CS310 Computational Geometry
Spring 2019

Assignment:	**6**	**Professor:**	**Dianna Xu**
Due Date:	**4/9/19**	**E-mail:**	`dxu@cs.brynmawr.edu`
Office:	**Park 203**	**URL:**	`http://cs.brynmawr.edu/cs310`

Assignment 6

1. Exercise 5.3.

2. Exercise 5.10.

3. Exercise 5.16.

4. We have seen that the worst-case complexity of the Minkowski sum $P \oplus R$ of two polygons P and R each with n and m edges respectively, might range from $O(n+m)$ (both convex) to as high as $O(n^2m^2)$ (both nonconvex), which is quite a gap. Let us consider an intermediate but realistic situation. Suppose that we assume that P is an arbitrary n-sided simple polygon, and R is a convex m-sided polygon. Typically m is much smaller than n. What is the combinatorial complexity of $P \oplus R$ in the worst case? and why? Draw an example of some P and R that achieve the worse case complexity.

5. We now consider a translating robot given as an m-sided convex polygon moving among a collection of polygonal obstables P_i having a total of n vertices. First we need some way of extracting the special geometric structure of the union of Minkowski sums. Recall that we are computing the union of $P_i \oplus R$, where the P_i's have disjoint interiors. A set of convext objects $\{o_1, \ldots, o_n\}$ is called a *collection of pseudodisks* if for any two distinct objects o_i and o_j both of the set-theoretic differences $o_i \setminus o_j$ and $o_j \setminus o_i$ are connected (see Figure 1).

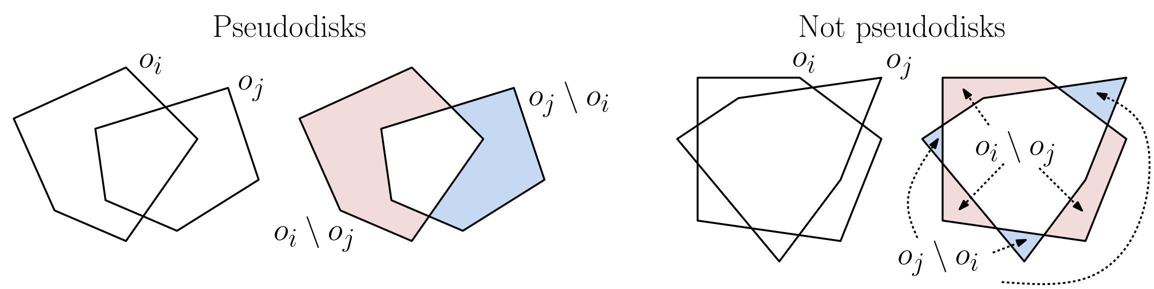

Figure 1: Pseudodisks

(a) Prove the following lemma: Given a set of convex objects $T_1, \ldots, T_n$ with disjoint interiors, and convex R, the set

$$\{T_i \oplus R | 1 \leq i \leq n\}$$

is a collection of pseudodisks (see Figure 2).

(b) Prove that given a collection of polygonal pseudodisks, with a total of n vertices, the complexity of their union is $O(n)$.

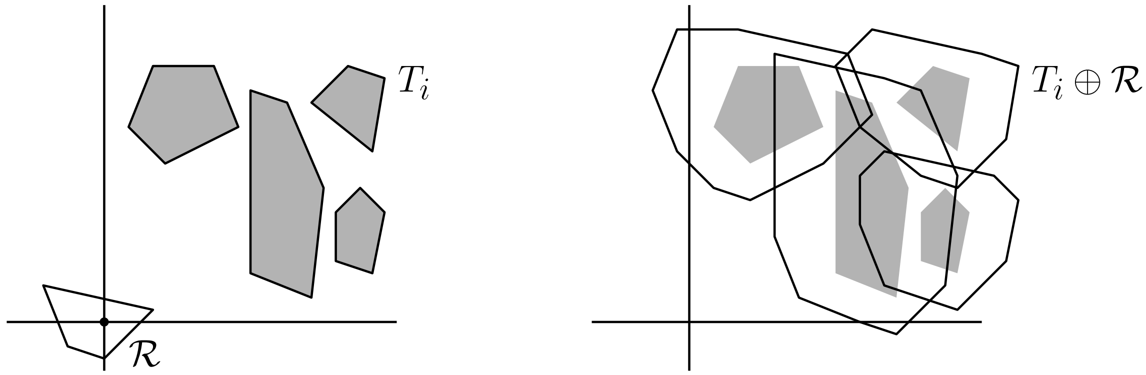

Figure 2: Union of Minkowsi sums

(c) Putting 5(a) and 5(b) together, what can you say about the combinatorial complexity of the Minkowski sum of $P_i \oplus R$?

6. Implementation O'Rourke Exercise 5.33. Write a program that takes as input a polygon (a list of vertices on the boundary, in ccw order), and repeatedly applies the midpoint transformation to it. Perhaps take the number of iterations as input as well. Work out some way to display the results, so you can see the shape after all iterations are completed. Form a conjecture based on experiments with your code. You should do enough experiments to make sure that you are not forming a theory based only on special cases.