Polygon Area Problems

Ralph P. Boland* Jorge Urrutia†

Abstract

In this paper we study the problem of preprocessing a simple polygon so that, for any query chord of the polygon, the area of the subpolygons determined by the chord can be determined quickly. We give a solution to this problem requiring linear space and preprocessing time and constant query time. This is an improvement by a factor of $O(\log n)$ in space, preprocessing time, and query time over the best known algorithm. Furthermore our solution is simpler as well; the most complex operation involves the computation of the area below a chain of length three. Finally we show that our algorithm can be used to solve a number of closely related problems in the same time complexities.

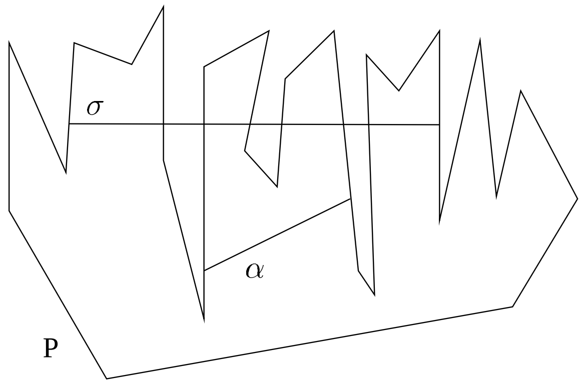

Figure 1.1: *A polygon P with a chord α and a pseudo chord σ (P_σ is simple).*

1 Introduction

One of the basic problems of geometry is the computation of the area or volume of geometric objects. In [2] a number of problems and algorithms are described relating to the computation of the area or volume of geometric objects. We focus on one these problems.

Let $P = \{v_0, \ldots, v_{n-1}\}$ be a polygon and σ be a chord of P. We denote by P_σ the subpolygon of P determined by σ that contains v_0 on its boundary unless both of the subpolygons contain v_0 in which case P_σ denotes the subpolygon that contains v_{n-1} on its boundary.

Problem 1.1. Let $P = \{v_0, \ldots, v_{n-1}\}$ be a simple polygon. Preprocess P so that for a query chord σ of P we can quickly determine the area of P_σ.

The solution to this problem presented in [2] requires $O(n \log n)$ preprocessing, $O(n \log n)$ space and $O(\log n)$ query time; see also [3]. In this paper we present a solution to this problem that requires linear preprocessing and space and $O(1)$ query time and furthermore our solution is much simpler. We also show that our algorithm can be used to solve several generalizations of problem 1.1 with the same preprocessing, space, and query complexities. Clearly, for all of these problems, our solutions are optimal.

Our first step is to generalize Problem 1.1. We call a line segment whose endpoints are on the boundary of a polygon P a *pseudo chord* of P. Note that, unlike a chord of P, a pseudo chord of P may intersect the boundary of P at points other than the endpoints of the pseudo chord and may do so multiple times. A pseudo chord of P may also intersect the exterior of P.

Problem 1.2. Let $P = \{v_0, \ldots, v_{n-1}\}$ be a simple polygon. Preprocess P so that for a query pseudo chord σ of P, such that P_σ is a simple polygon, we can quickly determine the area of P_σ.

This problem includes Problem 1.1 as a special case. However, the algorithm in [2] for solving Problem 1.1 cannot be used to solve Problem 1.2. In fact, at first glance, this problem appears to require linear time since σ may intersect $O(n)$ edges of P; see Figure 1.1. We show however, that like Problem 1.1, this problem can be solved with linear preprocessing and space and $O(1)$ query time. Furthermore the algorithm we present to solve this problem is very simple; it doesn't even require the triangulation of P. We also show that the algorithm can be generalized to work on self-overlapping polygons (see [4]) once the area for such polygons is defined.

*S.I.T.E. University of Ottawa, Canada
email: rboland@site.uottawa.ca
†Instituto de Mathematicas Universitidad Nacional Autonoma de Mexico
email: urrutia@math.unam.mx

2 Calculating the Area of a Polygon

There are numerous methods of calculating the area of a simple polygon; for a survey see [2]. Of particular interest to us is the method known as the polar formula. For any line segment $\sigma = \overline{(x_a, y_a)(x_b, y_b)}$ we denote $\overline{\mathbb{A}}(\sigma) = x_a y_b - x_b y_a$. We point out that $\overline{\mathbb{A}}(\sigma)$ equals twice the area of the triangle whose vertices are the origin and the endpoints of σ if $a \geq b$ and the negative of this value if $a < b$.

We denote the area of a polygon P by $\mathbb{A}(P)$. Let $P = \{v_0, e_0, v_1, e_1, \ldots, e_{n-1}\}$ be a simple polygon. Then the polar formula for the area of P can be written:

$$\mathbb{A}(P) = \frac{1}{2}\sum_{i=0}^{n-1} \overline{\mathbb{A}}(e_i) \tag{2.1}$$

Note that the right hand side of equation 2.1 can be applied to any polygon; that is the simplicity of P is not assumed. We now turn equation 2.1 around and say that equation 2.1 defines the area of P. Thus the area of every polygon is defined and if the polygon is simple then our definition of area corresponds to the normal definition.

Now to solve Problem 1.2 and thus also Problem 1.1 we use the idea of partial sums. For each edge $e_k \in V(P)$ we denote:

$$s_k = \sum_{i=0}^{k} \overline{\mathbb{A}}(e_i) \tag{2.2}$$

We say that a pseudo chord σ of P is a pseudo diagonal of P if the endpoints of σ are vertices of P. Let σ be a pseudo diagonal of P with endpoints v_j, v_k where $j < k - 1$. Then by equations 2.1 and 2.2 we have:

$$\begin{aligned} \mathbb{A}(P_\sigma) &= \frac{1}{2}\left(\sum_{i=0}^{j-1} \overline{\mathbb{A}}(e_i) + \overline{\mathbb{A}}(\sigma) + \sum_{i=k}^{n-1} \overline{\mathbb{A}}(e_i)\right) \\ &= \mathbb{A}\{P\} + \frac{-s_{k-1} + s_{j-1} + \overline{\mathbb{A}}(\sigma)}{2} \\ &= \frac{s_{n-1} - s_{k-1} + s_{j-1} + \overline{\mathbb{A}}(\sigma)}{2} \end{aligned} \tag{2.3}$$

Consider now a chain C with the same endpoints as σ. Let

$$C = \{v_j = w_0, h_0, w_1, h_1, \ldots, h_{m-1}, w_m = v_k\}$$

where the vertices of C are $\{w_i : 0 \leq i \leq m\}$ C and the edges of C are $\{h_i : 0 \leq i < m\}$. Consider the polygon

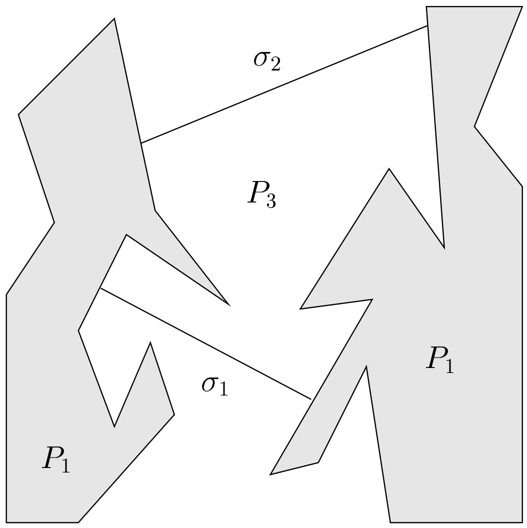

Figure 2.1: Using methods similar to those described here the polygons P_1 and P_2 can be preprocessed in linear time so that the area of a query polygon P_3 determined by line segments σ_1 and σ_2 can be computed in constant time.

$$\begin{aligned} P_C \quad = \quad & \{v_0, e_0, v_1, \ldots, v_j, h_0, w_1, h_1, w_2, \ldots, \\ & w_{m-1}, h_{m-1}, v_k e_k, v_{k+1}, \ldots, v_{n-1}, e_{n-1}\}. \end{aligned}$$

By equations 2.1 and 2.3 we have:

$$\mathbb{A}(P_C) = \frac{s_{n-1} - s_{k-1} - s_{j-1} + \sum_{i=0}^{m-1} \overline{\mathbb{A}}(h_i)}{2} \tag{2.4}$$

Consider now that σ is a pseudo chord of P with endpoints p_j, p_k such that $p_j \in \overline{v_j v_{j+1}}$ and $p_k \in \overline{v_{k-1} v_k}$. Then we can compute $\mathbb{A}(P_\sigma)$ by computing $\mathbb{A}(P_C)$ where the vertices of C in order are v_j, p_j, p_k, v_k.

We now have a method to compute $\mathbb{A}(P_\sigma)$; use C in equation 2.4. What will be the cost to compute $\mathbb{A}(P_\sigma)$? Note first that, for $0 \leq i \leq n-2$, clearly $s_{i+1} = s_i + \overline{\mathbb{A}}(e_i)$. Therefore the values s_i, for $0 \leq i < n$, can easily be computed in linear time in total. From these facts and Equation 2.4 we obtain the following result:

Theorem 2.1. *Problem 1.2 (and thus also Problem 1.1) can be solved with linear preprocessing, linear space, and a query complexity of* $O(1)$.

We can apply the results of this section to situations involving more than one polygon. Consider for example the following problem.

Problem 2.2. Let P_1, P_2 be two disjoint simple polygons. Preprocess P_1 and P_2 so that for a query simple

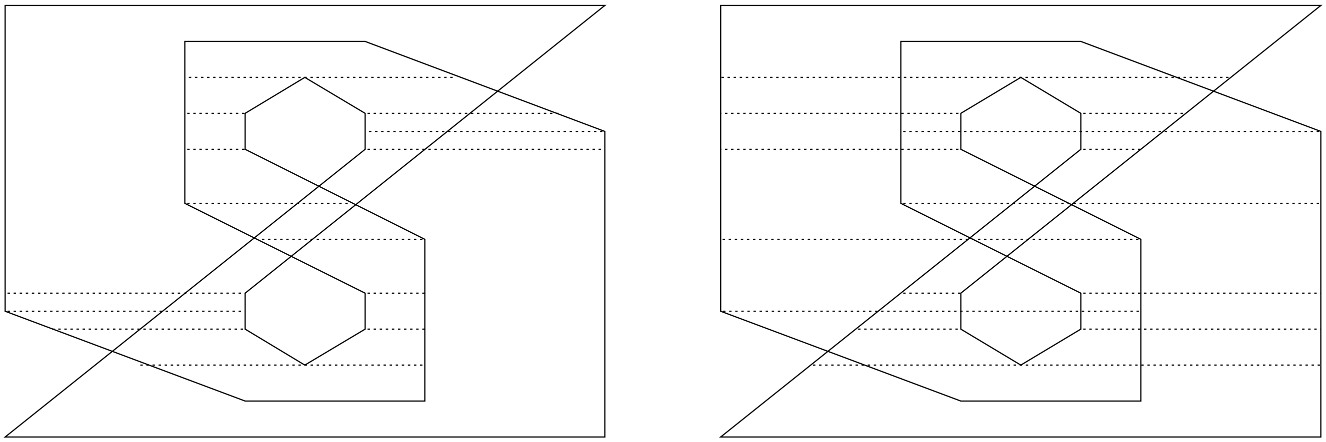

Figure 2.2:

A self-overlapping polygon may have multiple distinct trapezoidizations. However the total area of all trapezoids in a trapezoidization is independent of the trapezoidization chosen.

polygon P_3 whose boundary is constructed from a subchain C_1 of P_1, a subchain C_2 of P_2, and pair of line segments σ_1, σ_2 joining C_1 and C_2 the area of P_3 can be computed quickly.

It can be shown by methods similar to the above that Problem 2.2 can be solved with linear preprocessing and space and a query complexity of $O(1)$. See Figure 2.1.

3 Areas of Self-overlapping Polygons.

Self-overlapping polygons [4] have the interesting property that, even though they are not generally simple polygons, they can, by the definition of triangulation given in [4], be triangulated. Equivalently we can construct a horizontal trapezoidization of a self-overlapping polygon. However, in contrast to simple polygons, a self-overlapping polygon may have multiple distinct horizontal trapezoidizations; see Figure 2.2 and also [4]. This fact makes the question of the area of a self-overlapping polygon an interesting one.

We note first that the following observation follows directly from equation 2.1.

Lemma 3.1. *Let P be a (not necessarily simple) polygon and let σ be a pseudo chord of P. Let P_1 and P_2 be the two subpolygons of P determined by σ. Then $\mathbb{A}(P) = \mathbb{A}(P_1) + \mathbb{A}(P_2)$.*

It is easily shown using induction and Observation 3.1 that the sum of the areas of the trapezoids of a horizontal trapezoidization of a self-overlapping polygon is equal to the area computed for the polygon by Equation 2.1. Thus no matter which horizontal trapezoidization is used (assuming that there is more than one) the sum of the areas of the trapezoids will be the same. We summarize with the following result:

Lemma 3.2. *Let P be a self-overlapping polygon with more than one distinct horizontal trapezoidization and let T_1, T_2 be two distinct horizontal trapezoidizations of P. Let $\mathbb{A}_1$ and $\mathbb{A}_2$ be the sum of the areas of the trapezoids of trapezoidization T_1 and T_2 respectively. Then $\mathbb{A}_1 = \mathbb{A}_2$.*

In a straightforward manner we can generalize Lemma 3.2 to apply to self-overlapping curves [4, 6].

A comment about computing volumes of polyhedrons in three or higher dimensions is worthwhile. We point out that Equation 2.1 and Observation 3.1 can be generalized to apply to (not necessarily simple) polyhedrons of any fixed dimension; see [2, 5, 1]. Furthermore the concept of self-overlapping polygons can also be generalized to self-overlapping polyhedrons of any fixed dimension. It must therefore be the case that Lemma 3.2 also generalizes to self-overlapping polyhedrons of any fixed dimension.

A final nice point about Equations 2.1 through 2.4 is that the only operations are multiplications, additions, subtractions, and at most one divide (by two). Thus if the vertices of P and the endpoints of σ have only integer coordinates and all the x-coordinates (say) are a multiple of two then no floating point operations are required to compute the area of P_σ. Under these conditions $\mathbb{A}(P_\sigma)$ is computed exactly using only basic integer arithmetic.

References

[1] Eugene L. Allgower. Computing volumes of polyhedra. *Math. Comput.*, 46:171–174, 1986.

[2] Felipe Contreras-Alcalá. *Cutting polygons and a problem on illumination of stages.* Masters thesis, Dept. Comput. Sci.University of Ottawa., Ottawa Ont, Canada, 1998.

[3] Jurek Czyzowicz, Felipe Contreras-Alcalá, and Jorge Urrutia. On measuring areas of polygons. In *Proceedings of the 10th Canadian Conference on Computational Geometry*, page (to apear), Montréal, QC, Canada, August 1998. McGill University. Extended Abstract.

[4] P. W. Shor and C. J. Van Wyk. Detecting and decomposing self-overlapping curves. *Comput. Geom. Theory Appl.*, 2(1):31–50, 1992.

[5] T. Speevak. An efficient algorithm for obtaining the volume of a special kind of pyramid and application to convex polyhedra. *Math. Comput.*, 46:531–536, 1986.

[6] H. Whitney. On regular closed curves in the plane. *Compositio Math.*, 4:276–284, 1937.