6.849: GEOMETRIC FOLDING ALGORITHMS, SPRING 2017

Prof. Erik Demaine, Martin Demaine, Adam Hesterberg, Jason Ku, Jayson Lynch

Problem Set 4 Solutions

Due: Wednesday, March 8, 2017

Solve Problem 4.1 and *either* Problem 4.2 or 4.3.

Problem 4.1 [Mandatory, Collaboration OK]. On each problem set, we will ask you to write a problem (solved or unsolved) related to the material covered in class. The problem should be original to the best of your knowledge, so be creative and diverse! Folding can be applied to mathematics, computation, engineering, architecture, biology, and beyond, so write a problem that is related to a field that interests you. If you write a problem whose solution can be solved from the material covered in class, then we may adapt your problem for future problem sets. If you pose a problem whose solution is not yet known, we may try to solve it in class during our open problem sessions, or it may become inspiration for a class project. Feel free to include solutions or commentary for your problem. While writing a problem is required, your submission will be graded generously, so have fun and share with us your exploration of the course material.

Solve ONE of the two problems below.

Problem 4.2 [Collaboration OK]. Design and fold (but do not cut) a fold-and-cut model using the straight-skeleton method. Submit a PDF of your design (including crease pattern, in vector format) on Gradescope, and submit the folded version physically. We highly recommend that you use a vector drawing program that can compute accurate intersections, such as Inkscape (free), Cinderella (mostly free), Adobe Illustrator (commercial), AutoDesk Fusion 360 (free for students), or Rhino3D (commercial). Use your judgment of reasonable complexity to work within your folding ability.

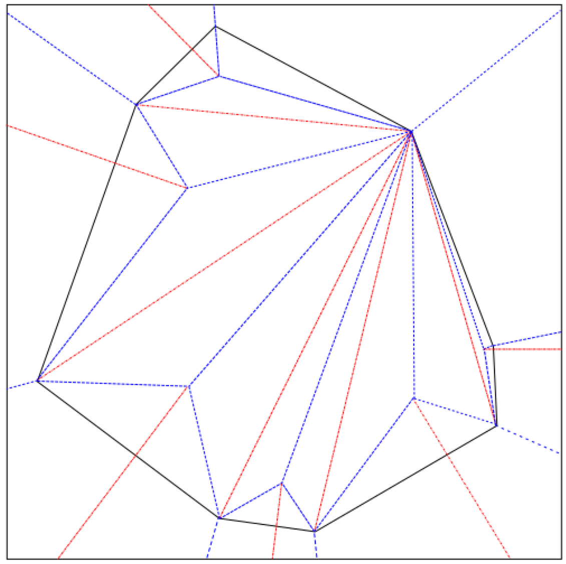
Problem 4.3 [Collaboration OK]. Consider a convex polygon P whose vertices (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) satisfy $-1 \le x_i, y_i \le 1$. You seek an efficient algorithm describing how to fold and cut P from a square piece of paper with vertices $(\pm 1, \pm 1)$.

(a) Show that there exists a crease pattern with $O(n)$ creases which can be folded so that one cut produces P.

(b) Find as efficient an algorithm as you can to compute such a crease pattern. Any correct algorithm will earn at least partial credit. Can you get $O(n \log n)$ or even $O(n)$ time?

Solution by Christine Soh and Albert Soh:

(a) We simply triangulate the polygon by choosing a vertex and making a mountain crease from it to each non-adjacent vertex ($n-3 = O(n)$ creases). This creates $n-2$ triangles; in each of these triangles, make valley creases from the incenter to all three vertices (folding along the angular bisector) and one mountain crease from the incenter to an edge that triangle shares with the polygon ($4(n-2) = O(n)$ creases). Finally, make a valley crease that is the angle bisector of the reflex angle of each vertex out to the edge of the paper ($n = O(n)$ creases). Altogether, there are $O(n)$ creases.



(b) The triangulation in the previous section suffices. Or:

The convex polygon P has vertices $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. Let's call the starting vertex (x_n, y_n). We create mountain creases from this vertex to all the non-adjacent vertices $(x_2, y_2), (x_3, y_3), ..., (x_{n-2}, y_{n-2})$, creating $n-2$ triangles, which will be indexed by the ith vertex such that the triangle T_1 has vertices $(x_1, y_1), (x_2, y_2), (x_n, y_n)$, and the triangle T_i has vertices $(x_i, y_i), (x_{i+1}, y_{i+1}), (x_n, y_n)$. There are $n-2$ creases made, and assuming that making a crease (i.e. drawing a colored line between two points on the coordinate system) takes $O(1)$ time, the run time for this part is $O(n)$.

The triangle T_i has sides $s_{i_n} = v_i v_{i+1}$, $s_{i_i} = v_n v_{i+1}$, and $s_{i_{i+1}} = v_{i+1} v_n$, and the side lengths are as follows:

$$|s_{i_1}| = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2},$$
$$|s_{i_i}| = \sqrt{(x_{i+1} - x_1)^2 + (y_{i+1} - y_1)^2},$$
$$|s_{i+1}| = \sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2}.$$

The coordinates of the incenter o_i has coordinates (o_{i_x}, o_{i_y}) such that

$$o_{i_x} = \frac{|s_1|x_1 + |s_i|x_i + |s_{i+1}|x_{i+1}}{|s_1| + |s_i| + |s_{i+1}|}$$
$$o_{i_y} = \frac{|s_1|y_1 + |s_i|y_i + |s_{i+1}|y_{i+1}}{|s_1| + |s_i| + |s_{i+1}|}$$

Create creases from the incenter of each triangle to the three vertices of the triangle. This should take $O(n)$ time as well.

Next, we crease the perpendicular down from the incenter through the side of the polyhedron all the way to the edge of the paper for each triangle. To find the point on the side of the polygon where the perpendicular intersects in order to go through the incenter, we can project the vector $< v_i, o_i >$ onto the vector $< v_i, v_{i+1} >$ and find the endpoint. Creating these creases will take $O(n)$ time.

Finally, we need the angle bisectors of the reflex angles of each vertex. For vertex v_i, consider the two side unit vectors $\vec{s_{i_1}} =< \frac{v_{i-1}}{\sqrt{(x_{i-1}-xi)^2+(y_{i-1}-y_i)^2}}, \frac{v_i}{\sqrt{(x_{i-1}-xi)^2+(y_{i-1}-y_i)^2}} >$ and $\vec{s_{i_2}} =< \frac{v_i}{\sqrt{(x_{i+1}-xi)^2+(y_{i+1}-y_i)^2}}, \frac{v_{i+1}}{\sqrt{(x_{i+1}-xi)^2+(y_{i+1}-y_i)^2}} >$ The vector of the angle bisector is $\vec{s_{i_1}} + \vec{s_{i_2}}$. Fold along this vector to the edge of the triangle, which I also will assume will take $O(1)$ time. Since there are n vertices, this will also take $O(n)$ time, which will result in an overall runtime of $O(n)$. This is based on the assumption that the runtime for creasing and finding the incenter and angle bisectors is in constant ($O(1)$) time.