Triangle Guarding*

Jocelyn Smith and William Evans
Department of Computer Science, University of British Columbia

Abstract We consider the problem of triangle guarding ($\triangle$-guarding) a simple, 2D polygon Q. A polygon Q is $\triangle$-*guarded* if every point q of Q is contained in the convex hull of some three guards that can all see q. This rather odd condition approximates a desire, for example, to see all sides of q or to locate (via triangulation) q from at least two angularly well-separated views. If we restrict the guards to lie in Q and Q has a transparent boundary then the vertices of the convex hull of Q ($CH(Q)$) form the minimum set of $\triangle$-guards. We examine two other variations of the problem. In the first, the guards may be placed within a transparent border surrounding Q. For this variation, we describe a polynomial time algorithm, related to the classic algorithm for finding min-link separators, that finds a minimum set of $\triangle$-guards. In the second, we consider the case where Q's boundary is opaque and show that, as in the traditional art gallery problem, finding a minimum set of vertex $\triangle$-guards is NP-hard.

Transparent Fences

Given simple polygons P and Q with Q contained in P, where $|Q| + |P| = n$, what is the minimum number and placement of guards in P needed to $\triangle$-guard Q, assuming that both Q and P have transparent boundaries?

Our algorithm for this problem is inspired by an algorithm for finding min-link separators between two convex polygons, one contained in the other [1]. The following basic lemmas motivate using this approach. Define *legal regions* to be regions of the outer polygon P, that are not contained in the interior of $CH(Q)$ (Figure 1).

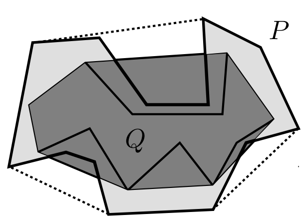

Figure 1: Legal regions. Q is the innermost polygon with $CH(Q)$ in dark gray. The dotted lines complete $CH(P)$. The legal regions are light gray.

*This research was supported in part by an NSERC Reseach Grant.

Lemma 1 *Guards in any minimal guarding set are in legal regions.*

Proof. The guards must be in P. Since every point of Q must be guarded, Q and $CH(Q)$ are in CH(guarding set). Since the guarding set is minimal, the guards are in convex position and thus lie outside the interior of $CH(Q)$. $\square$

A *legal separator* is a polygon that contains $CH(Q)$ and whose vertices are in legal regions. (Note: A legal separator separates $CH(Q)$ from $CH(P)$.) A *minimal legal separator* has the fewest number of edges of any legal separator.

Lemma 2 *Polygon K is a minimal legal separator if and only if its vertices form a minimal $\triangle$-guarding set.*

Proof. (Only if) By definition, K contains Q and thus its vertices form a $\triangle$-guarding set.

(If) If K's vertices form a $\triangle$-guarding set then they lie in P and K contains Q. Since K is minimal, it is convex and thus contains $CH(Q)$. $\square$

By Lemma 2, we can find a minimal $\triangle$-guarding set by finding a minimal legal separator.

The *Q-contact* of a tangent ℓ to Q is the vertex of Q that ℓ is tangent to.

A *left tangent* to Q is a directed tangent to Q that has Q to its left, when facing in the tangent's direction.

Given polygons P and Q with $Q \subseteq P$ and a left tangent ℓ to Q, let R_ℓ be the part of P that is on or to the right of ℓ. If e is an edge of $CH(Q)$, we write R_e to mean R_ℓ where ℓ is the left tangent coincident to e.

Lemma 3 (Lemma 2 [1]) *For every left tangent ℓ to Q, R_ℓ contains at least one vertex of any separator.*

Proof. Suppose separator T has no vertex in R_ℓ, then the Q-contact(s) of ℓ cannot be in a triangle formed by vertices of T. $\square$

The *extreme point* for ℓ is the point $p \in R_\ell$ whose left tangent to Q has the largest CCW angle with ℓ. See Figure 2.

The *extreme polygon* T_a for a given point $a \notin CH(Q)$ is obtained as follows: Let v_1 be the point a. In general, to obtain the vertex v_{i+1}, draw a left tangent ℓ to Q from the point v_i. Choose v_{i+1} to be the extreme point for

ℓ. Stop when the line from v_{i+1} to a doesn't intersect Q. Let $v_1, v_2, \ldots, v_k$ be the vertices constructed by this process. The extreme polygon T_a is $\langle a = v_1, v_2, \ldots, v_k\rangle$. Note that T_a may not be convex (for example, see Figure 2), however, T_a has at most one more vertex than a minimal legal separator (Lemma 3 [1]).

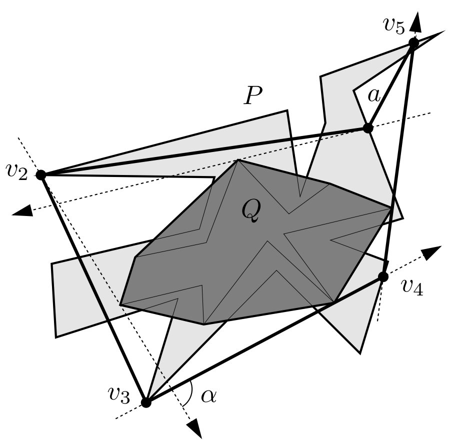

Figure 2: The construction of an extreme polygon $T_a = \langle a = v_1, v_2, v_3, v_4, v_5\rangle$. The point v_3 is chosen to maximize angle α.

For edge e of $CH(Q)$, let B_e be the set of points $p \in R_e$ that have a left tangent ℓ to Q such that every other point of R_e on the tangent lies between p and ℓ's Q-contact.

Lemma 4 *For any edge e of $CH(Q)$, there exists a minimal legal separator that is an extreme polygon T_a for some $a \in B_e$.*

Proof. (sketch) Suppose $T = \langle w_1, w_2, \ldots, w_{k^*}\rangle$ is a minimal legal separator that is not extreme. We first transform T into a minimal legal separator T' that has a vertex in B_e. We know (Lemma 3) that T must contain a vertex in R_e, which we assume without loss of generality is w_1. Let ℓ be the left tangent from w_1 to Q. Let a be the point in B_e with left tangent ℓ. We claim that $T' = \langle a, w_2, \ldots, w_{k^*}\rangle$ is a minimal legal separator. To see this, first note that T is convex, otherwise $CH(T)$ is a smaller legal separator. The vertex a lies in the intersection of halfplanes formed by (lines coincident with) the two edges of T adjacent to w_1 because Q, and hence ℓ's Q-contact, is in T. Thus $T \subseteq T'$ and T' is a minimal legal separator.

Let $T_a = \langle a = v_1, v_2, \ldots, v_k\rangle$. Consider the sequence of polygons $A_1 = T', A_2, \ldots, A_{k^*}$ where each A_i (for $i > 1$) is the same as A_{i-1} except that vertex i in A_i is the ith vertex in T_a. We claim that, for all i, A_i is a minimal legal separator. Since $|A_i| = |T|$, it suffices to show $CH(Q) \subseteq A_i$. The proof is by induction on i, where we have just established the base case $CH(Q) \subseteq T \subseteq T' = A_1$. Assume $CH(Q) \subseteq A_{i-1}$ and thus A_{i-1} is a minimal legal separator and hence convex. To show $CH(Q) \subseteq A_i$, consider the left tangent ℓ from v_{i-1}. By Lemma 3 and the fact that A_{i-1} is convex, w_i must lie on or to the right of ℓ. Since v_i lies on or to the right of ℓ as well, it suffices to show that $\overline{v_i w_{i+1}}$ doesn't intersect $CH(Q)$. Since v_i is the extreme point for ℓ and $w_i \in R_\ell$, v_i lies on or to the right of the left tangent t from w_i to Q. Since A_{i-1} is convex and contains Q, v_{i+1} lies on or to the right of t. Thus $\overline{v_i w_{i+1}}$ doesn't intersect Q. When $i = k^*$, $w_{i+1} = a$ and the same argument shows that $\overline{v_{k^*} a}$ doesn't intersect Q, establishing $k = k^*$. $\square$

Finding a Minimal Extreme Polygon Lemma 4 implies that a minimal extreme polygon T_a over all $a \in B_e$ is a minimal legal separator, and thus (Lemma 2) its vertices form a minimal $\triangle$-guarding set. We search B_e (for a single, arbitrary edge e) for a point a that forms a minimal extreme polygon T_a in a manner similar to Aggarwal *et al.* [1].

At a high level, we partition B_e, which is part of the boundary of P, into a finite number of contiguous pieces so that the locations of the vertices $v_2, v_3, \ldots, v_k$ are related by simple functions to the position of a, as a varies within a single piece. We can then search for the smallest extreme polygon for each piece, and take the minimum over all pieces.

To define the pieces of B_e, we look for *breakpoints* $b \in B_e$ such that T_b has at least one vertex that is on a different edge of P or has a different Q-contact than the corresponding vertex of T_a, where $a \in B_e$ immediately precedes b[1]. To find these points, we create a search structure that given any left tangent ℓ reports ℓ's extreme point and how far we can rotate ℓ before its extreme point changes.

To build this structure, imagine rotating a left tangent ℓ around Q. Let S_ℓ be the set of vertices to ℓ's right unioned with the set of intersections between ℓ and the boundary of P. The extreme point for ℓ is the point in S_ℓ whose left tangent forms the largest CCW angle with ℓ. A vertex $p \in P$ has a fixed left tangent (as ℓ rotates) and is in S_ℓ for ℓ (angularly) between the left tangent to p from Q and the left tangent from p to Q.[2] The intersection of ℓ and an edge $\overline{pq}$ of P is in S_ℓ for ℓ between the left tangents to p and q.[3] By ordering the angular ranges during which these vertices and edge intersections are in S_ℓ, and tracking the maximum CCW angle between ℓ and the left tangents from these points as ℓ rotates, we obtain our search structure in $O(n \lg n)$ time.

[1]The angle of left tangents from points in B_e determines their order.

[2]A left tangent ℓ is *to* a point p if ℓ's Q-contact precedes p on ℓ in ℓ's direction.

[3]The intersection with $\overline{pq}$ for ℓ between left tangents *from* p and q is never extreme.

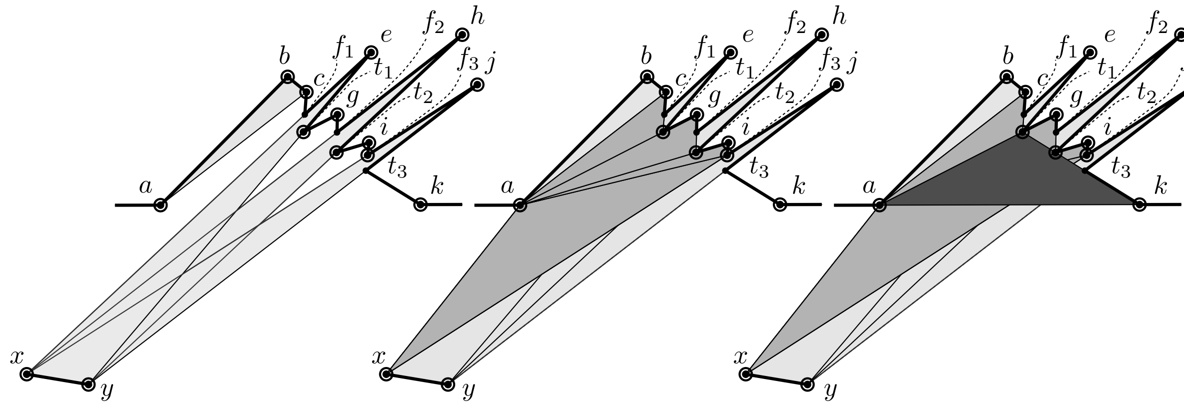

Figure 3: Three views of one clause junction. Guards b, c, e, g, j, i, j, x, and y are on convex vertices (property (1)). Guards a and k are are required by property (2) since no vertices outside the clause junction are colinear with $\overline{ab}$ or $\overline{t_3k}$. Together with x and y, these guards $\triangle$-guard most of the clause junction (shown in light gray). The entire clause junction can be guarded if and only if there is a guard on either f_i or t_i for each literal (forming the medium gray triangles) and at least one of the true vertices has a guard (for example, t_1, forming the dark gray triangle).

Using this structure, we can compute the next breakpoint in $O(k \log n)$ time. Solving a simple, rational function of a at each breakpoint enables us to find the minimal extreme polygon T_a within each contiguous piece of B_e (bounded by breakpoints) in constant time per piece. Notice that if any vertex v_i of T_a (except v_k) is an extreme vertex not on the left tangent from v_{i-1}, then v_k is not sensitive to changes in a until a crosses the next breakpoint.

Since there are $O(n)$ breakpoints, we can find the minimal extreme polygon in $O(nk \log n)$ time.

Simple Opaque Polygons

The difficulty of finding minimal $\triangle$-guarding sets changes dramatically when the polygon's boundary is opaque. We show:

Theorem 1 *Deciding if k vertex guards can $\triangle$-guard a simple polygon Q with an opaque boundary is NP-hard.*

Even without restricting guards to vertices, we know: (1) a guard is needed at every convex vertex of Q, and (2) every edge of Q must lie between two guards that are colinear to it.

Our proof is a reduction from 3SAT based on a similar reduction to traditional vertex guarding of a simple polygon [2, 3].

Literal Pattern For each occurrence of a literal in the 3SAT instance Φ, we create a literal pattern (Figure 4). Vertex a is the lower left vertex of the clause junction, described below, and is colinear to the edge $t'g$. The clause junction requires a guard at a, and vertex g requires a guard by property (1). To $\triangle$-guard the edge $\overline{gf}$, a guard is needed at either f or t. A guard at f corresponds to the literal having a value of false; a guard at t corresponds to the literal having a value of true. Guards at convex vertex h and the two convex vertices x and y (common to all literal patterns) $\triangle$-guard the spike fht.

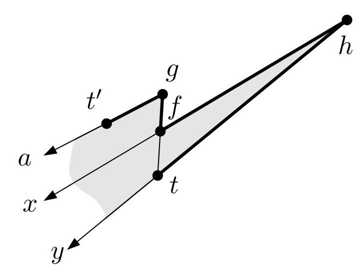

Figure 4: Literal Pattern.

Clause Junction The clause junction, see Figure 3, is constructed so that edge $\overline{t_3k}$ is $\triangle$-guarded if and only if at least one of the literals is true. Each clause junction includes a literal pattern for each literal in the clause. The vertices corresponding to a value of true for each literal are colinear to the lower right edge $\overline{t_3k}$ of the clause junction. The only other vertex colinear to this edge is its right end point k. To $\triangle$-guard this edge, a guard must be placed on k and on at least one of the true vertices. The clause junction can be guarded with 12 guards (+2 guards at x and y) if and only if each literal has exactly one truth setting (one guard on its f or t vertex) and at least one literal is true.

Variable Pattern The 3SAT instance Φ is only satisfied if the assignment to literals is consistent. To ensure consistency we introduce a variable pattern for each variable (Figure 5). Each variable pattern has two *wells*, one for *true* and one for *false*. We call the upper right vertex of a well a *variable consistency vertex*. Within each well are variable pockets, described below, one for each occurrence of the variable in Φ. If all occurrences are assigned the same truth value, only one additional guard is needed (for that variable) to guard all the pockets corresponding to the other truth value. A nook (at v_1) with edges colinear to the variable consistency vertices and q_1, ensures that at least one of these vertices has a guard. A nook (at v_2) between the two wells ensures the space between them is always guarded. The sides of the wells are colinear to the convex vertices q_1 and q_2 external to the variable pattern. Excluding the pocket guards, seven guards are needed to guard each variable pattern.

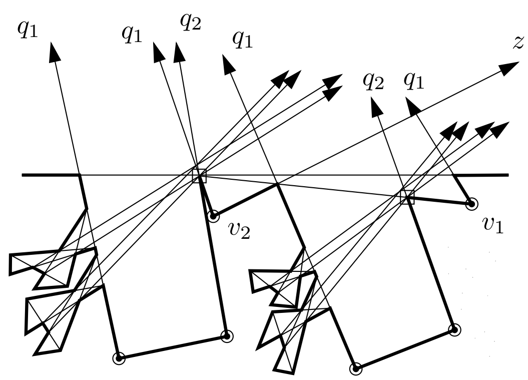

Figure 5: A variable pattern. Guards must always be placed at the circled vertices. A guard must be placed at one of the two variable consistency vertices in squares.

Variable Pocket There are two variable pockets for each occurrence of the variable in Φ: one in the variable's *true* well and the other in its *false* well. The variable pocket is constructed so part of it can be $\triangle$-guarded only if the corresponding t/f vertex in the clause junction has a guard, or the corresponding variable consistency vertex has a guard, or an additional guard is added inside the pocket (see Figure 6). Each variable pocket has four convex vertices. The two vertices l and o at the top of each pocket also require guards to guard the top edges ($\overline{lm_1}$ and $\overline{om_2}$) of the pocket. If the variable is *true* or *false*, a guard is placed at the variable consistency vertex corresponding to this value. This guard (already counted in the variable pattern) completes the triangle guarding of all the pockets in its well. By construction, the unguarded section of each pocket in the other well can be $\triangle$-guarded by the six guards in the pocket and the assigned t/f guard of the corresponding literal pattern. Thus all pockets are guarded without additional guards.

Figure 7 shows the entire construction for an example of two clauses and three variables. By this construction, we have shown:

Lemma 5 *A given 3SAT instance Φ is satisfiable if and only if the corresponding polygon Q can be $\triangle$-guarded by $48\times$(Number of Clauses) $+ 7\times$(Number of Variables) $+ 5$ vertex guards.*

Acknowledgments

We would like to thank Joe Mitchell and Alon Efrat for valuable discussions.

References

[1] A. Aggarwal, H. Booth, J. ORourke, and S. Suri. Finding minimal convex nested polygons. *Information and Computation*, 83:98–110, 1989.

[2] D. T. Lee and A. K. Lin. Computationaly complexity of art gallery problems. *IEEE transactions*, pages 276–282, 1986b.

[3] J. O'Rourke. *Art Gallery Theorems and Algorithms*. International Series of Monographs on Computer Science. Oxford University Press, 1987.

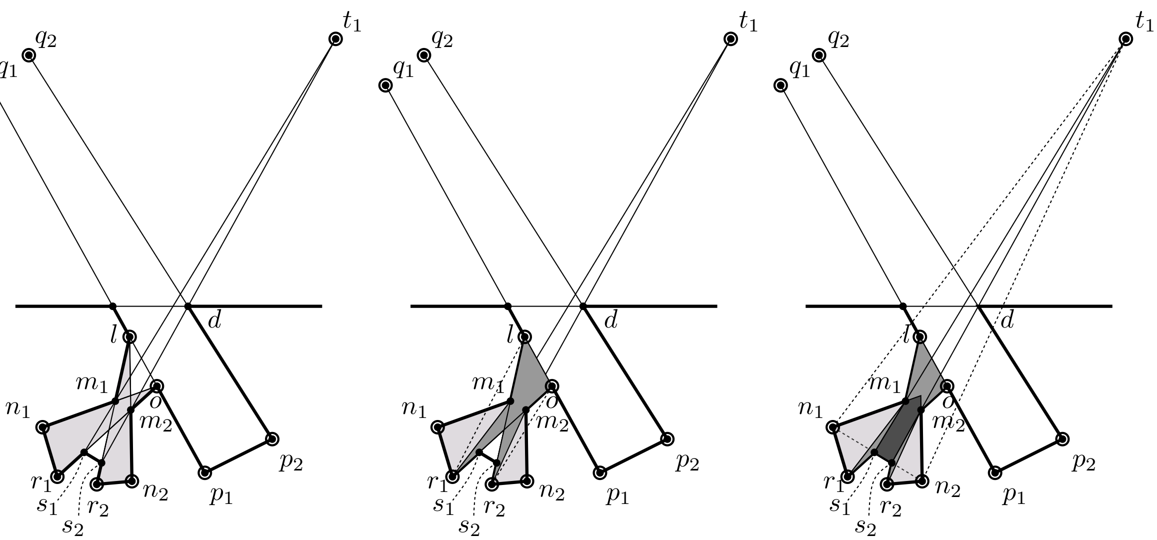

Figure 6: A variable pocket.

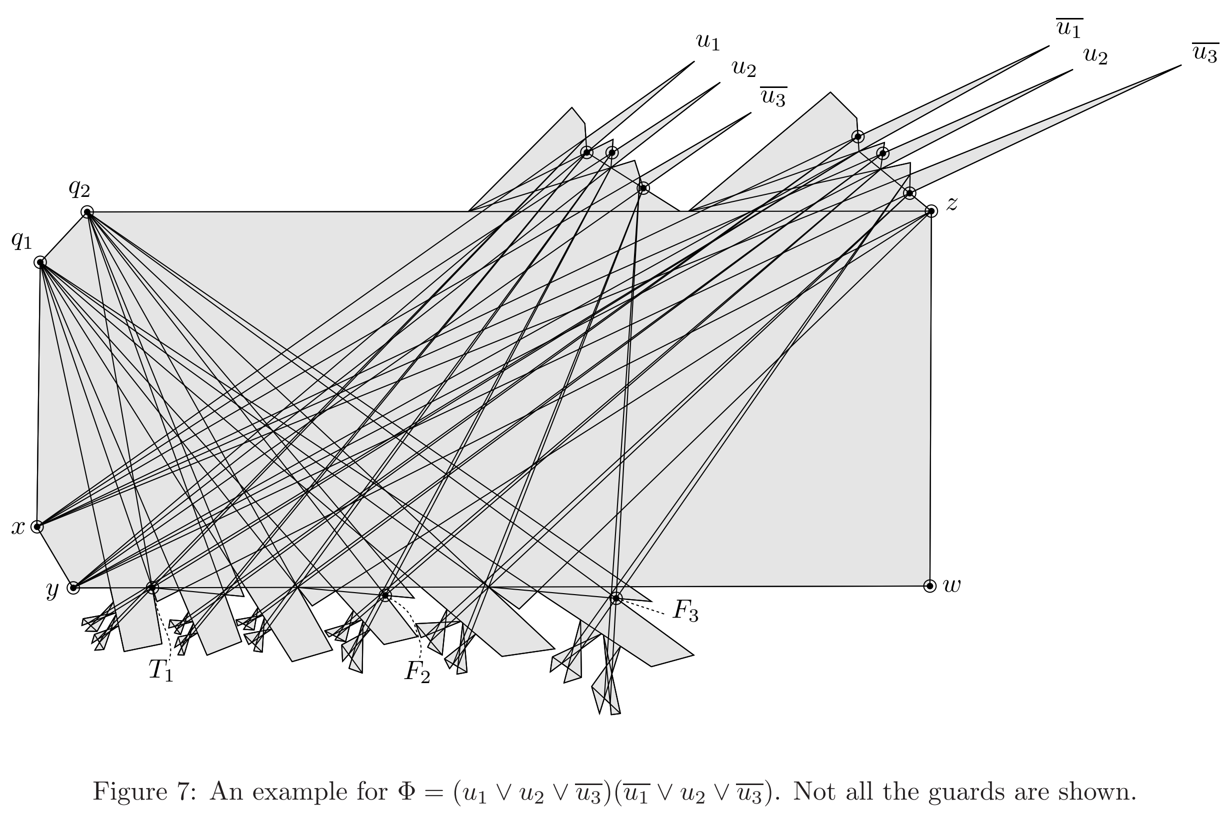

Figure 7: An example for $\Phi = (u_1 \vee u_2 \vee \overline{u_3})(\overline{u_1} \vee u_2 \vee \overline{u_3})$. Not all the guards are shown.