Computing Constrained Shortest Segments: Butterfly Wingspans in Logarithmic Time*

David Kirkpatrick Jack Snoeyink
Department of Computer Science
University of British Columbia

Abstract

We give a logarithmic-time algorithm to compute the shortest segment joining two convex n-gons A and B while avoiding another convex n-gon C. Our algorithm uses a tentative prune-and-search technique on standard representations of the polygons as arrays or balanced binary search trees.

1 Introduction

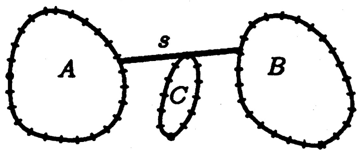

Figure 1: A shortest segment

This paper gives an optimal, logarithmic-time solution for a problem that Bhatacharya, Egyed, and Toussaint [3] called "computing the wingspan of a butterfly." We state the problem as a constrained search for a shortest segment: given three disjoint convex polygons, A, B, and C with n vertices apiece, compute the shortest segment s that joins A to B and does not intersect the interior of C, as shown in figure 1.

We assume that the polygons are represented as ordered lists of the vertices on their boundaries and that one can retrieve a "middle" vertex and one of its tangents from any list. This is certainly the case if polygon vertices are stored in an indexed array or in a balanced binary search tree that is threaded for successors to allow computation of the line segment joining adjacent vertices. With such representations of polygons A and B, the following can be computed in $O(\log n)$ time [9]:

- The shortest segment joining A to a point,
- Tangents to A through a point,
- Intersection of A with a line,
- Tangents to A with a given direction,
- Inner and outer common tangents to A and B [6, 8], and
- The shortest segment joining A and B [4, 5].

The first four can be computed by binary search; the last two by prune-and-search techniques, which repeatedly apply a constant-time test to local information to discard half of one of the polygons.

Bhattacharya et al. [3] defined a *butterfly polygon* to be an n-gon A with exactly four convex vertices $a_1, \ldots, a_4$ that lie on the boundary of the convex hull of A in the given order. The *wingspan* of A is the length of the shortest chord inside P that joins the concave chain between a_1 and a_2 to the concave chain between a_3 and a_4. Bhattacharya et al. gave an $O(\log^2 n)$ algorithm for the problem of *computing the wingspan of a butterfly*, which they used as a subproblem to compute shortest transversals of line segments [1, 2]. The problem of computing a constrained shortest segment is essentially equivalent: one problem can be reduced to the other in logarithmic time by computing common tangents.

In the next section we characterize the shortest segment $\overline{ab}$ that joins A and B and avoids the interior of C. In section 3 we give an optimal

*Both authors supported in part by NSERC Research Grants.

logarithmic-time algorithm to compute $\overline{ab}$ avoiding C by using ***tentative prune-and-search***, in which we may make tentative discards that are later certified or revoked [7].

2 Characterizing the shortest segment

Given disjoint convex n-gons A, B, and C, we want to compute $\overline{ab}$, the shortest segment joining A to B that does not intersect the interior of C. We assume, without loss of generality, that A is to the left and B is to the right of a vertical separating line. We further assume that the polygon vertices are in general position; e.g., that there are no pairs of parallel edges or collinear vertices. This makes the segment $\overline{ab}$ unique.

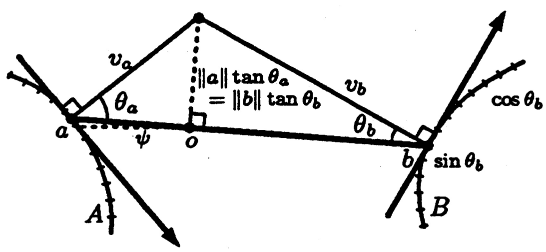

Figure 2: Shortest segment through o

For a particular choice of normals v_a and v_b, which are perpendicular to tangents at $a \in A$ and $b \in B$, we measure the angles θ_a counterclockwise (ccw) from $\overline{ab}$ to v_a and θ_b clockwise (cw) from $\overline{ab}$ to v_b. If no subsegment of $\overline{ab}$ joins A to B, then these angles are in the range $[-\pi, \pi]$. We define the ***balance line*** ℓ to be the line through $v_a \cap v_b$ and perpendicular to $\overline{ab}$ as shown in figure 2. The ***balance point*** is the intersection point $\overline{ab} \cap \ell$, if it exists. Note that if a or b is a vertex, there may be many balance points. We can characterize the shortest segment $\overline{ab}$ in terms of a balance point.

Theorem 1 *The shortest segment $\overline{ab}$ that joins A to B and avoiding C either*

- *is normal to A at a, to B at b, and does not intersect C, or*
- *is tangent to C and has normals v_a and v_b such that $\overline{ab}$ separates C and the normals, and the balance point is a point of tangency between $\overline{ab}$ and C.*

Proof: The first possibility is that the shortest segment $\overline{ab}$ that joins A to B ignoring C. If $\overline{ab}$ is not normal to both A and B then $\overline{ab}$ is not a local minima. There is only one globally shortest segment by convexity of A and B.

If C blocks this first possibility, then we must consider the second. If $\overline{ab}$ does not touch C, then $\overline{ab}$ can be shortened by moving a and b towards the endpoints of the blocked shortest segment. Therefore we can assume that $\overline{ab}$ is tangent to C.

Let the origin o be the leftmost point of tangency with C, and let v_a and v_b be the most-cw normals at $a \in A$ and $b \in B$. If we rotate $\overline{ab}$ ccw by angle ψ, then $\overline{ab}$ pivots on o and sweeps polygon edges with normals v_a and v_b. The change in length of the segment $\overline{oa}$ is

$$\frac{d}{d\psi}\|a\|\frac{\cos\theta_a}{\cos(\theta_a - \psi)} = -\|a\| \tan\theta_a.$$

Similarly, the change in $\overline{ob}$ is $\|b\| \tan\theta_b$. Therefore, a ccw rotation does not decrease the length of $\overline{ab}$ if and only if $\|a\| \tan\theta_a \leq \|b\| \tan\theta_b$.

If we choose the origin o as the rightmost point of tangency and choose the most ccw normals v_a and v_b, then we derive the opposite inequality for cw rotation of $\overline{ab}$. Taken together, these two inequalities imply that the length of $\overline{ab}$ is a local minimum if there exist normals v_a and v_b and an origin on $\overline{ab}$ and C such that $\|a\| \tan\theta_a = \|b\| \tan\theta_b$. This ***balance equation*** can hold only when θ_a and θ_b are both positive. In this range, both $\tan\theta_a$ and the length of the segment from a to C are monotone decreasing functions of the slope of $\overline{ab}$. Because $\|b\| \tan\theta_b$ is monotone increasing, any local minimum is the unique global minimum.

Geometrically, $\|a\| \tan\theta_a$ is the length of a segment $\overline{op}$ perpendicular to $\overline{ab}$ at o and intersecting the normal v_a at p. For the balance equation to hold, the point p must be the intersection of v_a and v_b. The balance point, which is the orthogonal projection of p onto $\overline{ab}$, must be a point of tangency with C. ■

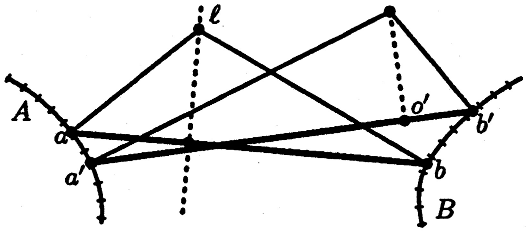

Figure 3: $\overline{a'b'}$ is ccw of $\overline{ab}$

In the rest of this paper, we assume that the normals v_a and v_b are above $\overline{ab}$. Figure 3 illustrates a corollary to theorem 1. If segments $\overline{ab}$ and $\overline{a'b'}$ both join A to B with a' below $\overline{ab}$ and b' above $\overline{ab}$, then we say that $\overline{a'b'}$ *is ccw of* $\overline{ab}$.

Corollary 2 *If $\overline{a'b'}$ is ccw of $\overline{ab}$, then any existing balance point of $\overline{a'b'}$ is right of the balance line ℓ of $\overline{ab}$.*

Proof: For the balance point o' to exist, normals $v_{a'}$ and $v_{b'}$ at a' and b' must make positive angles with $\overline{a'b'}$. Their intersection point $v_{a'} \cap v_{b'}$ must therefore be below v_a and above v_b—which places it to the right of ℓ and the projection of this intersection onto $\overline{a'b'}$ moves only further right of ℓ. ∎

3 An optimal algorithm using tentative prune-and-search

It is well known that the shortest segment $\overline{ab}$ joining A and B (ignoring C) can be computed in logarithmic-time by a prune-and-search algorithm. We sketch the details as an illustration of the technique of repeatedly throwing away half of something.

Theorem 3 (Edelsbrunner [5])
The shortest segment s joining two disjoint convex n-gons A and B can be computed in $O(\log n)$ steps.

Proof: Since we have assumed a vertical separating line, we can begin by computing outer common tangents in logarithmic time and clipping A and B to the portions that can be joined by segments. (One can do without the separating line by complicating the algorithm below.)

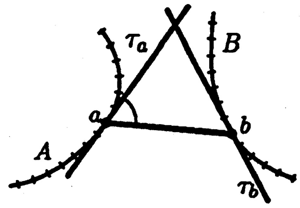

Figure 4: Discard from A

We want to find $\overline{ab}$ normal to A and B. Consider vertices $a \in A$ and $b \in B$ and look at their tangents τ_a and τ_b. Suppose that τ_a and τ_b separate $\overline{ab}$ from the polygons and, without loss of generality, that they form a triangle above $\overline{ab}$ in which the angle at a is acute, as in figure 4. Then all vertices below $\overline{ab}$ on A can be discarded: Any vertex $a' \in A$ has its normals below $\overline{ab}$, but the vertices of B with parallel normals are above $\overline{ab}$.

If a and b are always chosen as midpoints of their respective polygons, then $O(\log n)$ discards reduce one of the polygons to a single edge. We can complete the solution by computing a tangent parallel to this edge and the shortest segments from the edge's endpoints. ∎

We can check in logarithmic time whether the segment s found by this algorithm intersects C. If it does not, then s is the shortest segment joining A to B that avoids C. If it does, then we clip A and B at the endpoints of s and use inner common tangents with C to further clip A, B and C to portions that could be in contact with $\overline{ab}$ and have positive angles with the normals. The angle condition follows from the fact that both tangents must be positive for the balance equation of theorem 1 to hold. Notice that the portions remaining of A and B are monotone with respect to all tangents to the remainder of C.

We now use the tentative prune-and-search technique [7] to reduce one of the polygons to a single segment. Our *normal mode* of operation is like standard prune-and-search: we look at three middle points and their normals/tangents in constant time and try to discard half of one of the polygons A, B, or C. In one configuration we will not have enough information to determine what must be discarded; it is possible, however, to tentatively discard half of *each* polygon with the assurance that one of the discards is correct. We do so and switch into *tentative mode*.

In tentative mode we introduce midpoints on one polygon at a time in round-robin fashion and do one of two things. We either perform a tentative discard from the polygon under consideration or certify that all of the tentative discards that have been performed to one of the polygons have been correct, revoke all other tentative discards, and return to normal mode.

Lemmas 5 and 6 prove that the normal and tentative mode computations described above can be correctly implemented in constant time. Anticipating these lemmas, we use a potential function argument to establish the running time of the algorithm.

Theorem 4 *The shortest segment $\overline{ab}$ that joins A to B and avoids C can be computed in $O(\log n)$ time, given standard representations of the disjoint convex n-gons A, B and C.*

Proof: The state of a polygon A can be described by A_R, the number of segments remaining, and A_T, the number tentatively discarded. The potential of A is defined as $\Phi_A = 2\log A_R + 4\log(A_R + A_T)$. The global potential is the sum of the polygon potentials, plus five in tentative mode:

$$\Phi = \Phi_A + \Phi_B + \Phi_C + 5(A_T + B_T + C_T > 0).$$

Clearly, $\Phi \geq 0$ at all times and initially $\Phi = O(\log n)$. We show that Φ decreases by a constant at each step.

In three of the steps the decrease is easily seen: In normal mode, a step that stays in normal mode discards half of some polygon and decreases Φ by 6. In a transition to tentative mode, the net decrease in polygon potentials is 6, and Φ decreases by 1. In tentative mode, a step that stays in tentative mode tentatively discards half of the current polygon and decreases Φ by 2.

In a transition from tentative to normal mode that certifies the discards to polygon A, the change in Φ_A is $4(\log A_R - \log(A_R + A_T))$, which is (-4) times the number of tentative discards applied to A. The change in Φ_B is $2(\log(B_R + B_T) - \log B_R)$ and the change in Φ_C is similar. Since B and C have been subject to at most one more tentative step each than A, the change in Φ is at most $4 - 5 = -1$.

Since each step is implemented in constant time in lemmas 5 and 6, after $O(\log n)$ steps we have reduced one of the polygons to a single segment. Lemma 7 argues that we can then complete the computation in logarithmic time. ∎

If we have introduced $a \in A$, $b \in B$, and $c \in C$, then we denote the upper portions of A and B by $a\Uparrow$ and $b\Uparrow$, the lower portions by $a\Downarrow$ and $b\Downarrow$ and the left and right portions of C by $c\Leftarrow$ and $c\Rightarrow$.

Lemma 5 *In normal mode, we either discard half of a polygon or tentatively discard half of every polygon and enter tentative mode.*

Proof: We assume, by reflection if necessary, that we are given a configuration with the middle point $c \in C$ to the left of the balance line ℓ defined by $\overline{ab}$. We consider the tree of five normal cases in figure 5, based on the location of c and its tangent τ_c.

We justify the actions as follows. Cases N1–N4: Segments starting at or tangent to discarded points cross from below $\overline{ab}$ to above, touch C left of line ℓ but have their balance points right of ℓ according to corollary 2. Therefore, by theorem 1 they cannot participate in the shortest constrained segment. Case N5: This is the case that causes entry to tentative mode. The shortest segment $\overline{a'b'}$ can touch at most one of the three tentatively discarded polygons—e.g., if $\overline{a'b'}$ touches $a\Downarrow$ then it is clearly tangent to $c\Leftarrow$ and, in order for the balance point to be on C, it touches $b\Downarrow$. ∎

Once we enter tentative mode then we continue according to the next lemma.

Lemma 6 *In tentative mode, we either discard or tentatively discard half of the polygon under consideration or we certify all tentative discards made to one of the polygons and return to normal mode.*

Proof: We consider refining each of the polygons in turn: introducing c' as a middle point in the remaining portion of C, introducing a' in A, and finally introducing b' in B. Remember that c is left of the balance line ℓ.

Introducing c' in the undiscarded portion of C gives five cases of figure 6, which have the same justification as the five normal cases in figure 5.

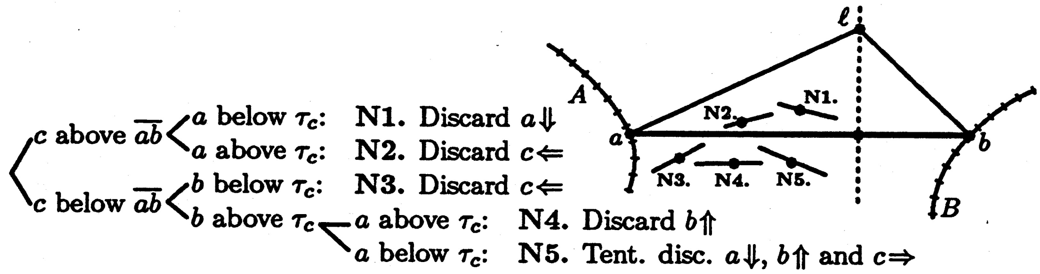

Figure 5: The five cases for $c \in C$ and tangent τ_c in normal mode

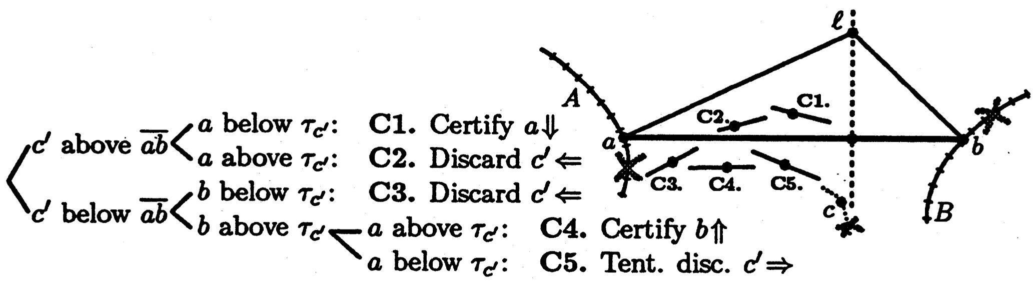

Figure 6: Five cases for $c' \in C$ and tangent $\tau_{c'}$ in tentative mode

In cases C2 and C3 we discard half of what remains of C. We call this a tentative discard to simplify the potential function argument, but in practice we can discard $c' \Leftarrow$ permanently. In cases C1 and C4 we discard a portion of A or B that has been tentatively discarded already. This is a *certifying* discard, which allows us to revoke other tentative discards and return to normal mode. In case C5 we are in essentially the same configuration as in N5—we can extend the tentative discard on C to c'.

Introducing a' in the undiscarded portion of A gives four cases shown in figure 7, which depend on the location of c with respect to $\overline{a'b}$ and the new balance line ℓ' for $\overline{a'b}$. Case A1 is the tentative discard configuration, so we extend the tentative discard to a'. Cases A2 and A3 are certifying discards to B and C. Case A4 is a normal discard to $a'\Uparrow$. Cases A3 and A4 are mirror images of normal cases in figure 5 because c is to the right of the balance line ℓ'.

Introducing b' gives the five cases of figure 8. In case B2 we extend the tentative discard to b'. Cases B1, B3 and B4 are certifying discards to A and C. Case B5 is a normal discard to $b'\Downarrow$. Cases B3–B5 are mirror images of normal cases because c is to the right of the balance line ℓ'. ∎

We need to argue that the computation can be completed in logarithmic time once one of the polygons has been reduced to a single segment.

Lemma 7 *If one of the polygons A, B or C is a single segment, then we can finish the computation of $\overline{ab}$ in $O(\log n)$ time.*

Proof: If C becomes a single segment s, compute where the extensions of s hit A and B and use theorem 1 to decide if the resulting segment $\overline{ab}$ is shortest or if the shortest $\overline{ab}$ is tangent to an endpoint of s. If the latter, then refine A and B and normal discard N1 or N4 applies.

If A or B becomes a single segment s, then one can compute the two segments $\overline{ab}$ from the endpoints of s, tangent to C, and touching the other polygon. If neither of these is the shortest segment, then refine the remaining polygons and choose on s the intersection with the tangent to

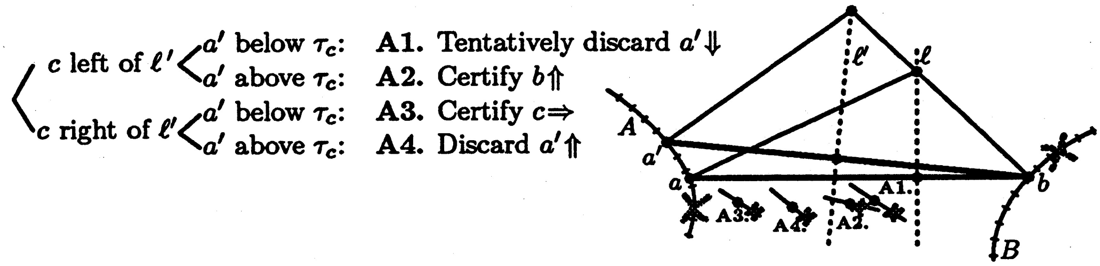

Figure 7: Four cases for a' in tentative mode. The balance line for $\overline{a'b}$ is ℓ'

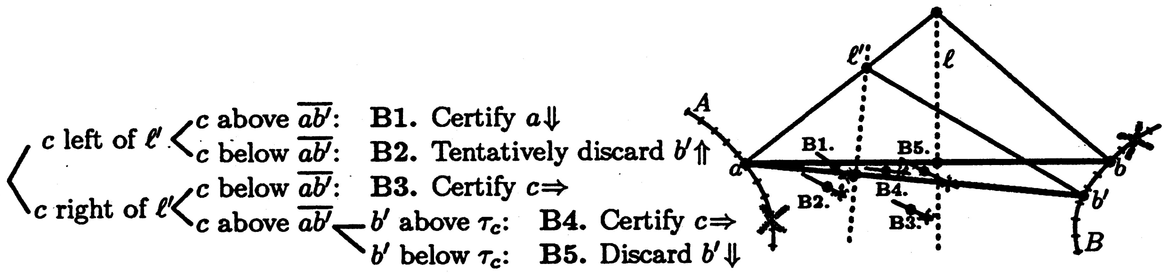

Figure 8: Five cases for b' in tentative mode. The balance line for $\overline{ab'}$ is ℓ'

the refined C. One of the normal discards N1–N4 applies.

When all the chains are single segments, the solution can be completed by simple algebra. ∎

Acknowledgement

We thank Lyudmil Aleksandrov for discussions on the characterization lemma.

References

[1] B. Bhattacharya and G. Toussaint. Computing shortest transversals. *Computing*, 46(3):93–119, 1991.

[2] Binay Bhattacharya, Jurek Czyzowicz, Peter Egyed, Ivan Stojmenovic, Godfried Toussaint, and Jorge Urrutia. Computing shortest transversals of sets. In *Proceedings of the Seventh Annual ACM Symposium on Computational Geometry*, pages 71–80, 1991.

[3] Binay Bhattacharya, Peter Egyed, and Godfried Toussaint. Computing the wingspan of a butterfly. In *Proceedings of the Third Canadian Conference on Computational Geometry*, pages 88–91, Simon Fraser University, Vancouver, 1991.

[4] David P. Dobkin and David G. Kirkpatrick. Fast detection of polyhedral intersection. *Theoretical Computer Science*, 27:241–253, 1983.

[5] H. Edelsbrunner. Computing the extreme distances between two convex polygons. *Journal of Algorithms*, 6:213–224, 1985.

[6] Leo Guibas, John Hershberger, and Jack Snoeyink. Compact interval trees: A data structure for convex hulls. *International Journal of Computational Geometry & Applications*, 1(1):1–22, 1991.

[7] David Kirkpatrick and Jack Snoeyink. Tentative prune-and-search for computing Voronoi vertices. In *Proceedings of the Ninth Annual ACM Symposium on Computational Geometry*, pages 133–142, 1993.

[8] M. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *Journal of Computer and System Sciences*, 23:166–204, 1981.

[9] Franco P. Preparata and Michael I. Shamos. *Computational Geometry—An Introduction*. Springer-Verlag, New York, 1985.