Star Unfolding of Boxes

Dani Demas
Technology, Amazon.com, Seattle, 98109 WA
daniellefdemas@gmail.com

Satyan L. Devadoss
Department of Mathematics, University of San Diego, San Diego, 92129 CA
devadoss@sandiego.edu

Yu Xuan Hong
Department of Mathematics, Pomona College, Claremont, 91711 CA
yu_xuan.hong@pomona.edu

Abstract

Given a convex polyhedron, the star unfolding of its surface is obtained by cutting along the shortest paths from a fixed source point to each of its vertices. We present an interactive application that visualizes the star unfolding of a box, such that its dimensions and source point locations can be continuously toggled by the user.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics, Theory of computation → Computational geometry

Keywords and phrases star unfolding, source unfolding, Voronoi diagram

Digital Object Identifier 10.4230/LIPIcs.SoCG.2018.76

Category Multimedia Exposition

1 Star Unfolding

The problem of unfolding polyhedra was popularized by Albrecht Dürer in the early 16th century, when he cut polyhedra along edges and represented them in their unfolded state. This process of edge unfolding created *nets*: connected faces that lay flat in the plane without overlap. One of the most beautiful problems in computational geometry asks whether every convex polyhedron has a net. In the absence of a resolution to this problem, it is of interest to create nets under different rules, allowing the polyhedral surface cuts to be arbitrary rather than only along the edges. The *star unfolding* of Alexandrov [1] is one possible method, dating back to 1948.

▶ **Definition 1.** The *star unfolding* of a convex polyhedron P fixes a generic point x on P, and cuts along the shortest paths from x to every vertex of P.

A source point is *generic* when there is a unique shortest path to each vertex. The reason these cuts suffice to flatten polyhedron P into a polygon $U^*(P)$ is that all the points of curvature (at the vertices) are resolved by these cuts. It is by no means obvious that this resulting polygon $U^*(P)$ does not overlap, and is therefore a net; this was shown to be true in 1992 by Aronov and O'Rourke [2].

Note that if P has n vertices, then the star unfolding $U^*(P)$ is a polygonal net with $2n$ vertices: the n vertices from the polyhedron interleaved with n copies of the source point x. Figure 1 shows the star unfolding produced by cutting along eight paths of a box. Note that two edges of $U^*(P)$ incident to a vertex v of P have the same length — they are the two "sides" of the cut along the shortest path from x to v on P.

34th International Symposium on Computational Geometry (SoCG 2018).
Editors: Bettina Speckmann and Csaba D. Tóth; Article No. 76; pp. 76:1–76:4
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

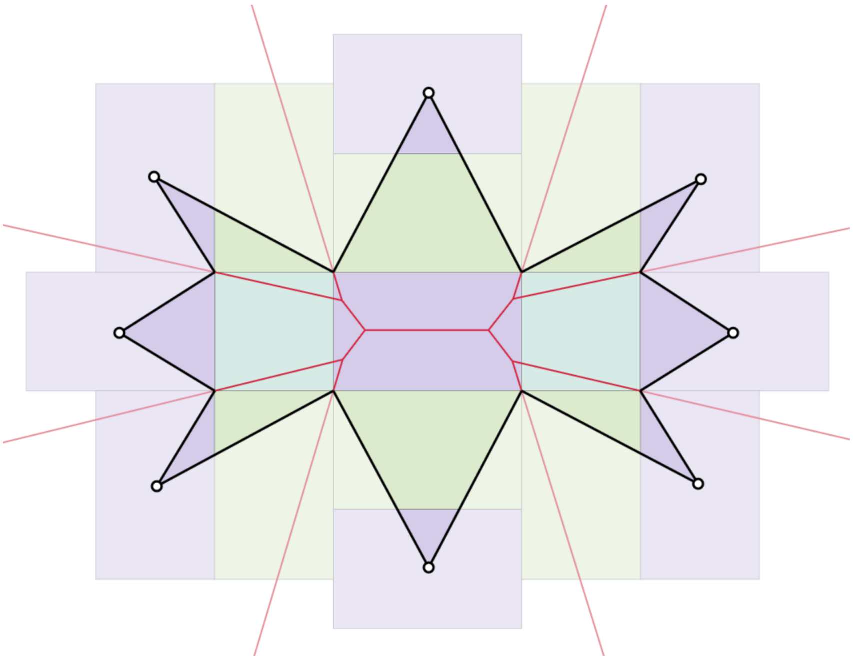

Figure 1 A star unfolding of a box.

Now we turn to another type of unfolding of a convex polyhedron, based on the cut locus.

▶ **Definition 2.** The *cut locus* $C(x)$ of a point x on the polyhedron is (the closure of) the set of all points y to which there is more than one shortest path from x. The *source unfolding* of a convex polyhedron P is obtained by cutting along $C(x)$ of a generic source point x.

The elegant underlying structure that relates the star and source unfoldings is the Voronoi diagram. In particular, the Voronoi diagram of the n copies of the source x, restricted to the interior of $U^*(P)$ (shown as red lines in Figure 1), is the cut locus $C(x)$ on P. Cutting along this Voronoi diagram in the star unfolding and rearranging the resulting n subpolygons yields the source unfolding. Indeed, the source and the star unfoldings are *scissors congruent* to one another [3, Chapter 1].

2 Algorithm for visualization

In our visualization, the user defines the dimensions of the box (up to scaling) along with the location of the source point. Pairs of opposite sides of the box are similarly color-coded, with the source point lying on a purple box face. Two steps are needed to obtain the star unfolding:

1. Find the shortest paths from the source point to each of the eight vertices of the box.
2. The relative order of the eight cuts is then used to determine the adjacency of edges in the star unfolding, allowing its construction.

Given a source point x, assume (without loss of generality) that it lies on the bottom face of the box. The shortest paths from x to the four vertices of the bottom face are simply straight line segments. To find the shortest paths from x to an arbitrary vertex v of the top face (as shown in Figure 2), there are six cases to consider: a path that goes through the bottom face and (1) the left face, (2) the back face, (3) the front and left faces, (4) the right

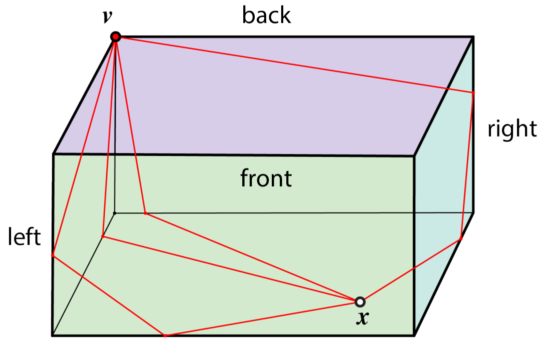

Figure 2 Four possible cases of shortest paths from source point x to vertex v.

and back faces, (5) the front and top faces, and (6) the right and top faces. However, since x is on the bottom face, it is straight forward to show that paths from (5) and (6) will never be shorter than those from (1) and (2), respectively. And so, there are only four cases that have a chance of yielding the shortest path to v, depending on the location of x. We calculate the shortest distance attainable in each of the four cases, and choose the path with minimal distance as a cut of the star unfolding.

Our visualization displays the background rectangles representing the faces that each cut passes through. The faces are unfolded onto the plane, so that each shortest cut is a straight line segment from a copy of the source point to a vertex; see Figure 3. Owing to the orthogonal nature of a box, adjacent edges that share a vertex are perpendicular to each other in the unfolding.

The Voronoi diagram of the eight copies of the source point is also drawn. Notice that when the source point crosses over the Voronoi diagram (with equidistant shortest paths to a vertex), the shape of the star unfolding will change, along with the background rectangles.

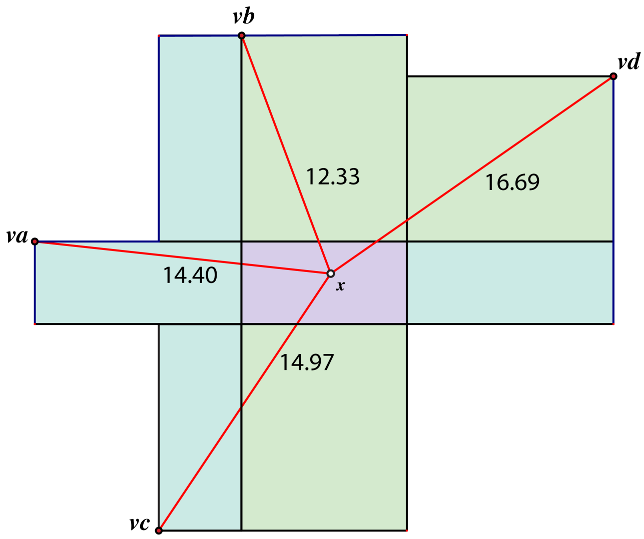

Figure 3 An example of possible cases of shortest path given source point x. The four points va, vb, vc, vd are copies of the same vertex v unfolded in ways corresponding to cases (1), (2), (3), and (4), respectively. In this example, the shortest path is to vertex copy vb.

3 Implementation

Our visualization is a browser application implemented in HTML5 and JavaScript, and can be accessed at `http://ddemas.github.io/box-unfolding-visualization/`. The user may adjust the *width*, *height*, and *length* of the box (up to rescaling) using the slide bars. The source point, located on the purple boxes, can also be altered, with dimensions *width* and *height*. The user also has the option to display or hide the Voronoi lines, background rectangles, and star unfolding.

The visualization works by constantly recalculating and redrawing the unfolding, using JQuery and Bootstrap libraries to help render the UI. The rectangle in the center and the four rectangles adjacent to it are fixed and represent five unique faces of the box. The rest of the rectangles change depending on the configuration that leads to the shortest path from the source point in the four corners to the nearest corner of the center rectangle, as explained in Section 2. The shortest distance to the remaining vertices of the box is inferred from the configuration of the rectangles in each corner.

Once the points of the box have been determined, we use an open source implementation of Fortune's algorithm written by Raymond Hill [4] to draw the Voronoi lines.

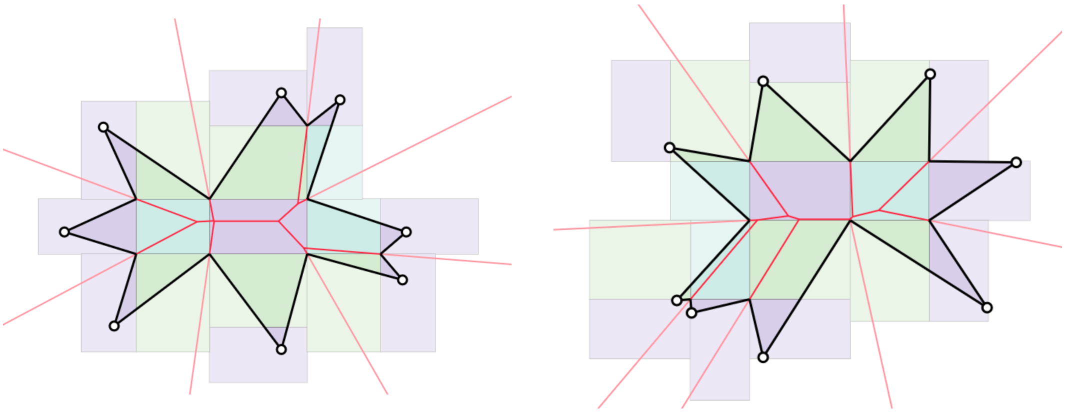

Figure 4 Screenshots from visualization that show examples from of different star unfoldings of the same box with different source points, with their underyling Voronoi diagrams.

References

1 A.D. Alexandrov. *Convex Polyhedra*. Springer Monographs in Mathematics. Springer Berlin Heidelberg, 2006. URL: `https://books.google.com/books?id=aoMreDT_DwcC`.

2 Boris Aronov and Joseph O'Rourke. Nonoverlap of the star unfolding. *Discrete & Computational Geometry*, 8(3):219–250, Sep 1992. `doi:10.1007/BF02293047`.

3 Satyan L. Devadoss and Joseph O'Rourke. *Discrete and Computational Geometry*. Princeton University Press, 2011.

4 Raymond Hill. Javascript-voronoi. `https://github.com/gorhill/Javascript-Voronoi`, 2013.