Computational Geometry — Fall 2020-21 — Dan Halperin

Assignment no. 2

due: Monday, November 30th, 2020

Exercise 2.1 Give an efficient algorithm to determine whether a simple polygon with n vertices is monotone with respect to some given line, not necessarily a vertical or horizontal one. Analyze the running time of your algorithm.

Exercise 2.2 The *stabbing number* of a triangulation of a simple polygon P is the maximum number of diagonals intersected by any line segment interior to P. Give an algorithm that computes a triangulation of a convex polygon that has stabbing number $O(\log n)$.

Exercise 2.3 Prove that the following polyhedron $\mathcal{P}$ cannot be tetrahedralized using only vertices of $\mathcal{P}$, namely its interior cannot be partitioned into tetrahedra whose vertices are selected from the vertices of $\mathcal{P}$ (see the enclosed figure).[1]

> Let a, b, c be the vertices (labeled counterclockwise) of an equilateral triangle in the xy-plane. Let a', b', c' be the vertices of abc when translated up to the plane $z = 1$. Define an intermediate polyhedron $\mathcal{P}'$ as the hull of the two triangles including the diagonal edges ab', bc', and ca', as well as the vertical edges aa', bb', and cc', and the edges of the two triangles abc and $a'b'c'$. Now twist the top triangle $a'b'c'$ by 30° in the plane $z = 1$, rotating and stretching the attached edges accordingly: this is the polyhedron $\mathcal{P}$.

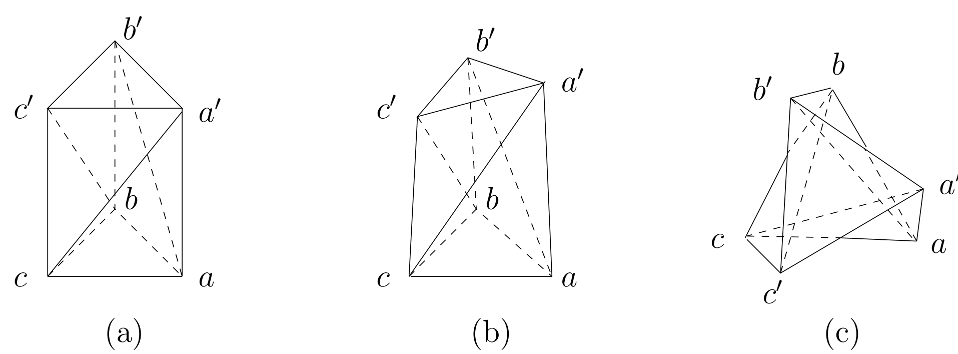

Figure 1: The untetrahedralizable polyhedron is constructed by twisting the top of a triangular prism (a) by 30° degrees, producing (b), shown in top view in (c)

Notice that there are additional exercises on the other side of the page.

[1] This construction is due to Schönhardt, 1928. The description here is taken from O'Rourke's *Art Gallery Theorems and Algorithms.*

Exercise 2.4 In each of the following settings, describe a construction (a polygon and guard placements) where the specified **vertex guards** do not fully cover the polygon in the art gallery sense, and such that your construction could be generalized to any number of vertices, as specified in the setting:
(a) A simple polygon with $2k$ vertices, for every $k > 2$, and a specific assignment of guards placed at **every other vertex** along the boundary of the polygon. Namely, guards placed at the vertices $v_i, v_{i+2}, v_{i+4}, \ldots$, do not fully cover the polygon.
(b) Similarly, a simple polygon with $3k$ vertices, for every $k > 2$, and a specific assignment of guards placed at **every third vertex** along the boundary of the polygon.
(c) A simple polygon with n vertices, for every $n > 5$, and guards placed only at **convex vertices**. A vertex is convex if its interior angle is less than π.

Exercise 2.5 (Optional, bonus) The *pockets* of a simple polygon are the areas outside the polygon, but inside its convex hull. Let P_1 be a simple polygon with n_1 vertices, and assume that a triangulation of P_1 as well as of its pockets is given. Let P_2 be a convex polygon with n_2 vertices. Show that the intersection $P_1 \cap P_2$ can be computed in $O(n_1 + n_2)$ time. (CGAA Ex. 3.12)

Exercise 2.6 (Self learning, do not submit) The first part **(a)** is in preparation for the next topic.
Let $f_i(x), i = 1, \ldots, n$ be a set of functions. The lower envelope Ψ of this set of functions is the pointwise minimum of these functions: $\Psi(x) = \min_i f_i(x)$.
(a) Assume that the functions are linear, namely $f_i(x) = a_i x + b_i$. We divide the x-axis into maximal interval such that for each interval the minimum is attained by the same function f_i.
(a1) What is the maximum number of such maximal intervals for any collection of linear functions.
(a2) Describe a divide-and conquer algorithm to efficiently compute the lower envelope of a set of linear functions. Analyze the time and space required by the algorithm.
(b, harder) The same as (a), only this time the functions are parabolas, namely $f_i(x) = a_i x^2 + b_i x + c_i$.