connected if they share a face. A polycube is a finite set of cubes that forms one connected component. The faces of a polycube are all the faces of its cubes that are not shared by another cube of the polycube. Two Faces are connected iff they share an edge and:

1. they belong to two face connected cubes and have the same normal (same direction).
2. they belong to the same cube.
3. they belong to two different, not face connected cubes, such that there exists a third cube sharing the same edge and connected to both cubes.

For a polycube C , let us consider the complementary polycube C^{-1} formed by all the cubes that do not belong to C . The faces of C may form several connected components. We are henceforth only considering the connected component whose faces belong to the infinite connected component of C^{-1} .

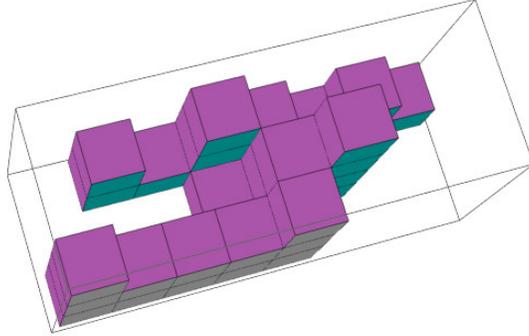
A Manhattan tower is a set of cubes defined by a rectangular 2D matrix (called Manhattan matrix) containing only positive or null integer values. A value n in $M(i, j)$ represents a stack of n cubes of coordinates between $(i, j, 0)$ to $(i, j, n - 1)$ and such that all the cubes of the Manhattan tower form a polycube.

An x -row (respectively y -column) in a Manhattan matrix $M(i, j)$ corresponds to all the values in the matrix for a fixed value i (respectively j). An H -convex (Horizontal convex) Manhattan tower is such that each x -row of the corresponding Manhattan matrix is formed of a unique contiguous sequence of strictly positive integer values (there can be zeroes at the beginning or end of the row but not in between non zero values). There is no such constraint on the y -columns. In Figure 1, an example of an H -convex Manhattan tower is presented with its matrix and unfolding. For the sake of clarity, the corresponding Manhattan matrix is written out in such a way that the lower left value of the matrix corresponds to the lowest abscissa-ordinate value.

x, y, z-faces. We call an x -face (resp. y or z -face) any face perpendicular to the x axis (resp. y or z).

Row R_i . A row R_i is the collection of cubes with $y = i$. The *base* of a row are the cubes of this row with $z = 1$. Do not mistake a row R_i with a row of the Manhattan matrix. The row R_i corresponds to the cubes represented by the i -th row of the Manhattan matrix.

Band B_i . A band B_i is the collection of x and z -faces of a row R_i .



(a) H-convex Manhattan tower with its corresponding matrix.

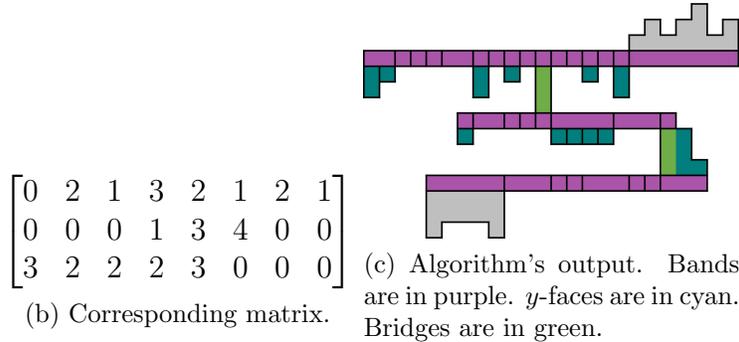


Figure 1: An example of H-convex Manhattan towers and its unfolding.

3. Algorithm

The result of our algorithm is a net structured as the example in Figure 1c: each band B_i is kept as a whole and placed horizontally, while the y -faces are placed in between with some of them forming a *bridge* linking two consecutive bands.

The algorithm iteratively places each band B_i and the corresponding y -faces. To unfold B_i , we need to consider the rows R_i and R_{i+1} . Comparing their bases, we can identify two different configurations (with $(i_1, i_2) = (i, i + 1)$ or $(i_1, i_2) = (i + 1, i)$, the order does not matter):

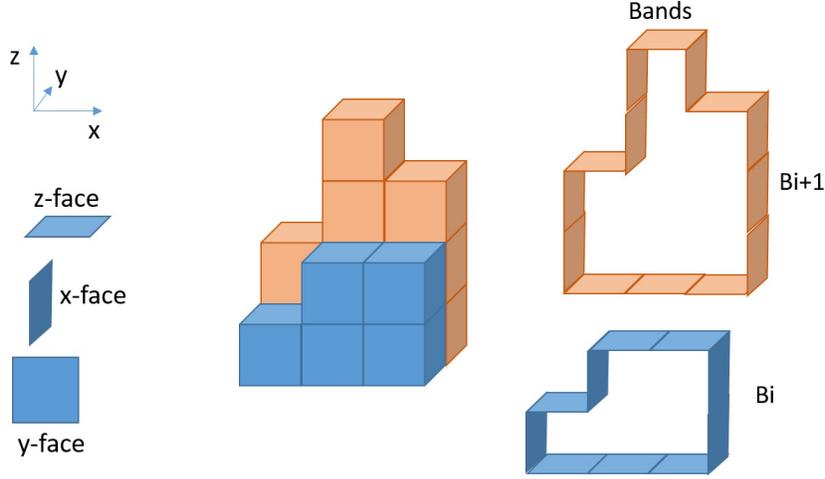


Figure 2: x, y, z -faces, rows and bands.

1. The base of row i_1 includes the base of the row i_2 ($ymin_{i_1} \leq ymin_{i_2}$ and $ymax_{i_1} \geq ymax_{i_2}$). See Figure 3a. This case is developed in section 3.1;
2. The base of row i_1 sticks out on the left and the base of row i_2 sticks out on the right ($ymin_{i_1} \leq ymin_{i_2}$ and $ymax_{i_1} \leq ymax_{i_2}$). See Figure 4a. This case is developed in section 3.2.

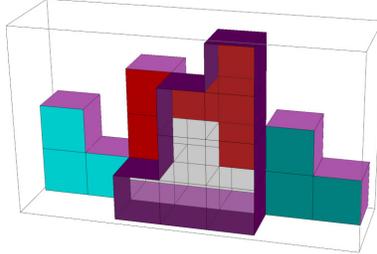
Using an example for each configuration, we will show how our algorithm ensures a non-overlapping unfolding.

3.1. Case 1: One base includes the other

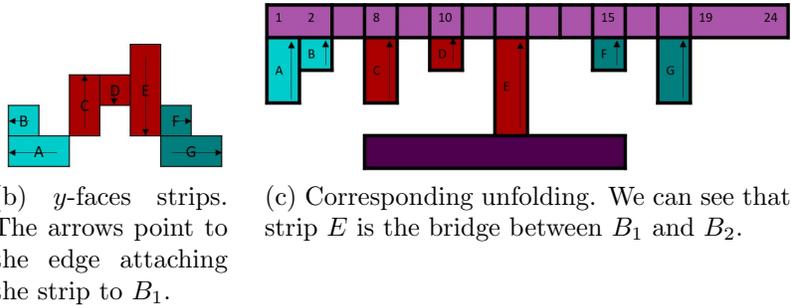
This case corresponds to the base of row i_1 that includes the base of the row i_2 ($ymin_{i_1} \leq ymin_{i_2}$ and $ymax_{i_1} \geq ymax_{i_2}$). This case is illustrated by rows 1 and 2 of the polycube in Figure 1a (two first rows of the matrix presented in Figure 1b). The faces that we have to place are the bands and the y -faces in between the two bands, as highlighted in Figure 3a.

3.1.1. Unfolding the first band

First, we unfold the band corresponding to the largest base. Let us call the band with largest base B_1 and the other band B_2 (in our example respectively the bands for $y = 1$ and $y = 2$). Note that the band with the largest base is not necessarily the band with the largest number of faces. We may have a



(a) B_1 is in light purple, B_2 in dark purple. The central y -faces are in red and the left and right y -faces are respectively in dark cyan and light cyan.



(b) y -faces strips. The arrows point to the edge attaching the strip to B_1 .

(c) Corresponding unfolding. We can see that strip E is the bridge between B_1 and B_2 .

Figure 3: Case 1: One base includes the other.

larger base with a small height while a band with a narrower base and high towers may have more faces.

Once B_1 is unfolded, we will form vertical and horizontal strips of width 1 with the y -faces and attach each strip to the unfolded band B_1 . We will show that, since we have chosen the band corresponding to the largest base, each strip has at least one common edge with B_1 . One of these strips will form the *bridge* where the band B_2 is attached, so we also have to ensure that at least one of the longest strips is attached by one edge to B_1 and shares its opposite edge with B_2 .

3.1.2. Three groups of y -faces

Let us look at the method in bit more details. Once B_1 has been unfolded, the next step is to place the y -faces. To do that we split the y -faces into

three groups. The base B_1 is defined on an interval $(ymin_1, ymax_1)$ while the base B_2 is defined on an interval $(ymin_2, ymax_2)$, with $ymin_1 \leq ymin_2$ and $ymax_1 \geq ymax_2$. We split the y -faces into three groups where y lies in one of the intervals $(ymin_1, ymin_2)$, called the left y -faces (in light cyan in Figure 3), $(ymin_2, ymax_2)$ called the central y -faces (in red in Figure 3) and $(ymax_2, ymax_1)$ called the right y -faces (in dark cyan in Figure 3).

3.1.3. Unfolding the central y -faces

With the central y -faces (in red in Figure 3), we form vertical strips, one for each $x = i$ (C , D and E on the Figure). Each strip shares at least one edge with a face of the band B_1 and will be attached to the strip on that edge. We can notice that the opposite edge of each strip is shared with B_2 , so any of them could be used as bridge between B_1 and B_2 . Note as well that the edge of such a strip in contact with B_1 and the edge in contact with B_2 are unique and distinct (an edge is only shared by two faces and there can not be another y -face that shares that edge). This means that none of the central strips are overlapping.

3.1.4. Unfolding the left and right y -faces

With the left y -faces, (in light cyan in Figure 3), and the right y -faces (in dark cyan in the Figure 3), we form horizontal strips, at least one (and possibly multiple) for each $z = i$. Each strip shares at least one edge, the exterior one, with B_1 , and will be attached to the strip B_1 on that edge. We can note that for $z = 1$, there can only be a single strip on each side (strips A and G in Figure 3), which will share its exterior edge with B_1 and its interior edge with B_2 . Let us note as well that among the left or right strips, the strip for $z = 1$ is necessarily the longest. This means they may be bridge candidates.

Again, each left (resp. right) strip has only one edge at the left (right) of the strip that is in common with B_1 . This edge (and the face it is attached to in B_1) is necessarily different from the other edges already considered for central y -faces and other left or right y -faces. Lastly, it is easy to see that, when attaching the left (resp. right) most face of a left (resp. right) horizontal strip to B_1 , the horizontal strip will become vertical on the net when B_1 is unfolded. The uniqueness of the face of B_1 it is attached to and the horizontally attachment means that it does not overlap with any other strip (be it central, left or right).

3.1.5. Choosing the bridge and attaching the second band

We now have placed all the y -faces with no overlaps between any of them. The last step is to find where to attach the band B_2 so that it does not overlap with the previously placed y -faces. In [3] the choice is to attach to the leftmost or rightmost strip and to have band B_2 cut in such a way that it overhangs away from the other strips, which restrains the way we can unfold B_2 . To avoid any overlap, we can simply attach the band B_2 to (one of) the longest y -face strip (see Figure 3c).

3.1.6. About cutting the bands

We have not discussed about the way a band is cut yet. For each band, when unfolding it into an horizontal sequence of faces, we have to decide what will be the first (and last) face of the sequence (where to cut the band). There is no constraint for this case, however, as we will see, for case 2 (section 3.2) we need to cut the band on a specific edge. We will explain how to deal with this problem in general once we have explained what the constraints are for case 2.

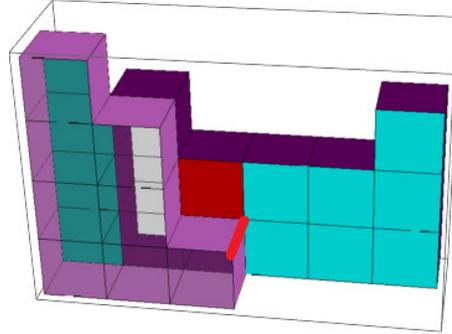
3.2. Case 2: Each base sticks out on one side

This case corresponds to the base of row R_{i_1} that sticks out on the left and the base of row R_{i_2} sticks out on the right ($ymin_{i_1} \leq ymin_{i_2}$ and $ymax_{i_1} \leq ymax_{i_2}$). This second case is illustrated by lines 2 and 3 of the polycube in Figure 1a. The faces that we have to place are the bands and the y -faces in between the two bands, as highlighted in Figure 4a.

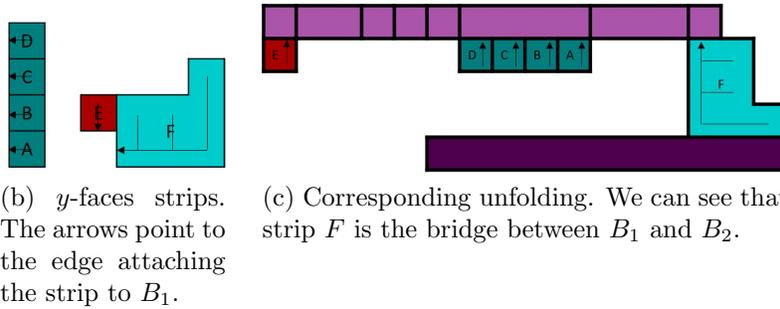
The band we unfold first for case 2 does not matter as long as it is done in a consistent way for the whole algorithm (see section 3.3 for a discussion on this matter). Once again, let us call B_1 (corresponding to row R_1) the band we choose to unfold first, and B_2 the other band (with R_2 the corresponding row). Let us consider, for the sake of simplicity, that R_1 sticks out on the left and R_2 sticks out on the right (switch left and right for what follows if R_1 sticks out on the right and R_2 sticks out on the left).

3.2.1. Cutting the band and unfolding the right y -faces

Unlike case 1, we need to control the cut location of band B_1 to avoid overlaps. One suitable cut location is found as follows: let us call \mathcal{F} the right face of the rightmost voxel of R_{i_1} with $z = 1$. The band B_1 is cut along the top edge of face \mathcal{F} (Red edge in Figure 4a).



(a) R_1 (with B_1 in light purple) sticks out on the left while R_2 (with B_2 in dark purple) sticks out on the right. The central y -faces are in red and the left and right y -faces are respectively in dark cyan and light cyan. The cut location for B_1 is highlighted in red.



(b) y -faces strips. The arrows point to the edge attaching the strip to B_1 .

(c) Corresponding unfolding. We can see that strip F is the bridge between B_1 and B_2 .

Figure 4: Case 2: One base sticks out on each side.

The set of right y -faces is rotated by 90° to the left and attached to B_1 on \mathcal{F} . Since \mathcal{F} is the leftmost face of unfolded B_1 (because of the cutting), all the right y -faces correspond either to a vertical strip on the final net attached to face \mathcal{F} (corresponding to the horizontal strip of y -faces with $z = 1$) or stick out on the left of the unfolded band B_1 (set of faces F in Figure 4b and 4c). If this set of y -faces were not attached to the leftmost face of B_1 , sticking out on the outside, we could have a strip from the left or central part that would overlap.

We can note that the horizontal strip of faces for $z = 1$ is always connected with B_1 at one end and B_2 at the other end. Let us note as well that the rightmost face of this strip has the biggest x value and therefore, in the net,

there can be no other right y -face that is located higher. This means that the $z = 1$ y -face strip is a potential candidate for the bridge between B_1 and B_2 .

3.2.2. *Unfolding the remaining y -faces*

When we look only at the central and left y -faces, the configuration is actually the same as in case 1 where this part of the base of R_{i_2} is included in the base of R_{i_1} . The remaining central and left y -faces can therefore be handled exactly in the same way as in case 1.

3.3. *Row Unfold Order*

Let us consider the rows R_1 , R_2 and R_3 and the corresponding bands B_1 , B_2 and B_3 . For each couple of rows, (R_1, R_2) and (R_2, R_3) , we may have two constraints. For a couple in case 1, we must unfold the band corresponding to the largest base first and for a couple in case 2 we must cut one of the bands at a precise location.

3.3.1. *Case 1 and Case 1*

When both couples (R_1, R_2) and (R_2, R_3) are in case 1 (see rows 1, 2 and 3 on Figure 5), we can simply deal with the first couple of rows and unfold it and then deal with (R_2, R_3) . Now, let us imagine that band B_2 had the longest base of (B_1, B_2) and B_3 the longest base of (B_2, B_3) , then we would have to unfold B_3 before B_2 which seems to be in contradiction with the fact that we unfold (R_1, R_2) first before considering (R_2, R_3) . If B_3 has a longer base than B_2 and if B_2 is already unfolded, we have actually two simple possibilities to deal with this apparent contradiction: the first one consists simply to predetermine the bridge. It is rather simple since it corresponds to (one of) the longest y -face strip(s) and we know exactly how these strips are made. Once the longest y -face strip between B_2 and B_3 is determined and unfolded, the band B_3 and all the y -faces attached to B_3 (and facing towards B_2) can be put down.

The second solution consists simply in unfolding B_3 somewhere away from the net and once the band and all the y -faces are put down, to translate all these faces so as to make the bridge with B_2 .

3.3.2. *Case 1 and Case 2*

When (R_1, R_2) is in case 1 and (R_2, R_3) is in case 2 (see rows 2, 3 and 4 on Figure 5), we can simply deal with the first couple of rows and unfold it

and then deal with (R_2, R_3) . The idea is to unfold the band B_2 with a cut at the specific location corresponding to the configuration related to case 2 of (R_2, R_3) (since we have a choice between B_2 and B_3). The fact that B_2 may have the longest base and is unfolded first does not matter since we can always unfold B_2 before B_1 with a specific cut location.

3.3.3. Case 2 and Case 1

When (R_1, R_2) is in case 2 and (R_2, R_3) is in case 1 (see rows 4, 5 and 6 on Figure 5), we can simply deal with the first couple of rows and unfold it and then deal with (R_2, R_3) . Here we can simply choose to cut B_1 at a specific location which has no consequence on whether B_2 or B_3 has the longest base. In case B_3 has a longer base than B_2 , we can apply the same strategies than for case 1 - case 1 in section 3.3.1.

3.3.4. Case 2 and Case 2

When both couples (R_1, R_2) and (R_2, R_3) are in case 2 (see rows 3, 4 and 5 on Figure 5), we can simply deal with the first couple of rows and unfold it and then deal with (R_2, R_3) . The idea is to unfold B_1 first with a specific cut location to deal with case 2 of the couple (R_1, R_2) , and then later on unfold B_2 with a specific cut location to deal with the case 2 of the couple (R_2, R_3) .

As we can see, each time it is possible to unfold a couple of rows (R_i, R_{i+1}) before unfolding the next couple (R_{i+1}, R_{i+2}) as long as we identify the cases of consecutive couples of rows.

3.4. Extremal y -faces

Once all the y -faces between two bands have been placed, there remains the y -faces that are not between two bands, on either side of the H-convex Manhattan towers (in the direction of the rows). These faces can be attached directly as strips or as a whole set of faces to the first and last band respectively (See the grey faces in Figure 1c).

4. Extension: Up-and-Down Orthoterrains

This algorithm can easily be extended to a subclass that we call *Up-and-Down Orthoterrains* (UDO). An UDO is defined by rectangular matrix where all the values are formed of a pair of integers $(m_{i,j}, M_{i,j}) \in \mathbb{Z}^2$, such that $m_{i,j} < M_{i,j}$. Each pair represents a set of cubes of coordinates $(i, j, z) \in \mathbb{Z}^3$ with $m_{i,j} \leq z < M_{i,j}$. Such a set of cubes is called a tower of base coordinates

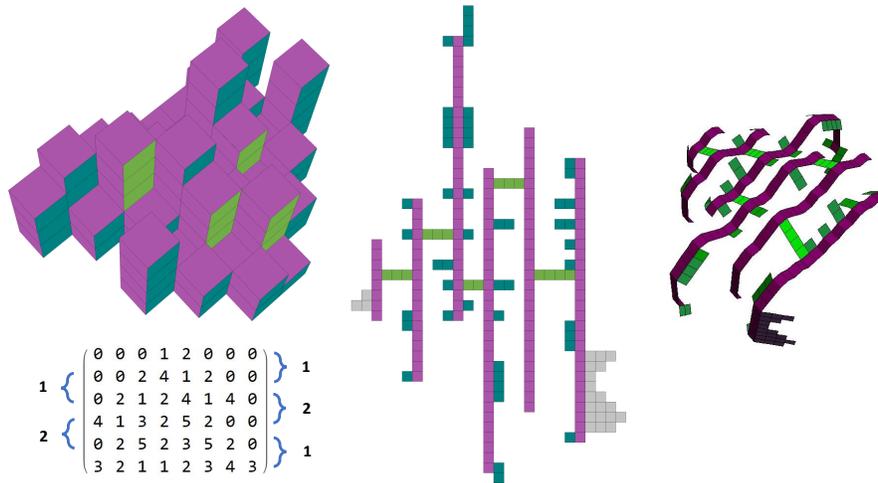
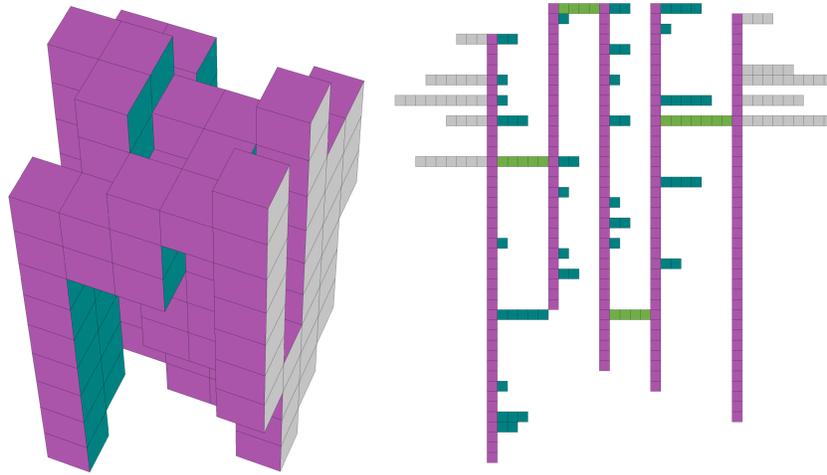


Figure 5: The Manhattan Tower with the corresponding matrix, the net and the net as it is refolded.

(i, j) . In order to have a valid polycube for our algorithm, each tower has to be face-connected to every of its neighboring towers. Every towers must be composed of at least one cube, i.e. the projection along z -axis is rectangular. See Figure 6 for an example.

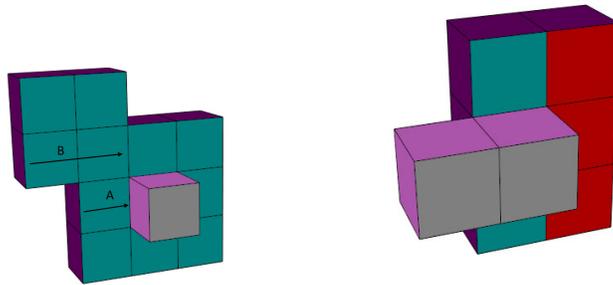
To unfold this kind of polycube, we can apply a simplified version of the algorithm we previously exposed in section 3. Since the polycube is rectangular along the z -axis, every line will have the same width. The consequence is that all the y -faces will be in the central case (attached vertically to the band).

We cannot extend the definition of the Up-and-Down Orthoterrains to accept H-convex projection: the way we treat left and right y -faces in the previous algorithm would not work here. As we can see on Figure 7a, the possible bridges between both bands here may be shorter than some strips and thus lead to overlaps. In Figure 7b we can see another problem: for case 2 in the H-convex Manhattan unfolding method (section 3.2), the y -faces that stick out on one of the sides are rotated by 90 degrees so that the y -faces stick out from the unfolded band (see section 3.2.1). This will not be the case here and therefore there can be overlaps.



$$\begin{bmatrix} (1 & 3) & (-2 & 3) & (-3 & 5) & (-4 & -1) & (-1 & 5) \\ (-1 & 1) & (-1 & 0) & (-4 & 4) & (-1 & 4) & (4 & 4) \\ (-5 & -1) & (-2 & -1) & (-4 & 4) & (-1 & 3) & (2 & 4) \\ (-3 & 3) & (-4 & -1) & (-3 & 5) & (1 & 5) & (2 & 3) \\ (1 & 3) & (-3 & 1) & (-3 & 5) & (-4 & 1) & (-5 & 3) \end{bmatrix}$$

Figure 6: An example of an Up-and-Down Orthoterrain with its corresponding matrix and unfolding.



(a) A: One of the longest bridges. B: A strip longer than the bridge

(b) Part in red should rotate of 90 degrees on the unfolding, but there would be possible overlapping.

Figure 7: Limits on UDO

5. Conclusion

We have established an algorithm for unfolding Manhattan towers with an H-convex base without any refinement. Previous grid unfolding algorithms were able to treat orthoterrains (with rectangular base) [8, 9] or general Manhattan towers [9] but with a 4×5 refinement [6]. The algorithm we propose does not need any refinement and its complexity is linear in terms of number of faces. We also proposed an extension we called Up-and-Down Orthoterrain as a direct consequence of the method we proposed for H-convex Manhattan towers. A next step could be to propose more general Manhattan tower landscapes including, for instance, central parks (convex regions with values 0 inside the Manhattan matrix). The unfolding of more general polycubes remains an open question.

- [1] Bern, M. W., Demaine, E. D., Eppstein, D., Kuo, E., Mantler, A., Snoeyink, J., 2003. Ununfoldable polyhedra with convex faces. *Comput. Geom.* 24 (2), 51–62.
- [2] Biedl, T. C., Demaine, E. D., Demaine, M. L., Lubiw, A., Overmars, M. H., O’Rourke, J., Robbins, S., Whitesides, S., 1998. Unfolding some classes of orthogonal polyhedra. In: *Proceedings of the 10th Canadian Conference on Computational Geometry*, McGill University, Montréal, Québec, Canada, August 10-12, 1998.
- [3] Chambers, E. W., Sykes, K., Traub, C., 2012. Unfolding rectangle-faced orthostacks. In: *CCCG*. pp. 23–28.
- [4] Damian, M., Demaine, E. D., Flatland, R. Y., O’Rourke, J., 2017. Unfolding genus-2 orthogonal polyhedra with linear refinement. *Graphs and Combinatorics* 33 (5), 1357–1379.
- [5] Damian, M., Flatland, R., Meijer, H., O’Rourke, J., 2005. Unfolding well-separated orthotrees. In: *15th Annual Fall Workshop on Computational Geometry and Visualization*. pp. 23–25.
- [6] Damian, M., Flatland, R. Y., O’Rourke, J., 2008. Unfolding manhattan towers. *Comput. Geom.* 40 (2), 102–114.
- [7] Demaine, E. D., O’Rourke, J., 2007. *Geometric folding algorithms - linkages, origami, polyhedra*. Cambridge University Press.
- [8] O’Rourke, J., 2007. Unfolding orthogonal terrains. *CoRR* abs/0707.0610.
- [9] O’Rourke, J., 2008. Unfolding orthogonal polyhedra. *Contemporary Mathematics* 453, 307.