The Possible Hull of Imprecise Points*

William Evans† Jeff Sember†

Abstract

We pose the problem of constructing the possible hull of a set of n imprecise points: the union of convex hulls of all sets of n points, where each point is constrained to lie within a particular region of the plane. We give an optimal algorithm for the case when $n = 2$, and the regions are a point and a simple (possibly nonconvex) polygon. We then describe how the algorithm leads to an optimal algorithm for the case when $n \geq 2$, and each region is a simple polygon.[1]

1 Introduction

Let $S = \{s_1, \ldots, s_n\}$ be a planar point set.[2] If we are not given the locations of these points, but are told only that each point s_i lies within a particular region of uncertainty R_i, then the points are *imprecise*. The convex hull of a set of imprecise points cannot be determined since it is one of possibly many *feasible hulls* (each of which is the convex hull of a *feasible set* $\{s_1 \in R_1, \ldots, s_n \in R_n\}$).

Kreveld and Löffler investigate the problem of finding the feasible hull with maximal or minimal area or boundary length [3]. The problem of determining the intersection of all feasible hulls has also been investigated [5],[6],[2],[1],[7].

We define the *possible hull* of a set of imprecise points (or their corresponding regions of uncertainty) as being the union of the feasible hulls of the points. To motivate this problem, consider the scenario where each island in a group of islands contains a sensor whose exact location is uncertain, and that each pair of these sensors can detect any object that passes between them. To avoid being detected, a boat traveling near the islands would need to remain outside of their possible hull.

One reason that possible hulls have received little attention until now is that when the regions of uncertainty are convex, the possible hull is simply the convex hull of the regions [5]. In this paper, we investigate possible hulls of more general uncertain regions. We present an algorithm for constructing the possible hull of a point and a simple (possibly nonconvex) polygon, and describe how this algorithm can be used as a subroutine to construct the possible hull of two or more simple polygons. See Figure 1.

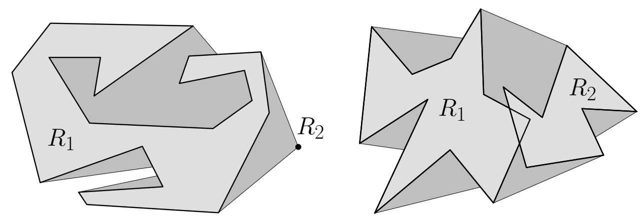

Figure 1: Possible hulls of pairs of uncertain regions.

2 Properties

We will denote the convex hull of a point set S by $\mathrm{CH}(S)$, and the possible hull of uncertain regions $\mathcal{R} = \{R_1, \ldots, R_n\}$ by $PH(\mathcal{R})$ (or, when clear from the context, by PH). Formally,

$$PH(\mathcal{R}) = \bigcup_{\{s_1 \in R_1, \ldots, s_n \in R_n\}} \mathrm{CH}(\{s_1, \ldots, s_n\}).$$

From this definition, we can derive the following additional properties of possible hulls.

Lemma 1 *$PH(\{A\}) = A$.*

Lemma 2 *$PH(\{A, B\}) = \bigcup_{a \in A, b \in B} \overline{ab}$.*

Lemma 3 *$\bigcup_{R_i \in \mathcal{R}} R_i \subseteq PH(\mathcal{R})$.*

Lemma 4 *If $\mathcal{A}$ and $\mathcal{B}$ are nonempty sets of uncertain regions, then $PH(\mathcal{A} \cup \mathcal{B}) = PH(\{PH(\mathcal{A}), PH(\mathcal{B})\})$.*

Proof. Let $Q = PH(\{PH(\mathcal{A}), PH(\mathcal{B})\})$. Suppose p is a point within $PH(\mathcal{A} \cup \mathcal{B})$. Then there exists a feasible set S of $\mathcal{A} \cup \mathcal{B}$ such that $p \in \mathrm{CH}(S)$. Let S_a and S_b be the subsets of S corresponding to the subsets $\mathcal{A}$ and $\mathcal{B}$. By using the definition of convex hull, it is easy to show that (i) $\mathrm{CH}(S) = \mathrm{CH}(\mathrm{CH}(S_a) \cup \mathrm{CH}(S_b))$, and (ii) there exist points a and b within $\mathrm{CH}(S_a) \cup \mathrm{CH}(S_b)$ such that $p \in \overline{ab}$. Now, without loss of generality, either (i) $a, b \in \mathrm{CH}(S_a)$, or (ii) $a \in \mathrm{CH}(S_a)$ and $b \in \mathrm{CH}(S_b)$. If (i),

*Research supported by NSERC and Institute for Computing, Information and Cognitive Systems (ICICS) at UBC

†Department of Computer Science, University of British Columbia, [will,jpsember]@cs.ubc.ca

[1]An applet demonstrating these results can be found at http://www.cs.ubc.ca/~jpsember/uh.html.

[2]All sets in this paper are assumed to be multisets.

then $\overline{ab} \subseteq \mathrm{CH}(S_a)$, and since (by definition) $\mathrm{CH}(S_a) \subseteq PH(\mathcal{A})$, $\overline{ab} \subseteq PH(\mathcal{A})$, which implies (by Lemma 3) that $\overline{ab} \subseteq Q$. If (ii), then since $\mathrm{CH}(S_a) \subseteq PH(\mathcal{A})$ and $\mathrm{CH}(S_b) \subseteq PH(\mathcal{B})$, Lemma 2 implies $\overline{ab} \in Q$. Hence $PH(\mathcal{A} \cup \mathcal{B}) \subseteq Q$.

If p is a point in Q, then by Lemma 2, $p \in \overline{ab}$, where $a \in PH(\mathcal{A})$ and $b \in PH(\mathcal{B})$. There must then exist feasible sets S_a of $\mathcal{A}$ and S_b of $\mathcal{B}$ where $a \in \mathrm{CH}(S_a)$ and $b \in \mathrm{CH}(S_b)$. Note that $\overline{ab}$ is within $\mathrm{CH}(S_a \cup S_b)$, and since $S_a \cup S_b$ is a feasible set of $\mathcal{A} \cup \mathcal{B}$, $\mathrm{CH}(S_a \cup S_b)$ is within $PH(\mathcal{A} \cup \mathcal{B})$; hence $Q \subseteq PH(\mathcal{A} \cup \mathcal{B})$. □

Lemma 5 *The possible hull of any set of two or more connected uncertain regions is simply connected.*

Proof. Let $\mathcal{R} = \{A, B\}$ be a set of connected uncertain regions (Lemma 4 implies that the proof extends by induction to sets of more than two regions). By Lemma 2, every point of PH is connected within PH to both A and B; and by Lemma 3, both A and B lie within PH. Hence PH is connected, and we need only show that it has no holes. We will do this by showing that every exterior point of PH is the source of a ray exterior to PH. Let q be any point exterior to PH. First, observe that if no lines through q that are tangent to A exist, then (i) every line through q will intersect A, and (ii) at least one line through q will intersect A to both sides of q. Since the same argument applies to B, if no such lines exist for A or B, then some segment $\overline{ab}$ exists (where $a \in A$ and $b \in B$) that contains q, implying that $q \in PH$, a contradiction. Hence, we can assume that there exist directed lines L_1 and L_2 through q that are right-tangent to (without loss of generality) A. Let W_1 (resp., W_2) be the wedge lying on or to the right (resp., left) of both L_1 and L_2. Note that W_1 contains A. Note also that W_2 cannot intersect B, otherwise some segment $\overline{ab}$ exists that contains q. Let R be any ray from q lying in W_2. Observe that no point $r \in R$ can lie on a segment $\overline{ab}$, since for any choice of $a \in A$, the portion of ray $\overrightarrow{ar}$ lying at or beyond r lies within W_2 (Figure 2). Hence, by Lemma 2, R does not intersect PH. □

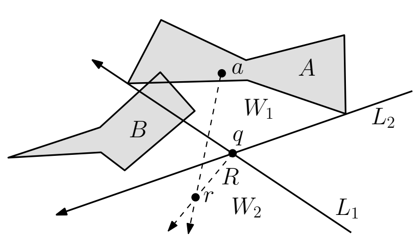

Figure 2: Lemma 5

Corollary 6 *Let $(p_1, \ldots, p_k, p_1)$ be a cyclic sequence of points. If each consecutive pair (p_i, p_{i+1}) are endpoints of a segment known to lie within PH, and P is a simple polygon whose edges lie on these segments, then the interior of P lies within PH.*

Theorem 7 *The possible hull of any set of two or more connected uncertain regions is star-shaped.*

Proof. Let $\mathcal{R} = \{A, B\}$ be a set of two connected uncertain regions (Lemma 4 can be applied to prove the claim for sets of more than two regions). We will prove that PH is star-shaped by showing that it has a nonempty kernel.

Let $I = \mathrm{CH}(A) \cap \mathrm{CH}(B)$. If $I \neq \emptyset$, then let s be any point in I. Observe that there must exist points $a_1, a_2 \in A$, and $b_1, b_2 \in B$ where s lies on both $\overline{a_1a_2}$ and $\overline{b_1b_2}$. Let q be any point in PH. By Lemma 2, there exist points $a' \in A$, $b' \in B$ such that $q \in \overline{a'b'}$. Without loss of generality we can assume that a_1, b_1, and a' are on or to the left of $\overrightarrow{sq}$, and that a_2, b_2, and b' are on or to the right of $\overrightarrow{sq}$. There are now two cases (Figure 3): s, b_1, and a_2 are either to the left or to the right of $\overrightarrow{a_1b_2}$. In both cases, we can construct a cyclic sequence of points defining a polygon (per Corollary 6) that lies within PH, and which contains $\overline{sq}$. (In the former case, the sequence is $(a_1, b_2, a_2, b', a', b_1)$; and in the latter case, it is (b_1, a_2, b', a').) Since this holds for any $q \in PH$, s is in the kernel of PH.

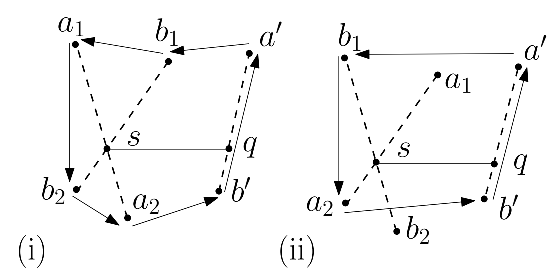

Figure 3: Theorem 7

If $I = \emptyset$, then there exists line L_1 right-tangent to A and left-tangent to B, and line L_2 left-tangent to A and right-tangent to B. Let s be the point where L_1 and L_2 cross, and q be any point in PH. By Lemma 2, $q \in \overline{ab}$, for some $a \in A$, $b \in B$. Note that there exist points $a' \in A$ and $b' \in B$ such that $s \in \overline{a'b}$ and $s \in \overline{ab'}$. We can again construct a sequence of points that defines a polygon that lies within PH and contains $\overline{sq}$. If s lies to the right of $\overline{ab}$, this sequence is (b, a, b', a'); otherwise, it is (a, b, a', b'). □

From this point on, we will assume that each region of $\mathcal{R}$ is a polygon (or a point, which can be viewed as a degenerate polygon). We will refer to an edge of each such polygon as a *native segment*, and to a segment connecting vertices of distinct polygons of $\mathcal{R}$ as a *bridge segment*.

Theorem 8 *If $\mathcal{R}$ is a set of uncertain polygons with a total of n vertices, then $PH(\mathcal{R})$ is a star-shaped polygon with at most n vertices.*

Proof sketch. PH is star-shaped, by Theorem 7. It can be shown (we omit the details) that every point on the boundary of PH lies on either a native segment or a bridge segment of $\mathcal{R}$; hence, PH is a polygon. To bound its complexity, it can also be shown that each boundary vertex v that is not already a vertex of $\mathcal{R}$ can be associated with a subset of the boundary of one of the polygons A of $\mathcal{R}$ that (i) is disjoint from the subset associated with any other vertex of PH, and (ii) contains a vertex of A (in the interior of PH) to which we can charge v. $\square$

3 Point and Polygon

By Lemma 5, the possible hull of a point s and a polygon P, $PH = PH(\{s, P\})$, is equal to the possible hull of s and the boundary of P. Hence, it will suffice for our algorithm to construct the possible hull of a point and a simple polygonal chain.

Our algorithm is reminiscent of Melkman's algorithm for finding the convex hull of a simple polygonal chain [4] for two reasons: first, both are on-line algorithms; and second, both look for points where the chain enters and emerges from the interior of the hull, and rely upon the simplicity of the chain to perform this efficiently.

We motivate our algorithm with the following observation: the possible hull of a point s and a chain P is equal to a union of triangles, where each triangle's vertices are s and the endpoints of an edge of P.

Let $(p_1, \ldots, p_n)$ be the ordered vertices of P. We will denote the connected subset of P from a to b by $(a \ldots b)$. We start with point u initialized to p_2, and the current possible hull H initialized to $PH(\{s, (p_1 \ldots p_2)\}) = \triangle sp_1p_2$. We advance u along P, processing each new edge (or part of an edge) in one of two ways. If the edge is exterior to H, then we expand H by adding its associated triangle; and if the edge is interior to H, then we skip the edge and advance u until it emerges from H. In either case, at the start of each iteration, u is a point that is on both P and the boundary of H. When u reaches p_n, H will equal $PH(\{s, P\})$.

We assume the vertices of H (which, by Theorem 8, is a simple polygon) have a ccw ordering. For added flexibility, we will associate with H a variable orientation whose values are *ccw* or *cw*. If a and b are adjacent vertices of H, with b ccw from a, then we will consider b to follow a when H has ccw orientation, and to precede a when H has cw orientation. Our algorithm has these steps:

1. Initialize H to be $\triangle sp_1p_2$, and u to be p_2.
2. If $u = p_n$, stop.
3. Set the orientation of H to ccw (resp., cw) if s lies to the left (resp., right) of $\overline{uv}$, where v is the vertex of P following u.[3]
4. If the points of P immediately following u lie in the interior of H, go to step 6.
5. **Expansion step.** Let $T = \triangle suv$ ($\overline{uv}$'s contribution to the hull). Starting with $x = u$, advance x along H, deleting those edges that lie within T, until the first of three events occurs:

   (i) H has no more edges (Figure 4). Replace H with T, advance u to v, and go to step 2.

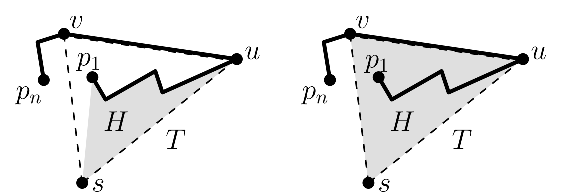

Figure 4: Expansion step, case (i), before and after changes to H.

   (ii) x reaches the point where edge $\overline{ab}$ of H intersects $\overline{sv}$. Replace $\overline{ab}$ with edges $\overline{uv}$, $\overline{vx}$, and $\overline{xb}$ (Figure 5). Advance u to v, and go to step 2.

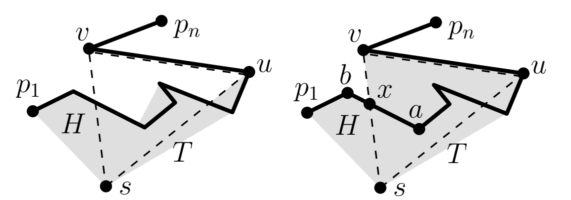

Figure 5: Expansion step, case (ii), before and after changes to H.

   (iii) x reaches the point where edge $\overline{ab}$ of H intersects $\overline{uv}$. Replace $\overline{ab}$ with $\overline{ux}$ and $\overline{xb}$ (Figure 6). Advance u to x, and go to step 2.

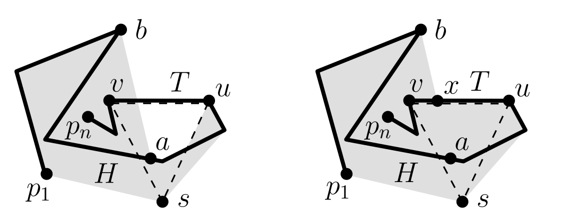

Figure 6: Expansion step, case (iii), before and after changes to H.

[3] In each of the figures that follow, H has ccw orientation according to this rule.

6. **Interior step.** Starting with $x = u$, advance x along P until the first of two events occurs:
   - (i) x reaches p_n; stop.
   - (ii) x reaches the point where P emerges from H's interior (Figure 7). Split $\overline{cd}$, the edge of H containing this point, into edges $\overline{cx}$ and $\overline{xd}$. Advance u to x, and go to step 2.

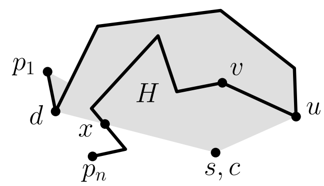

Figure 7: Interior step, case (ii).

We will use induction to show that at the start of each iteration of the algorithm, the following invariants hold:

1. H is the possible hull of s and $(p_1 \dots u)$.
2. u lies on the boundary of H.

The invariants clearly hold for the base case, since the initial hull, H, is triangle $\triangle sp_1u$ (where $u = p_2$), and is thus equal to $PH(\{s, (p_1 \dots u)\})$.

Suppose the invariants hold for every iteration until u reaches a particular position along P, and an expansion step is to be performed. If every edge of H lies within T (case i), then T is the possible hull of $(p_1 \dots v)$, and the invariants are satisfied. Suppose instead that some edge of H does not lie within T. Consider the boundary points of H following u. Since s lies in the kernel of H, the first such point where the boundary crosses an edge of T must lie on $\overline{sv}$ (case ii), or in the interior of $\overline{uv}$ (case iii). In the former case, modifying the boundary of H as stated has the effect of expanding H to include $\triangle suv(= T)$; and in the latter, it has the effect of expanding H to include $\triangle sux(\subset T)$. Since u is advanced to v in the former and x in the latter, we are thus adding exactly that portion of the hull contributed by those points of P between the old and new u, which satisfies invariant (1); and since the new u is not interior to H, invariant (2) is also satisfied.

Now consider the case where an interior step is to be performed. It is easy to show that if H is the possible hull of s and a set J, then $H = PH(\{s, H \cup J\})$. Hence, we can ignore points on $(u \dots x)$. The only change we make to H is to split edge $\overline{cd}$ at x. As this does not actually change the boundary of H, and x (the new location of u) lies on this boundary, both invariants are satisfied.

Since the invariants hold for each iteration of the algorithm, we can claim:

Theorem 9 *The above algorithm generates the possible hull of a point and a simple polygonal chain.*

We now examine the running time of the algorithm. To simplify the analysis, we assume that s is not collinear with any two vertices of P. Step 4 takes constant time, since the points of P immediately following u lie in the interior of H iff the vertex of H following u is to the right (or, if the hull has cw orientation, to the left) of $\overline{uv}$.

In step 6, we must determine which edge of H contains the point x where P emerges from H's interior. To do this efficiently, we start by characterizing each boundary edge of H as being either a *polygonal* edge (lying on an edge of P) or a *radial* edge (lying on a ray from s through a vertex of P). By assumption, no edge can be both.

Lemma 10 *At the start of any iteration in which an interior step occurs, the following conditions hold: (i) exactly one of the edges of H incident with u is a radial edge; and (ii) if this radial edge is not incident with s, then x (the point where P emerges from H's interior) must lie on this edge as well; otherwise, x must lie on an edge incident with s.*

Proof. At the start of an interior step, the edges of H incident with u cannot both be polygonal edges, otherwise (since an interior step is about to occur) this would imply that u is incident to three edges of P, which is impossible. Suppose instead that that they are both radial edges. Note that each radial edge of H is induced by a distinct vertex of P (unless a radial edge has just been split in step 6(d); but each such step is immediately followed by an expansion step that removes one of these two edges). Hence, s and two distinct vertices of P must be collinear, contradicting our general position assumption.

To prove (ii), we first note that since P is simple, x must lie in the interior of $\overline{cd}$, a radial edge of H. Let $\overline{ab}$ be the radial edge of H incident with u. Assume by way of contradiction that $\overline{ab}$ and $\overline{cd}$ are distinct edges, and that at least one of them is not incident with s. This edge must then be adjacent to (distinct) polygonal edges y_1 and y_2 (see, for example, edge $\overline{ab}$ in Figure 8). Now observe that $(u \dots x)$ partitions H into two pieces, and since $\overline{ab} \neq \overline{cd}$, y_1 and y_2 must lie on opposite sides of $(u \dots x)$. We now have a contradiction, since the interiors of paths $(p_1 \dots u)$ and $(u \dots x)$ must intersect, which implies that P is nonsimple. □

Lemma 10 implies that in order to find x while moving along edges of P during an interior step, we need to check for intersections of P with at most two edges of H: the single radial edge incident with the point of entry u, and (if that edge is also incident with s) the other edge incident with s.

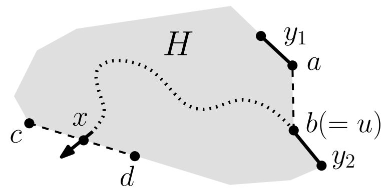

Figure 8: Lemma 10.

Let us determine the total number of vertices processed by the algorithm. These include the vertices of P, plus any vertices that ever appear in H. Consider the start of a particular iteration, where H is the current hull, u is the current position on P, and v is the vertex of P following u. We will show that at most three vertices are introduced to H by the addition of triangle $T = \triangle suv$.

Each new vertex (other than v) is a point where the boundaries of H and T cross, and hence must lie on either $\overline{uv}$ or $\overline{vs}$ (since $\overline{su} \subset H$). Suppose for the sake of contradiction that two new vertices, p and q, lie in the interior of $\overline{uv}$. We can assume that $\overline{uv}$ first enters H at p, then exits H at q. Since $\overline{uv} \subset P$, and P is simple, p and q must lie on radial edges of H. These radial edges must lie on rays $\overrightarrow{sa}$ and $\overrightarrow{sb}$ respectively, where a and b are vertices of P preceding u (Figure 9). The path $(a \dots b \dots u)$ (or $(b \dots a \dots u)$) in P cannot cross rays $\overrightarrow{pa}$ or $\overrightarrow{qb}$ (otherwise p or q would lie in H's interior), nor can it cross $\overline{pq}$ (since P is simple). We now have a contradiction, since this implies that $(a \dots b)$ is not connected to $(u \dots v)$ within P. Hence, at most one new vertex lies in the interior of $\overline{uv}$; and since s is in the kernel of H, at most one of the new vertices lies in the interior of $\overline{vs}$.

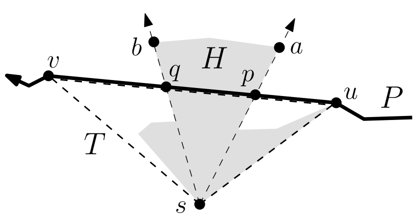

Figure 9: p and q cannot both be new vertices of H.

Theorem 11 *The above algorithm generates the possible hull of a point and a simple polygonal chain of n vertices in $O(n)$ time.*

Proof. We store the vertices of H and P in doubly-linked lists, so that inserting or removing a vertex, or accessing a vertex's neighbor, can be done in constant time. Every step of the algorithm can be done in constant time, except for the expansion and interior steps, which can be done in time proportional to the number of vertices that are: (i) visited on the chain; (ii) inserted into the hull; or (iii) removed from the hull. Since a vertex can be removed from the hull only once, the total running time of the algorithm is bounded by the number of vertices processed by the algorithm (which, as we have shown, is $O(n)$) and the number of iterations (which is also $O(n)$, since each advances u to a distinct vertex of H or P). $\square$

4 Possible Hull of Polygons

In this section, we provide an overview of an algorithm to construct the possible hull of a pair of uncertain polygons A and B (a more detailed presentation, which includes a correctness proof, can be found in [7]). It employs the algorithm of the previous section as a subroutine to achieve an optimal running time.

Our algorithm starts with a polygon H equal to $\mathrm{CH}(A \cup B)$, then modifies H's boundary until H is equal to $PH(\{A, B\})$. If a boundary edge of $\mathrm{CH}(A \cup B)$ is a polygonal segment or a bridge segment, then by Lemmas 2 and 3, and the fact that $PH(\{A, B\}) \subseteq \mathrm{CH}(A \cup B)$, it lies on the boundary of $PH(\{A, B\})$ as well. Otherwise, its vertices must be nonadjacent vertices from the same polygon (e.g., a_i and a_j). As we manipulate H, both a_i and a_j will remain vertices of H, but the path $(a_i \dots a_j)$ on H's boundary will change. We will refer to this path as a *pocket* of H, and to segment $\overline{a_i a_j}$ as the pocket's *lid*.

Our algorithm has these steps:

1. Initialize H to $\mathrm{CH}(A \cup B)$.
2. For each pocket lid $\overline{a_i a_j}$ of H, perform the following *hull contraction* steps:
   (a) Using the algorithm of the previous section, construct J, the possible hull of $(a_i \dots a_j)$ (on the boundary of A) and any point s from B.
   (b) Replace $\overline{a_i a_j}$ with path $(a_i \dots a_j)$ on the boundary of J.
3. Repeat the hull contraction steps with the roles of A and B reversed.
4. Perform the following *hull expansion* steps:
   (a) Set u to h_1, any vertex of $\mathrm{CH}(A \cup B)$.
   (b) Determine ray T_u as follows. If u is a vertex of A, then set T_u to the ray from u that is left-tangent to B; otherwise, set T_u to the ray from u that is left-tangent to A.
   (c) If the vertex v of H following (i.e., in ccw direction) u is not left of T_u, then go to (f).
   (d) Let x be the point where the boundary of H next crosses T_u.

(e) Replace $(u \dots x)$ with $\overline{ux}$.

(f) Advance u to the next convex vertex of H. If $u \neq h_1$, go to (b).

5. Repeat the hull expansion steps, substituting cw for ccw, and right for left.

The hull contraction steps use the algorithm of the previous section to replace each pocket lid with a portion of the boundary of a possible hull associated with the pocket (Figure 10). This contracts H by 'taking bites' out of the convex hull of the two polygons. Corollary 6 implies that after this modification, each pocket lies within PH. The hull expansion steps traverse the boundary of H, find tangent rays that potentially contain bridge segments of PH, and modify H to incorporate these segments. This has the effect of expanding H, by adding back some portions that were removed in the hull contraction steps. It can be shown that after the hull expansion steps have been performed for both ccw and cw directions, $H = PH(\{A, B\})$.

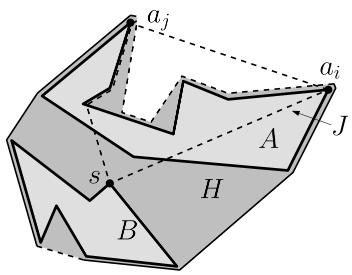

Figure 10: Hull contraction step: pocket lid $\overline{a_i a_j}$ replaced by $(a_i \dots a_j)$ of J.

Step 1 can be performed in $O(n)$ time, where n is the number of vertices of A and B: first by constructing the convex hulls of both A and B (in $O(n)$ time, e.g., by using Melkman's algorithm [4]); then by using the rotating calipers method [8] to construct $\mathrm{CH}(A \cup B)$.

In the hull contraction steps, for each pocket lid $\overline{a_i a_j}$, we construct the corresponding subset of the boundary from A, then calculate the possible hull of this boundary and an arbitrary point of B. Since each edge of A appears in only one of these subsets, and the possible hulls can be constructed in time linear in the size of the subset (Theorem 11), we can perform these steps in $O(n)$ time.

Step 2(b) plays a crucial role in the algorithm. It ensures that throughout the hull expansion steps, each pocket $(a_i \dots a_j)$ is a sequence of points that have monotonically increasing polar angles with respect to a point $s \in B$ (the pocket would not necessarily have this property if, for example, it was instead initialized to path $(a_i \dots a_j)$ on the boundary of A). The monotonicity property implies that the tangent rays T_u in step 4(b) can be found by using the rotating calipers method [8], which (it can be shown) implies that the running time of each hull expansion step is $O(n)$. Hence:

Theorem 12 *The possible hull of a pair of polygons with n total vertices can be constructed in $O(n)$ time.*

The running time of our algorithm is clearly optimal, since it matches the input size. We can adapt the algorithm to the case where there are more than two polygons:

Theorem 13 *The possible hull of k polygons with a total of n vertices can be constructed in $O(n \log k)$ time, and this running time is optimal in the worst case.*

Proof. Lemma 4 implies that we can apply our $O(n)$ algorithm for pairs of polygons recursively, in a divide-and-conquer manner, to construct the possible hull of k polygons. In doing so, we increase the running time by a factor that is logarithmic in the height of a binary tree of k elements. It is worst-case optimal, since if the input consists of $n/3$ small triangles distributed along a circle, the problem reduces to constructing the convex hull of $O(n)$ points (each of which lies on the hull). $\square$

Acknowledgment

The authors would like to thank David Kirkpatrick for his helpful discussions and valuable insights.

References

[1] A. Edalat and A. Lieutier. Foundation of a computable solid modelling. *Theor. Comput. Sci.*, 284(2):319–345, 2002.

[2] A. Edalat, A. Lieutier, and E. Kashefi. The convex hull in a new model of computation. In *CCCG*, pages 93–96, 2001.

[3] M. Löffler and M. J. van Kreveld. Largest and smallest convex hulls for imprecise points. *Algorithmica*, 56(2):235–269, 2010.

[4] A. A. Melkman. On-line construction of the convex hull of a simple polyline. *Inf. Process. Lett.*, 25(1):11–12, April 1987.

[5] T. Nagai, Y. Seigo, and N. Tokura. Convex hull problem with imprecise input. In *Revised Papers from the Japanese Conference on Discrete and Computational Geometry*, pages 207–219, London, UK, 2000. Springer-Verlag.

[6] T. Nagai and N. Tokura. Tight error bounds of geometric problems on convex objects with imprecise coordinates. In *JCDCG '00: Revised Papers from the Japanese Conference on Discrete and Computational Geometry*, pages 252–263, London, UK, 2001. Springer-Verlag.

[7] J. Sember. *Guarantees Concerning Geometric Objects with Uncertain Imputs.* PhD thesis, University of British Columbia, Forthcoming 2011.

[8] M. I. Shamos. *Computational geometry.* PhD thesis, Yale University, 1978.