Counting Convex Polygons in Planar Point Sets

Joseph S. B. Mitchell [a,1], Günter Rote [b], Gopalakrishnan Sundaram [c], and Gerhard Woeginger [b]

[a] *Applied Mathematics and Statistics, SUNY Stony Brook, NY 11794-3600. email: jsbm@ams.sunysb.edu*

[b] *Technische Universität Graz, Institut für Mathematik, Steyrergasse 30, A-8010 Graz, Austria. email: rote@ftug.dnet.tu-graz.ac.at, woeginger@fmatbds01.tu-graz.ac.at*

[c] *Environmental Systems Research Institute, 380 New York St., Redlands, CA 92373. email: gsundaram@esri.com*

Abstract

Given a set S of n points in the plane, we compute in time $O(n^3)$ the total number of convex polygons whose vertices are a subset of S. We give an $O(m \cdot n^3)$ algorithm for computing the number of convex k-gons with vertices in S, for all values $k = 3, \ldots, m$; previously known bounds were exponential ($O(n^{\lceil k/2 \rceil})$). We also compute the number of *empty* convex polygons (resp., k-gons, $k \leq m$) with vertices in S in time $O(n^3)$ (resp., $O(m \cdot n^3)$).

Key words: Computational geometry, convexity, combinatorics, dynamic programming

1 Introduction

Let S denote a set of n points in the plane. A subset $T \subseteq S$ is said to be in *convex position* if T is the vertex set of a convex polygon, and we then say that T *determines a convex k-gon*, where $k = |T|$. We say that a polygon is *empty* if it contains no point of S in its interior.

[1] Partially supported by grants from Boeing Computer Services, Hughes Research Laboratories, Air Force Office of Scientific Research contract AFOSR-91-0328, and by NSF Grants ECSE-8857642 and CCR-9204585.

If S is in convex position, then there are exactly $\binom{n}{k}$ convex k-gons determined by subsets of S. In general, however, S may determine far fewer convex k-gons. Consider, for example, a set S of $n = 3K$ points, with K points along each of three rays emanating from the origin, such that the three rays positively span the plane. Then there is no convex k-gon determined by a subset of S for any value of $k \geq 5$.

In this note we show that the total number of convex polygons determined by S can be computed in time $O(n^3)$. Further, we show that the number of convex k-gons determined by S can be tabulated, for all values of $k = 3, \ldots, m$, in total time $O(m \cdot n^3)$. Within these same time bounds, we can compute the total number of empty convex polygons determined by S (in time $O(n^3)$) or tabulate, for $k = 3, \ldots, m$, the number of empty convex k-gons determined by S (in time $O(m \cdot n^3)$). Finally, we can compute for a given point u (not necessarily from the set S) the number of convex k-gons determined by S that contain u in total time $O(m \cdot n^3)$, for all $k \leq m$. In general, the numbers that we compute can be large — e.g., n points in convex position determine roughly 2^n convex polygons. We assume a real RAM model of computation in which arithmetic operations on large integers can be done in constant time.

Relation to previous work. Khuller and Mitchell [6] showed how to compute the number of triangles determined by S containing each point $p \in S$ in total time $O(n^2)$. Rote et al. [8] showed how to count the total number of convex k-gons determined by S in time $O(n^{k-2})$, improving over the trivial bound of $O(n^k)$. This result has recently been improved to $O(n^{\lceil k/2 \rceil})$ by Rote and Woeginger [9]. Unfortunately, these bounds are exponential in k. Our bounds are polynomial in n and k.

Counting problems are closely related to *optimization* problems for convex polygons, because both types of problems show different aspects of their natural common generalization: the enumeration of convex polygons. Furthermore, both types of problems can be solved by dynamic programming. This is also the approach that we will take here. The details of our recursions are similar to some recursions applied by Arkin, Khuller, and Mitchell [1] in the context of various optimization problems associated with selecting subsets of S to enclose with a "fence", or similar recursions of Eppstein et al. [5] for computing polygons of smallest area. Chvátal and Klincsek [2] also apply similar recursions to the problem of searching for maximum weight convex point sets (see also the book of Korte, Lovász, and Schrader [7], pp. 170–171). An overview of various optimization problems associated with polygons determined by a given point set can be found in Eppstein [4].

2 Counting all convex polygons

Consider first the problem of computing the total number of convex polygons determined by S. For simplicity of presentation, we assume that no three points of S are colinear and that no two points of S have the same y-coordinate; our methods easily extend to the degenerate cases. Fix attention on one point $s \in S$; we will compute in time $O(n^2)$ the number of convex polygons determined by S such that s is the lowest vertex of the polygon. Let U_s denote the open halfspace of all points whose y-coordinate is greater than that of s.

Following the "sweep-line" algorithm of [1] for computing maximum value enclosures, we devise a set of recursions based on imagining a "sweep-ray" rotating clockwise about s, starting with a leftward ray and ending with a rightward ray out of s. Let $H_{p,q}$ denote the open halfspace to the left of the oriented line pq.

Let p and q be points of $S \cap U_s$, such that $q \in H_{s,p}$ (i.e., q is counterclockwise from p with respect to s, and $p \neq q \neq s$). Then, we define $f(p, q; s)$ to be the number of convex polygons which (a) have vertices among the points S; (b) lie in the closure of the cone $U_s \cap H_{s,p}$; and (c) have qp and ps as two edges. Refer to Figure 1. Then, the number $f(p, q; s)$ is obtained as follows: We test

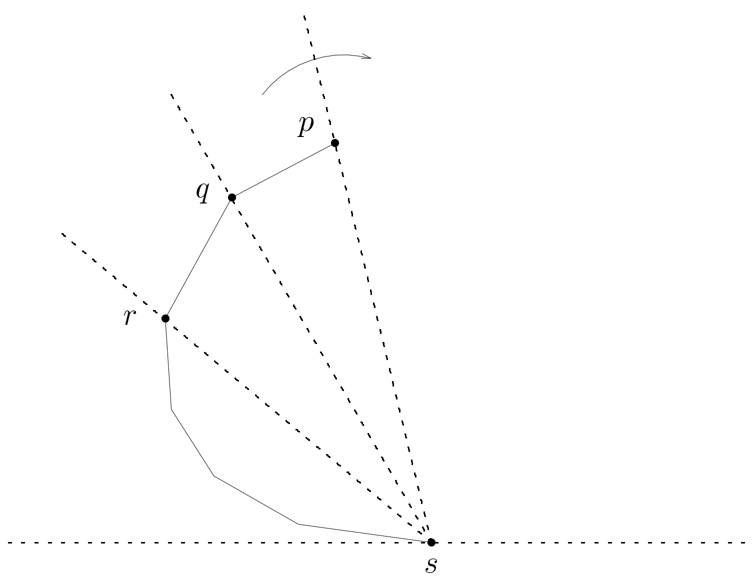

Fig. 1. Notation used in formulating the recursions.

each point $r \in U_s$; if $r \in H_{p,q} \cap H_{s,q}$, then we add up the numbers $f(q, r; s)$, obtaining

$$f(p, q; s) = 1 + \sum_{r \in U_s \cap H_{p,q} \cap H_{s,q}} f(q, r; s). \tag{1}$$

This recursion can be evaluated systematically, if we tabulate the values of $f(p, q; s)$, for all $q \in S \cap U_s \cap H_{s,p}$, for p in clockwise angular order about s. The justification of the expression for $f(p, q; s)$ is simple: A convex polygon satisfying (a)–(c) is either a triangle (with third edge qs) or is obtained by attaching the triangle pqs to a convex polygon counted in $f(q, r; s)$, for some $r \in S$ in the cone $U_s \cap H_{s,q}$ that lies left of the oriented line pq.

As written, these recursions can be evaluated in $O(n)$ time for each choice of p, q, s, and overall $O(n^4)$ time. But this can be improved by noting that, for fixed values of q and s, we can evaluate $f(p, q; s)$ incrementally for points $p \in U_s \cap H_{q,s}$ in clockwise order about q. Specifically, if the points of S in $H_{q,s}$ are labelled $p_1, p_2, \ldots$, in clockwise order about q, then we can compute $f(p_i, q; s)$ from $f(p_{i-1}, q; s)$ according to

$$f(p_i, q; s) = f(p_{i-1}, q; s) + \sum_{r \in H_{p_i,q} \cap H_{q,p_{i-1}}} f(q, r; s). \qquad (2)$$

Refer to Figure 2. First, $f(p_1, q; s)$ is evaluated directly, using (1), and then all other values are obtained using (2). We can charge off the work involved in the summation to the points r, each of which is considered at most once during the angular sweep about q. The result is that, for the fixed choice of s, the values $f(p, q; s)$ can be tabulated in total time $O(n^2)$. (The $O(n)$ angular sorts of points about each choice of q can be done, using the standard method of computing the arrangement of dual lines, in total time $O(n^2)$; see [3].)

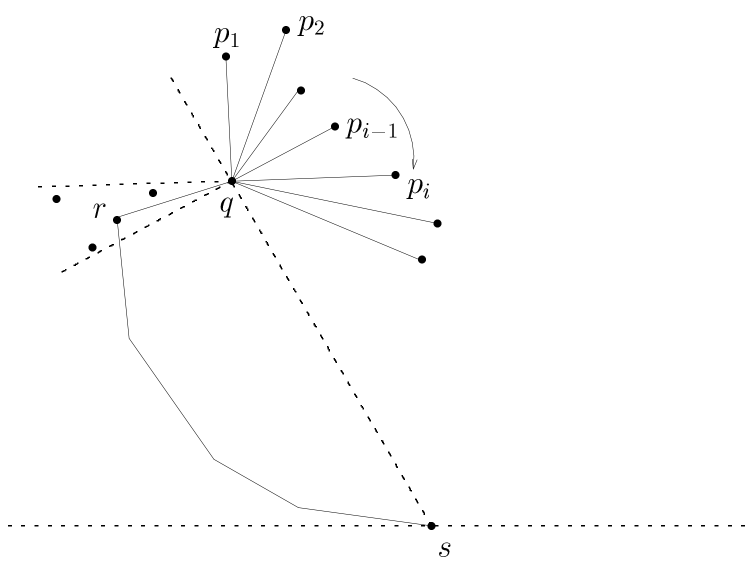

Fig. 2. Notation used in recursion (2).

The total number of convex polygons determined by S is obtained simply by summing:

$$\sum_{s \in S} \sum_{p \in U_s} \sum_{q \in U_s \cap H_{s,p}} f(p, q; s).$$

3 Counting convex k-gons

Now consider the problem in which we count only convex k-gons, for a given integer k. (In the process of counting k-gons, we will in fact also count convex j-gons, for $j \leq k$.) Let p and q be points of U_s, such that $q \in H_{s,p}$. We define $g(p, q, j; s)$ to be the number of convex polygons which fulfill the conditions (a)–(c), as in the previous section, *and* have exactly j edges. Then the number $g(p, q, j; s)$ is obtained as follows:

$$g(p, q, j; s) = \begin{cases} 1, & \text{if } j = 3; \\ \displaystyle\sum_{r \in U_s \cap H_{p,q} \cap H_{s,q}} g(q, r, j-1; s), & \text{otherwise;} \end{cases} \tag{3}$$

and the total number of convex k-gons is given by summing:

$$\sum_{s \in S} \sum_{p \in U_s} \sum_{q \in U_s \cap H_{s,p}} g(p, q, k-1; s).$$

As in (2), we can improve the efficiency of evaluating the recursion in (3) incrementally, obtaining $g(p_i, q, j; s)$ from $g(p_{i-1}, q, j; s)$, plus a sum (over r) of $g(q, r, j-1; s)$, for points $p_1, p_2, \ldots$ sorted in clockwise order about q.

4 Counting empty convex polygons

If we restrict attention to *empty* convex polygons we only have to rewrite the recursions (1) and (3), adding the restriction that triangle Δpqs have no points of S in its interior. For example, the number of j-edge convex polygons (with lowest point s) whose convex hull encloses no points of S is given by the following recursion:

$$G(p, q, j; s) = \begin{cases} 0, & \text{if } \Delta pqs \cap S \neq \emptyset; \\ 1, & \text{if } \Delta pqs \cap S = \emptyset \text{ and } j = 3; \\ \displaystyle\sum_{r \in U_s \cap H_{p,q} \cap H_{s,q}} G(q, r, j-1; s), & \text{otherwise.} \end{cases}$$

Note that the test to see if Δpqs is empty can be done in constant time: As shown in Theorem 2.1 of [5] or in Theorem 1 of [1], we can preprocess the set S in time $O(n^2)$ into a data structure of size $O(n^2)$, so that for any query consisting of three points of S, we can check in constant time if the triangle determined by the three points is empty or not. (Actually, the theorem allows us to sum the "weights" of the contained points; but we do not need this generality.)

5 Counting convex polygons containing a given point

Another variation is the problem of computing the number of convex k-gons containing a given point u (not necessarily from the set S) in the interior. The case $k = 3$ was considered in [6], where an $O(n^2)$ algorithm was given for computing, for each $u \in S$, the number of triangles containing u.

For each of the four versions of our problem given above (all convex polygons vs. convex k-gons; arbitrary convex polygons vs. empty convex polygons), we can solve the variant in which we require point u to be inside the polygons that are counted. We again fix attention on counting those polygons that have point $s \in S$ as their lowest vertex. If $u \notin U_s$ the answer is 0. Otherwise, we can compute, for example, the number $g'(p, q, j; s)$ of j-edge convex polygons which fulfill the conditions (a)–(c), as before, *and* contain point u, by solving the following recursion:

$$g'(p,q,j;s) = \begin{cases} 0, & \text{if } u \notin H_{s,p} \cap U_s; \\ 1, & \text{if } u \in \Delta pqs \text{ and } j = 3; \\ \displaystyle\sum_{r \in U_s \cap H_{p,q} \cap H_{s,q}} g'(q,r,j-1;s), & \text{if } u \notin \Delta pqs; \\ \displaystyle\sum_{r \in U_s \cap H_{p,q} \cap H_{s,q}} g(q,r,j-1;s), & \text{if } u \in \Delta pqs; \end{cases}$$

where $g(p, q, j; s)$ has been already computed by equations (3).

6 Summary

For the space complexity, note that we never have need to store more than $O(n^2)$ or $O(kn^2)$ numbers (for the cases of counting all polygons, or all k-gons, respectively). The time complexity is obtained simply by multiplying $O(n)$

(the number of choices for s) by the number of entries ($O(n^2)$ or $O(kn^2)$) that we need to store for each choice of s.

Theorem 1 *Given a set S of n points in the plane, in time $O(n^3)$ (resp., $O(mn^3)$) and space $O(n^2)$ (resp., $O(mn^2)$), the total number of convex polygons (resp., convex k-gons, for all $k \leq m$) whose vertices are a subset of S can be computed. Also, in the same time and space complexities, the total number of empty convex polygons (empty convex k-gons) can be computed, or the total number of convex polygons (convex k-gons) containing a given point can be computed.*

Note: We can extend the above recursions to compute the number of t-subsets $T \subseteq S$ whose convex hull is a k-gon, in time $O(k(t-k+1) \cdot n^4)$. (This number is denoted by $\#(t, k)$ in [8,9].) For $t = k$ this reduces to the part of the above theorem.

It is straightforward to decrease the storage requirement to $O(n)$, at the expense of a factor of n in the running time.

Acknowledgement

We thank the referees for observing the improvement by a factor of n in the running time for the evaluation of the recursions.

References

[1] E.M. Arkin, S. Khuller, and J.S.B. Mitchell, Geometric knapsack problems, *Algorithmica* **10** (1993), 399–427.

[2] V. Chvátal and G. Klincsek, Finding largest convex subsets, in: *Proc. 11th Southeastern Conf. Combinatorics, Graph Theory, and Computing*, (Boca Raton, Florida, 1980), Vol. II; *Congressus Numerantium* **29** (1980), 453–460.

[3] H. Edelsbrunner and L. Guibas, Topologically sweeping an arrangement, *Journal of Computer and System Sciences* **38** (1989), 165–194.

[4] D. Eppstein, New algorithms for minimum area k-gons, in: *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms (SODA)*, 1992, pp. 83–88.

[5] D. Eppstein, M. Overmars, G. Rote, and G. Woeginger, Finding minimum area k-gons, *Discrete & Computational Geometry* **7** (1992), 45–58.

[6] S. Khuller and J.S.B. Mitchell, On a triangle counting problem, *Information Processing Letters* **33** (1990), 319–321.

[7] B. Korte, L. Lovász, and R. Schrader, *Greedoids*, in the series *Algorithms and Combinatorics*, Vol. 4, Springer-Verlag, 1991.

[8] G. Rote, Z. Wang, G. Woeginger, and B. Zhu, Counting k-subsets and convex k-gons in the plane, *Information Processing Letters* **38** (1991), 149–151.

[9] G. Rote and G. Woeginger, Counting convex k-gons in planar point sets, *Information Processing Letters* **41** (1992), 191–194.