# Unfolding Some Classes of One-Layer Polycubes

Josef Minařík*


## Abstract

An *unfolding* of a polyhedron is a cutting along its surface such that the surface remains connected and it can be flattened to the plane without any overlap. An *edge-unfolding* is a restricted kind of unfolding, we are only allowed to cut along the edges of the faces of the polyhedron. A *polycube* is a special case of orthogonal polyhedron formed by glueing several unit cubes together face-to-face. In the case of polycubes, the edges of all cubes are available for cuts in edge-unfolding. We focus on one-layer polycubes and present several algorithms to unfold some classes of them. We show that it is possible to edge-unfold any one-layer polycube with cubic holes, thin horizontal holes and separable rectangular holes. The question of unfolding general one-layer polycubes remains open.


## 1 Introduction

An *orthogonal polyhedron* is a polyhedron whose edges are parallel to the Cartesian axes and whose faces meet at right angles. Each face of an orthogonal polyhedron is parallel to one of the Cartesian coordinate planes. A *polycube* is a special case of an orthogonal polyhedra. It is formed by glueing several unit cubes together by whole faces. Polycubes are three-dimensional generalisations of planar polyominoes. A one-layer polycube is a polycube of height 1. In other words, the centers of all unit cubes are in one plane. One-layer polycubes with non-zero genus have some *holes* in them. If a hole consists of only one missing unit cube, we call this hole *cubic*.

An *unfolding* of a polyhedron is a cutting along its surface such that the surface remains connected and it can be flattened to the plane without any overlap. We usually only care about interior overlap and there may be touching edges after unfolding to the plane. *Edge-unfolding* is a restricted kind of unfolding. In this case, we can only cut along the edges of the faces of the polyhedron. It is quite easy to show that there exist non-convex orthogonal polyhedra that cannot be edge-unfolded [7]. We are mostly interested in edge-unfolding of polycubes. In the case of polycubes, the edges of all cubes are available for cuts. This means that we can cut the faces of our polyhedron along the edges of the $1 \times 1$ grid. Different kinds of unfolding are discussed in greater detail in [6] and [8].

Unfoldings of many classes of orthogonal polyhedra have been studied. For example orthostacks, orthotubes [1] or Manhattan towers [4]. There are also known unfoldings of special cases of polycubes, such as well-separated orthotrees [3]. One-layer orthogonal polyhedra with arbitrary genus $g$ can be edge-unfolded using only $2(g-1)$ additional cuts [2]. Kiou, Poon and Wei proved that it is possible to unfold one-layer polycubes with sparse cubic holes [5], which are one-layer polycubes such that each connected component in a column contains at most one hole. We generalize this result and present an algorithm for unfolding general one-layer polycubes with cubic holes.

**Theorem 1** *It is possible to edge-unfold any one-layer polycube with cubic holes.*

In Sections 2.5 and 2.6 we further generalize this approach for other classes of one-layer polycubes.

**Definition 1** *A hole is called thin horizontal if it is a rectangle of height 1.*

**Theorem 2** *It is possible to edge-unfold any one-layer polycube with thin horizontal holes.*

**Definition 2** *We call a set of rectangles separable if it satisfies the following property. If we extend any edge of any rectangle to a line, it does not cut any other rectangle.*

**Theorem 3** *It is possible to edge-unfold any one-layer polycube with separable rectangular holes.*

## 2 Results

### 2.1 Definitions

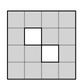
Let us consider a one-layer polycube $\mathcal{P}$ placed in the $xy$ plane such that the centers of all cubes have integer coordinates. The exact position of the polycube is not important, we only need to be able to index the cubes by coordinates. By a cube with coordinates $x, y$ we mean a cube whose center has such coordinates. Let us denote the set of holes $\mathcal{H}$. A one-layer polycube $\mathcal{P}$ has a top base $T$ and bottom base $B$. There also is

*Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University

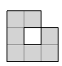
an *external boundary* $E$ and several *internal boundaries* $\mathcal{I} = \{I_h \mid h \in \mathcal{H}\}$, each corresponding to some hole $h$. The boundaries are formed by cyclic stripes of unit squares.

Since we are only interested in one-layer polycubes, we will display them as 2-dimensional objects. In all the figures, we are looking at the polycube from above, which means that we see the top base. With respect to that, we will be using terms such as "left", "right", "up" and "down" to describe directions. For example, the boundary of a hole consists of four not necessarily connected parts: left, right, upper and lower.

We require the surface of $\mathcal{P}$ to be *simple*, that is, every edge of $\mathcal{P}$ is incident to exactly two $1 \times 1$ squares on the surface of $\mathcal{P}$. The holes are not allowed to "touch" each other by corners nor to "touch" the external boundary, examples of such disallowed configurations are in Figure 1.



(a) Example of holes touching.



(b) Example of a hole touching the external boundary.

Figure 1: Examples of polycubes that are **not** allowed.

We will describe several algorithms for unfolding polycubes. Let $n$ denote the number of unit cubes forming $\mathcal{P}$. All of the presented algorithms can be implemented in $\mathcal{O}(n)$ time if we are provided with a reasonable representation of the polycube as input (for example a sorted list of all unit cubes). We mainly focus on the existence of the unfolding and the existence of such an algorithm is more important for us than the exact implementation. However, an implementation of all the presented algorithms should be mostly straightforward.

## 2.2 No holes

Let us start with a simple example to get familiar with the techniques we will be using. Without holes, we only need to unfold $B$, $T$ and $E$. The algorithm starts with the external boundary $E$. The external boundary can be unfolded to a single stripe of height 1. Let us place this stripe horizontally in the plane. We do not cut $B$ and $T$. We simply connect them to the unfolded $E$, each being placed in a different half-plane. They are connected to $E$ by the cube with the lowest $y$ coordinate (if there are two or more, we can choose one arbitrarily). The resulting shape is connected and it is easy to see that there are no overlaps. See Figure 2 for an example of an unfolding of a polycube without holes.

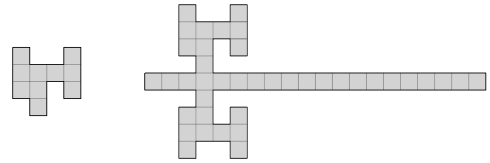
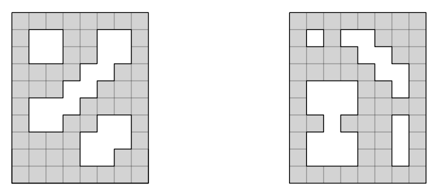


Figure 2: Unfolding of a one-layer polycube without holes.

Note that this is an edge-unfolding in the standard sense, we only used cuts along the faces of the polyhedron. We did not use any additional cuts along the edges of the unit cubes.

## 2.3 Wide holes

**Definition 3** *A hole $h$ is called narrow if $I_h$ contains two squares with distance 1 facing each other. A hole is wide if it is not narrow. In other words, we say that a hole is wide if it satisfies the following property. If there is a missing cube with a center on coordinates $[x, y]$, then there is at least one missing cube at coordinates $[x+1, y], [x-1, y]$ and at least one missing cube at coordinates $[x, y-1], [x, y+1]$.*



(a) Examples of wide holes.

(b) Examples of **non**-wide holes.

Figure 3: Examples of wide and non-wide holes.

We can unfold one-layer polycubes with wide holes using the following algorithm. We start by unfolding $B$, $T$ and $E$ in the same way as above in Section 2.2. Thanks to the wideness of the holes, there is a lot of space inside $B$ and $T$. For every hole $h$, we will unfold $I_h$ in two steps. In the first step, we unfold the upper and the lower faces of $I_h$. In the second step, we unfold the left and the right faces of $I_h$. In the first step, we unfold parts of $I_h$ into the top base $T$, inside the holes. There will be no overlap because the holes are wide. The second step is almost the same, the only difference is that we use the bottom base $B$ instead. Figure 4 shows an example of an unfolding produce by this algorithm.

We again used only cuts along the edges of $\mathcal{P}$.

## 2.4 Cubic holes

This algorithm is slightly more complicated, we will need to cut $T$ and $B$. Note that cutting $T$ or $B$ is necessary to unfold even a single cubic hole. The idea

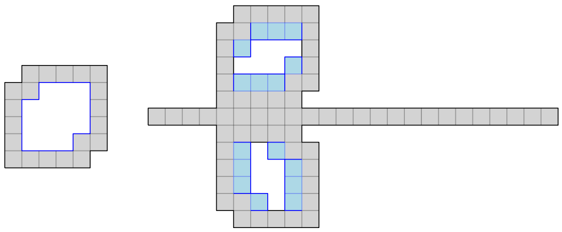


Figure 4: Unfolding of a one-layer polycube with a wide hole.

is similar to the algorithm in Section 2.3, we will unfold some parts (the upper and the lower faces) of the internal boundaries by connecting them to $T$ and some of them (the left and the right faces) by connecting them to $B$.

The first steps are still the same: we unfold the external boundary $E$. Now let us color the squares of $T$ using orange and red. The squares whose $y$-coordinate is 0 or 1 modulo 4 will be orange, the remaining ones will be red. In other words, we are coloring pairs of rows orange and red. Example of such coloring can be seen in Figure 6. Consider the connected components formed by orange or red squares, which would be formed by cutting edges separating squares of different colors. The leftmost and the rightmost square of every connected component must be incident to $E$. This is due to the holes being cubic; they are not big enough to separate components.

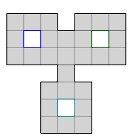


Figure 5: Example of a one-layer polycube with cubic holes.

(a) Coloring of the top base.

(b) Coloring of the bottom base.

Figure 6: Coloring of the one-layer polycube with cubic holes in Figure 5.

We will connect all the orange components to the external boundary on their left side by their leftmost square. Analogously every red component will be connected to the boundary by its rightmost square. The current stage of unfolding is illustrated in Figure 7. Quite simple casework shows that there is a distance of at least 2 between any pair of connected components

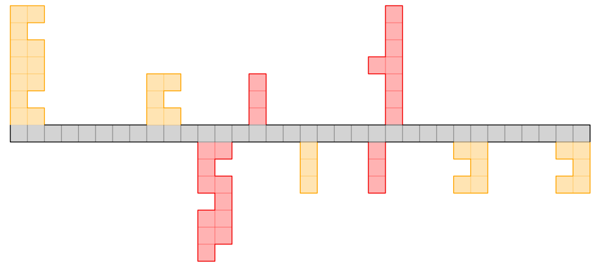


Figure 7: The first step of unfolding the polycube in Figure 5.

after placing them in the plane next to unfolded $E$. Suppose there are two stripes that have a distance of less than two. There are two cases:

1. Both of the stripes have the same color. We can suppose without loss of generality they are orange. Now consider where these stripes come from in the polycube. They either come from the same pair of rows or a different pair of rows. In the first case, the distance would have to be at least 3, in the second case, it would have to be at least 2, a contradiction. See Figure 8 for an illustration.

2. The stripes have different colors. Without loss of generality, we can assume that the left stripe is orange. Let us again consider where those stripes were before the unfolding. If they don't come from the neighboring pair of rows, the distance would obviously have to be at least 4. There are two remaining (symmetric) cases: the red rows could be either above or below the orange rows. In both of those cases, the distance is at least 2, contradiction again, see Figure 9.

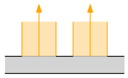
Now, we will take every left or right face of the internal boundaries and connect it to the only square of the already unfolded top base it is incident to. There are no overlaps because the connected components have a distance of at least 2 and there is enough space for two unit squares between them.

We repeat the process for the bottom base $B$. This time, we color pairs of columns instead of rows. This base and parts of holes are unfolded to the opposite half-plane so there will be no overlaps with previously placed parts.

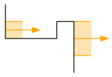
### 2.5 Thin horizontal holes

The approach in Section 2.4 can be quite easily generalized to holes of dimensions $1 \times k$, but only if all of them are oriented in the same way (either all horizontal or all vertical).

**Definition 4** *A hole is called thin horizontal if it is a rectangle of height 1.*



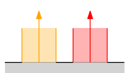
(a) Suppose the distance of two orange stripes is at most 1.



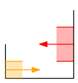
(b) In the first case, the distance after unfolding must be at least 3.

(c) In the second case, the distance after unfolding must be at least 2.

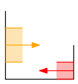
Figure 8: Two orange stripes cannot be too close to each other.



(a) Suppose the distance of an orange and red stripe is at most 1.



(b) In the first case, the distance after unfolding must be at least 2.



(c) In the second case, the distance after unfolding must be at least 2.

Figure 9: Two stripes of different colors cannot be too close to each other.

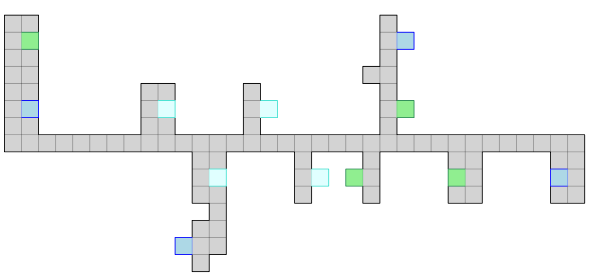
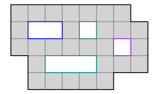
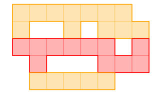


Figure 10: Unfolding of the one-layer polycube with cubic holes in Figure 5.



(a) Example of a polycube with thin horizontal holes.



(b) Coloring of the top base.

Figure 11: Coloring of a one-layer polycube with thin horizontal holes.

Let us start by unfolding $E$, $T$ and longer (horizontal - upper and lower) faces of holes in the same way as in Section 2.4. The Figures 12 and 13 show the first two steps of the algorithm.

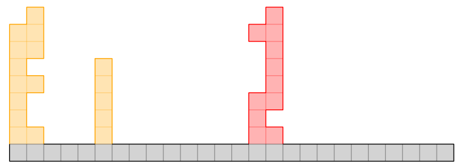


Figure 12: The first step of unfolding the polycube in Figure 11.

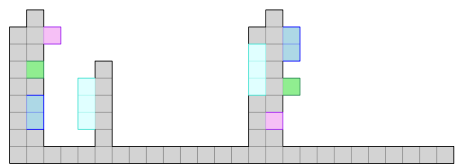


Figure 13: The second step of unfolding the polycube in Figure 11.

It remains to unfold the bottom base $B$ and the short (left and right) faces of holes. We cannot do that in the same way as before, because if we cut $B$ into stripes of width 2, they would not be necessarily incident to the external boundary. We can instead connect one face of each hole $h \in \mathcal{H}$ to one of the two already unfolded faces of $I_h$. In case of holes in the orange stripes, we unfold the right face, in case of holes in the red stripes, we unfold the left face. Let us look at the already unfolded horizontal faces of $I_h$. One of the faces is unfolded "inside" of a stripe but the other is "outside". For example, consider a hole in the lower row of a red stripe: the upper face of this hole is unfolded "inside" the red stripe while the lower face is unfolded "outside" of an orange stripe. The face unfolded outside has empty space around it and we can connect the one face of $I_h$ here (this face is only one $1 \times 1$ square). There cannot be an overlap - we are outside a stripe, so there could only be a face of some hole or external boundary. External boundary cannot be there because it has distance at least 1 from all holes (and it also lies in the opposite direction than the one in which we place the face).

The same is true for holes, they are at a distance of at least 1 from each other, so the unfolded longer faces are not next to each other. Two faces unfolded in this step cannot overlap either because they are unfolded in the same direction.

The last part is the bottom base $B$ and exactly one face of every hole. This is rather simple since all of the remaining faces are just $1 \times 1$ squares. We can unfold the rest in a similar fashion to unfolding wide holes in Section 2.3. See the Figure 14 for an example of the last steps.

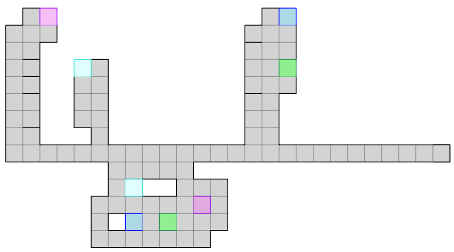


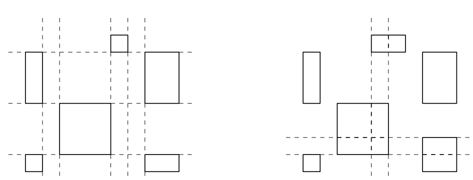
Figure 14: Unfolding of the polycube in Figure 11.

### 2.6 Separable rectangular holes

A slightly more general class of one-layer polycubes than the polycubes with cubic holes can also be unfolded using a similar algorithm.

**Definition 5** *We call a set of rectangles separable if it satisfies the following property. If we extend any edge of any rectangle to a line, it does not cut into any other rectangle (it does not contain an interior point of any other rectangle).*

One-layer polycubes whose holes are separable rectangles can be unfolded by an algorithm very similar to the one in Section 2.4. Note that cubic holes are trivially separable, thus one-layer polycubes with cubic holes can also be unfolded using this algorithm.



(a) Example of separable rectangles.

(b) Example of **non**-separable rectangles.

Figure 15: Examples of separable and non-separable rectangles.

Let us extend the edges of all rectangles that are parallel to $x$-axis to lines. This creates several horizontal stripes. Analogously, we can create vertical stripes. Instead of coloring pairs of neighboring rows or columns of $T$ and $B$ as in Section 2.4, we color pairs of neighboring horizontal stripes. You can see an example of such coloring in Figure 17. The distance between any pair of stripes is again at least 2 for the same reasons as in the algorithm for cubic holes. We omit the case analysis this time.

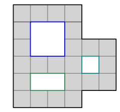


Figure 16: Example of a one-layer polycube with separable rectangular holes.

(a) Coloring of the top base.

(b) Coloring of the bottom base.

Figure 17: Coloring of the polycube in Figure 16.

The rest of the algorithm is the same as in Section 2.4. We consider connected components of both colors. The leftmost and rightmost squares of connected components are incident to $E$ and will be connected on the left or right side depending on their color. We then unfold the horizontal and vertical faces of internal boundaries separately. Since the distance between neighboring stripes is at least 2, there are no overlaps. Figures 18 and 19 show the steps of this algorithm.

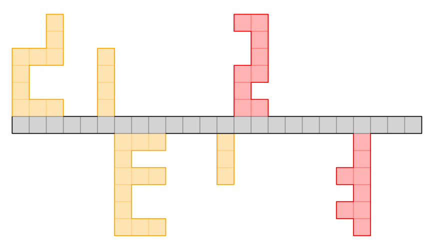


Figure 18: The first step of unfolding of the polycube in Figure 16.

## 3 Conclusion

We presented several linear-time algorithms for edge-unfolding of special cases of one-layer polycubes. The question of unfolding one-layer polycubes with arbitrary holes remains open. Interestingly, we are able to unfold one-layer polycubes with very small (cubic) holes and with very big (wide) holes. These are, in some sense, opposite types of one-layer polycubes. Generalising our

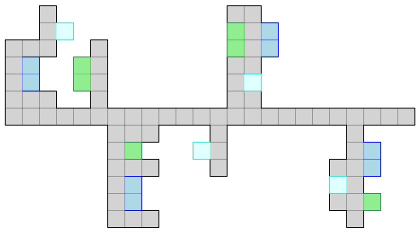


Figure 19: Unfolding of the polycube in Figure 16.

approach to unfold other classes of one-layer polycubes seems rather difficult since it relies on being able to cut the top and bottom bases into stripes such that all the connected components are incident to the external boundary.

## References


[1] 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.*, 01 1998.

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

[3] M. Damian, R. Flatland, H. Meijer, and J. O'Rourke. Unfolding well-separated orthotrees. pages 23–25, 2005.

[4] M. Damian, R. Flatland, and J. O'Rourke. Unfolding Manhattan towers. *Comput. Geom.*, 40(2):102–114, 2008.

[5] M.-H. Liou, S.-H. Poon, and Y.-J. Wei. On edge-unfolding one-layer lattice polyhedra with cubic holes. *Computing and combinatorics*, 8591:251–262, 2014.

[6] J. O'Rourke. Unfolding orthogonal polyhedra. *Surveys on discrete and computational geometry*, 453:307–317, 2008.

[7] J. O'Rourke. Unfolding polyhedra. 08 2008. `https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.150.8196`

[8] J. O'Rourke. Unfolding polyhedra. *Proc. 31st Canad. Conf. Comput. Geom.*, 2019.