Relative Convex Hulls in Semi-Dynamic Subdivisions

Mashhood Ishaque[1⋆] and Csaba D. Tóth[2⋆⋆]

[1] Dept. of Comp. Sci., Tufts University, Medford, MA, mishaq01@cs.tufts.edu
[2] Dept. of Mathematics, University of Calgary, AB, cdtoth@ucalgary.ca

Abstract. We present data structures for maintaining the relative convex hull of a set of points (*sites*) in the presence of pairwise non-crossing line segments (*barriers*) that subdivide a bounding box into simply connected faces. Our data structures have $O((n+m)\log n)$ size for n sites and m barriers. They support $O(m)$ barrier insertions and $O(n)$ site deletions in $O((m+n)\,\text{polylog}\,(mn))$ total time, and can answer analogues of standard convex hull queries in $O(\text{polylog}\,(mn))$ time.
Our data structures support a generalization of the sweep line technique, in which the sweep *wavefront* may have arbitrary polygonal shape, possibly bending around obstacles. We reduce the total time of m online updates of a polygonal sweep wavefront from $O(m\sqrt{n}\,\text{polylog}\,n)$ to $O((m+n)\,\text{polylog}\,(mn))$.

1 Introduction

Relative convex hull of a set of sites in a simply connected polygonal domain. The *convex hull*, $\text{ch}(S)$, of a set S of points (*sites*) in the plane is the shortest polygon that circumscribes S. If the configuration space is restricted to a polygonal domain, then the Euclidean distance is typically replaced by the Euclidean *shortest path* (or *geodesic*) distance. There is a *unique* shortest path between any two sites iff the domain is *simply connected*. We consider two interpretations of the *relative convex hull*. For a finite set S of point sites and a simply connected open polygonal domain P,

• the *geodesic hull* $\text{gh}_P(S)$ is the shortest weakly simple polygon contained in P and circumscribing $S\cap P$;
• the *bubble hull* $\text{bh}_P(S)$ is a collection of *simple* polygons such that they jointly circumscribes all sites of $S\cap P$, each polygon is $\text{gh}_P(S')$ for some subset $S'\subseteq S$, and the number of polygons is minimal.

See Fig. 1(b-c). If D is a set of pairwise interior-disjoint polygons, then we denote by $\text{gh}_D(S)$ (resp. $\text{bh}_D(S)$) the collection of geodesic (resp., bubble) hulls of S w.r.t. the polygons in D. The set of all points circumscribed by the polygons in $\text{gh}_D(S)$ (resp. $\text{bh}_D(S)$) is denoted by $\overline{\text{gh}}_D(S)$ (resp. $\overline{\text{bh}}_D(S)$). Clearly, we have

⋆ Partially supported by NSF grant CCF-0431027.
⋆⋆ Partially supported by NSERC grant RGPIN-328155.

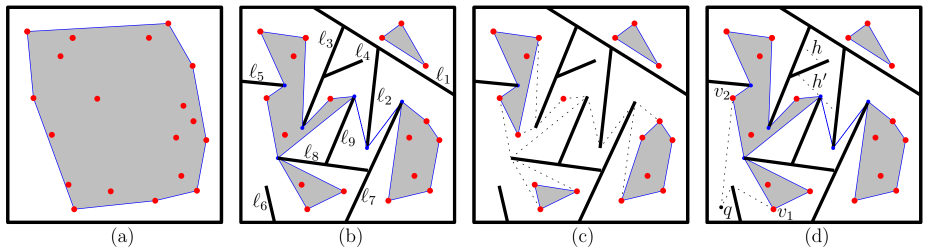


Fig. 1. (a) A convex hull of 20 points; (b) the geodesic hulls of the points in two simply connected domains formed by 9 barriers; (c) the bubble hull of the same point set in the same subdivision; (d) a tangent query q and an extreme point query h.

$\overline{\mathrm{bh}}_D(S) \subseteq \overline{\mathrm{gh}}_D(S)$. Intuitively, the bubble hull is the kernel of the geodesic hull, it can be obtained from the geodesic hull by successively splitting it at singularities (i.e., reflex vertices of P visited twice by the geodesic hull). Obviously. if P is convex, then $\mathrm{gh}_P(S) = \mathrm{bh}_P(S) = \mathrm{ch}(S \cap P)$.

The *geodesic hull* was introduced by Sklansky *et al.* [20] in the digital imaging community and later rediscovered in computational geometry (c.f. [21]).

Results. We present data structures to maintain the bubble hull and the geodesic hull. For n sites and m barriers they both have size $O((n+m)\log n)$, and can be built in $O((n+m)\,\mathrm{polylog}\,(mn))$ preprocessing time. They both support a mixed sequence of $O(m)$ barrier insertions and $O(n)$ site deletions in $O((n+m)\mathrm{polylog}\,(mn))$ total time. The worst case time of a single update operation is $O(m\,\mathrm{polylog}\,(nm))$ for a site deletion, and $O((m+n^{1/2+\delta})\,\mathrm{polylog}\,(mn))$ for a barrier insertion for any $\delta > 0$.

Queries. We would like our data structures to answer certain elementary queries, similar to classical convex hull queries. Here $\mathrm{rch}_P(S)$ may refer to either one of the two relative convex hulls $\mathrm{gh}_P(S)$ and $\mathrm{bh}_P(S)$); queries (iv) and (v) are defined for $\mathrm{gh}_P(S)$ only, since $\mathrm{bh}_P(S)$ may consists of several components:

(i) find the adjacent vertices of $\mathrm{rch}_P(S)$ for a given vertex (*gift-wrapping*);
(ii) decide whether a given point in P lies in $\overline{\mathrm{rch}}_P(S)$ (*point inclusion*);
(iii) find the intersection $\ell \cap \mathrm{rch}_P(S)$ for a given line segment $\ell \subset P$ (*line stabbing*).
(iv) for a given point q in P but outside of $\overline{\mathrm{gh}}_P(S)$, find all vertices $v \in \mathrm{gh}_P(S)$ such that the segment of the geodesics between q and v is tangent to $\mathrm{gh}_P(S)$ at vertex v (the analogue of the *tangent* query, Fig. 1(d)).
(v) For a given chord h of P disjoint from $\mathrm{gh}_P(S)$, find a parallel chord h', if possible, which is tangent to $\mathrm{gh}_P(S)$ and separates h from $\mathrm{gh}_P(S)$ (the analogue of the *extreme point* query, Fig. 1(d)).

Our bubble hull data structure answers query (i), defined below, in $O(1)$ time, since the bubble hull is maintained explicitly; it also answers queries (ii)–(iii) in $O(\mathrm{polylog}\,(mn))$ time. Our geodesic hull data structure answers queries (i)–(v) in $O(\mathrm{polylog}\,(mn))$ time. Our geodesic hull data structure relies on the bubble hull structure, and hence the updates and the queries are slightly more expensive than for the bubble hull.

1.1 Applications

Adversarial polyline sweep. In a classical sweep line algorithm, a vertical line scans the plane from left to right. The sweep line meets any set of sites in the order determined by their x-coordinates. It is not so easy to determine the order in which sites are scanned if the plane is swept by a polyline wavefront, driven by obstacles that has to be avoided and paths that has to be followed, although polygonal wavefronts are used in many applications. We consider a model where we are given n sites, and an adversary sweeps the plane with a polyline wavefront in m *moves*. The wavefront is a simple polygon at all times: initially it is an empty triangle, and each move expands the interior of the polygon by a triangle with one side adjacent to the current wavefront, each move may be modeled as a continuous deformation of the wavefront. Determining the order in which n sites are swept with currently available data structures, would require either m distinct relative convex hull computations (in $O(m(n+m)\log n)$ total time) or m simplex range reporting queries (in $O(m\sqrt{n})$ total time).

The adversarial polyline sweep problem can be solved with our data structure in $O((m+n)\,\mathrm{polylog}\,(mn))$ time, using site deletions and barrier insertions only. For each move of the adversary, we insert the two new edges of the wavefront boundary into our subdivision. These edges separate a triangle adjacent to the current wavefront. With respect to a triangle (or any convex polygon), the relative convex hull is just the (classical) convex hull. We can move continuously the wavefront from one edge of the triangle to the two other edges, and repeatedly delete the first site hit by the wavefront.

1.2 Related results

Sweep-line algorithms. Graham's scan [13] computed the convex hull of n points in the plane in $O(n \log n)$ time by scanning the plane with a line rotating about an extremal point, it is considered the first sweep-line algorithm. A typical plane sweep, of Bentley and Ottmann [2], scans the plane with a vertical line from left to right. The topological sweep of Chazelle and Edelsbrunner [5, 10] (originally developed for optimal segment intersection detection) scans the plane with a polyline *wavefront* which deforms in response to the data it encounters.

Dynamic convex hulls. Preparata [19] gave a semi-dynamic (insert-only) convex hull data structure, which supports point insertion in $O(\log n)$ time. Chazelle [4] and later Hershberger and Suri [14] gave a semi-dynamic (delete-only) data structure, which supports n point deletions in $O(n \log n)$ time. The classic data structure for fully dynamic convex hull in the plane is due to Overmars and van Leeuwen [18], supporting updates in $O(\log^2 n)$ worst-case time. So far, Brodal and Jacob [3, 15] gave the best data structure for dynamic convex hull in the plane. It supports updates in $O(\log n)$ amortized time, and basic convex hull queries in $O(\log n)$ time.

Geodesic paths. The theoretical study of geodesics in the interior of a simple polygon was pioneered by Toussaint [21, 22]. He showed that the geodesic hull $\mathrm{gh}_P(S)$ of a set S of n points in a simple n-gon P can be computed in $O(n \log n)$

time, and any line segment in the interior of P crosses at most two edges of $\mathrm{gh}_P(S)$. Mitchell [17] and Ghosh [11] survey results on geometric shortest paths.

Dynamic subdivisions. Chiang *et al.* [9] dynamically maintained the trapezoidal subdivision of n noncrossing segment barriers in the plane with $O(\log^3 n)$ amortized update time. Goodrich and Tammasia [12] gave an improved method based on balanced geodesic triangulation for maintaining dynamic planar subdivisions. The data structure uses $O(n)$ space and $O(\log^2 n)$ update time.

Range reporting. A data structure for simplex reporting, which is based on Matoušek's technique of simplicial partitioning with low crossing number [16], uses $O(n \operatorname{polylog} n)$ space, and achieves a query time of $O(n^{1/2+\epsilon} + k)$. The best lower bound for simplex reporting queries in the plane is due to Chazelle and Rosenberg [7] who showed that, on a pointer machine, a query time of $O(n^{\delta} + k)$ requires $\Omega(n^{2(1-\delta)-\epsilon})$ space. Thus for any data structure for planar simplex reporting that uses $O(n \operatorname{polylog} n)$ space, there is a lower bound of $\Omega(n^{1/2-\epsilon} + k)$ on the query time.

2 Tools

Shortest paths, point location, and ray shooting in a dynamic subdivision. Chazelle *et al.* [6] showed that a balanced geodesic triangulation of a polygon with n vertices can be used for answering ray shooting queries in the polygon in $O(\log n)$ time. Goodrich and Tamassia [12] generalized this data structure to dynamic subdivisions defined by noncrossing line segments where each face is a simple polygon. They maintain a balanced geodesic triangulation of each face. For m segments, the data structure has $O(m)$ size. Each segment insertion and deletion, point location, and ray shooting query takes $O(\log^2 m)$ time. It reports a shortest (geodesic) path between two points in $O(\log^2 m + k)$ time, where k is the length of the path. However, it can report $O(1)$ information about the shortest paths in $O(\log^2 m)$ time.

Geometric partition trees. A *geometric partition tree* for n points in the plane is a rooted binary tree T where (1) every node $v \in T$ corresponds to a convex cell C_v in the plane; (2) the root at level 0 corresponds to the plane (or a bonding box); (3) for every nonleaf node $v \in T$ the cell C_v is tiled by the two convex cells corresponding to the children of v; and (4) every cell C_v, $v \in T$ at level k of T contains at most n/λ^k points, for some fixed λ, $1 < \lambda \leq 2$. In particular, the convex cells C_v, for all leaf nodes $v \in T$ form a subdivision of the entire plane (or the bounding box).

Geometric partition trees with low stabbing numbers. The *stabbing number* of a geometric graph or a subdivision is the maximum number of edges crossed by a straight line. Chazelle *et al.* [8, 1] showed that one can construct a geometric partition tree in $O(n \log n)$ time such that the stabbing number of the corresponding subdivision of the plane is $O(n^{\frac{1}{2}+\delta})$, for any fixed $\delta > 0$.

3 Barriers and geometric partition trees

We present a few basic structural properties of the bubble hull (without proof).

Proposition 1. *1. For a set S of sites and a simply connected domain P, the bubble hull $\mathrm{bh}_P(S)$ is unique.*

2. A line segment $\ell \subset P$ intersects both $\mathrm{gh}_P(S)$ and $\mathrm{bh}_P(S)$ in at most two points.

3. If $P_1 \subset P_2$, then $\overline{\mathrm{bh}}_{P_1}(S) \subseteq \overline{\mathrm{bh}}_{P_2}(S)$.

The motivation for introducing bubble hulls is the following feature of geodesic hulls: The insertion of m barriers may induce $\Omega(m(m+n))$ combinatorial changes in $\mathrm{gh}_P(S)$ (see Fig. 2). However, we show in Proposition 2 below that the insertion of m barriers induce only $O(m+n)$ combinatorial changes in $\mathrm{bh}_P(S)$.

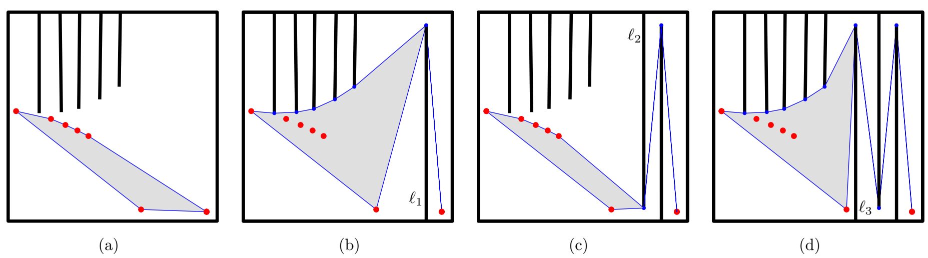


Fig. 2. The successive insertion of m barriers may induce $\Omega(m(m+n))$ combinatorial changes in $\mathrm{gh}_P(S)$.

Representation of geodesic hulls. In each simply connected face f, the geodesic hull $\mathrm{gh}_f(S)$ is connected and its vertices are sites and barrier endpoints. If two consecutive vertices are sites, then they belong to the same component of $\mathrm{bh}_f(S)$. The portions of $\mathrm{gh}_f(S)$ between two consecutive sites is the geodesic path between those sites. We represent $\mathrm{gh}_f(S)$ as a cyclic alternating sequence of paths $(b_1, g_1, b_2, g_2, \ldots, g_k)$, where b_i is a portion of a component of $\mathrm{bh}_f(S)$ between two sites and g_i is a geodesic path between sites along distinct components of $\mathrm{bh}_f(S)$; only the first and last vertex of each portion will be stored.

The backbone of our data structure is a geometric partition tree T for n point sites in a bounding box B, which is computed at preprocessing and remains fixed thereafter. Each node $v \in V(T)$ corresponds to a cell C_v. If D is a subdivision of the bounding box B into simply connected faces, let $D(v)$ denote the subdivision of cell C_v by the portion of the barriers clipped to C_v. Storing $D(v)$ for every node $v \in V(S)$ would be prohibitively expensive: if every barrier intersects $\Omega(\sqrt{n})$ cells then storing $D(v)$ for all leaf nodes would require $\Omega(m\sqrt{n})$ space. Therefore, we store only some carefully chosen portions of $D(v)$ (c.f. Section 4). The size of the relative convex hulls, however, for all $v \in V(T)$ is close to linear.

Lemma 1. *The union of all $\mathrm{bh}_{D(v)}(S)$ (resp., $\mathrm{gh}_{D(v)}(S)$) for all nodes $v \in V(T)$ of a geometric partition tree T is a plane graph with $O(n + m\log n)$ edges.*

Proof. For the planarity, note that due to the hierarchy of the geometric partition tree T, we have $\overline{\mathrm{bh}}_{D(w)}(S) \subseteq \overline{\mathrm{bh}}_{D(v)}(S)$ and $\overline{\mathrm{gh}}_{D(w)}(S) \subseteq \overline{\mathrm{gh}}_{D(v)}(S)$ wherever w is a descendant of v. Hence, the edges of the bubble hulls (resp., geodesic hulls) of different levels cannot cross.

Assume w.l.o.g. that the $2m$ endpoints of the barriers are disjoint (using virtual coordinates, if necessary). We say that an endpoint of barrier ℓ is incident to a face f if the endpoint and an incident portion of ℓ lie on the boundary of f. At the leaf nodes of T, $S \cap C_v$ is a singleton, and its convex hull has no edges. Consider now a non-leaf node $v \in V(T)$ whose children are w_1 and w_2. The faces of $D(w_1)$ are separated from the faces of $D(w_2)$ by a line h_v. Each face $f \in D(v)$ is the union of some faces $F_1 \subseteq D(w_1)$ and $F_2 \subseteq D(w_2)$ of the two child subdivisions (Fig. 3, middle). Hence, $\mathrm{rch}_f(S)$ can be constructed by merging the relative convex hulls $\mathrm{rch}_{D(w_1)}(S)$ and $\mathrm{rch}_{D(w_2)}(S)$. The merge step creates new edges along geodesic paths between components of $\mathrm{rch}_{D(W_1)}(S)$ and $\mathrm{rch}_{D(w_2)}(S)$ (each may consists of several components). Each geodesic is either a single edge (common tangent) or passes through barrier endpoints lying in C_v. At most two geodesics pass through any barrier endpoint in $\mathrm{gh}_f(S)$, and at most one geodesic for $\mathrm{bh}_f(S)$. If the merge step reduces the number of components by γ_v and the new geodesics pass through m_v barrier endpoints, then $O(\gamma_v + m_v)$ new edges are created. Summing up the terms $O(\gamma_v)$ over all $v \in V(T)$, we obtain $O(n)$. Summing up the terms $O(m_v)$ over all $v \in V(T)$ at a level of T, we have $O(m)$, which gives $O(m \log n)$ over all $\log n$ levels of T.

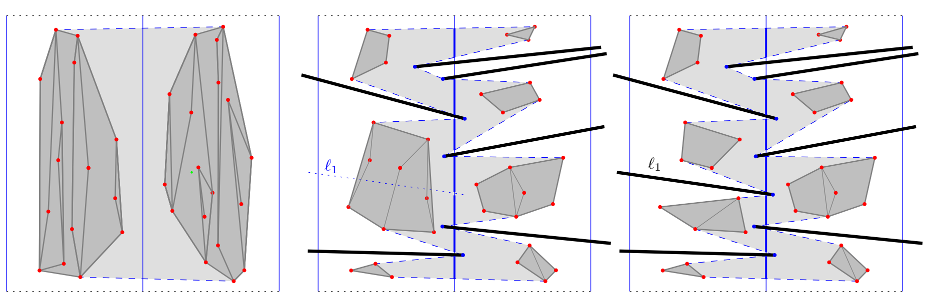


Fig. 3. Merging the relative convex hulls of two convex hulls (left). Merging the geodesic hulls of the subdivisions of two consecutive vertical slabs before (middle) and after (right) inserting a new line segment ℓ_i.

Combinatorial changes. If a new barrier ℓ partitions a face $f \in D(v)$ into two faces f_1 and f_2, we will replace $\mathrm{rch}_f(S)$ by $\mathrm{rch}_{f_1}(S)$ and $\mathrm{rch}_{f_2}(S)$. If ℓ has an endpoint in the interior of f (note that at most one endpoint of ℓ may lie in the interior of f since every face is simply connected), then f is deformed to a face f' and we will replace $\mathrm{rch}_f(S)$ by $\mathrm{rch}_{f'}(S)$. Of course, an update of a relative convex hull w.r.t. the subdivision $D(v)$ is necessary only if ℓ intersects $\mathrm{rch}_f(S)$. The following lemma counts the intersections between the edges of the relative convex hulls and successively inserted barriers.

Lemma 2. *We are given a set S of n sites in a bounding box B and a geometric partition tree T. If we update $\mathrm{rch}_{D(v)}(S)$ for all $v \in V(T)$ during an intermixed sequence of m barrier insertions and $O(n)$ site deletions, then there are altogether $O(m+n)$ intersections between new barriers and current edges of $\mathrm{rch}_{D(v)}(S)$ for all $v \in V(T)$ at each level of T.*

Proof. For every $i \in \mathbb{N}$, there are $O(2^i)$ nodes $v \in V(T)$ at level i such that C_v contains at least $n/2^{i+1}$ and less than $n/2^i$ sites. Distinguish two types of intersections between a new barrier ℓ and the current relative convex hull $\mathrm{rch}_{D(v)}(S)$ for $v \in V(T)$.

Type 1: ℓ partitions a current face $f \in D(v)$ into two faces, both of which contain sites of S. In this case, the number of connected components of $\mathrm{rch}_{D(v)}(S)$ increases by one. A set of k sites can recursively be partitioned into nontrivial subsets at most $k-1$ times. Hence, summing all type 1 events for all m barriers and all $v \in V(T)$ at level i, we obtain $O(n/2^i)O(2^i) = O(n)$.

Type 2: ℓ has an endpoint in the interior of a current face $f \in D(v)$, which contains sites of S. At each level of T, each barrier endpoint lies in a unique cell of $D(v)$ for a unique $v \in V(T)$. Hence there are $O(n)$ type 2 events. □

In Section 4 below, we describe in detail how to maintain the relative convex hulls w.r.t. each subdivision $D(v)$, $v \in V(T)$.

Proposition 2. *Given n sites in a bounding box B, a mixed sequence of m barrier insertions and $O(n)$ site deletions induce $O(m+n)$ combinatorial changes in $\mathrm{bh}_D(S)$.*

Proof. $\mathrm{bh}_D(S)$ may split into several components due to a barrier insertion or a site deletion. A barrier insertion and a site deletion both decrease the region circumscribed by the bubble hull, that is, $\overline{\mathrm{bh}}_{D_1}(S_1) \subseteq \overline{\mathrm{bh}}_{D_2}(S_2)$ if $S_1 \subseteq S_2$ and $\mathrm{int}(D_1) \subseteq \mathrm{int}(D_2)$. Hence, if a point $p \in S$ is a vertex of $\mathrm{bh}_D(S)$ at one step, it remains a vertex of a polygon of $\mathrm{bh}_D(S)$ until p is deleted. If an endpoint q of a barrier ℓ is a vertex of $\mathrm{bh}_D(S)$ at one step, it remains a vertex of a polygon in $\mathrm{bh}_D(S)$ until all sites on one side of the line through ℓ are deleted, or a component of $\mathrm{bh}_D(S)$ visits q twice and it is split into two components. Every edge deletion or creation in $\mathrm{bh}_D(S)$ can be charged to an event involving an endpoint of that edge. There are four possible events: a site $s \in S$ becomes a vertex of $\mathrm{bh}_D(S)$; a vertex $s \in S$ of $\mathrm{bh}_D(S)$ is deleted; a barrier endpoint q becomes a vertex of $\mathrm{bh}_D(S)$; or a barrier endpoint q is no longer a vertex of $\mathrm{bh}_D(S)$. There are $n+m$ possible events, each one is responsible for two edge changes in $\mathrm{bh}_D(S)$. □

Proposition 3. *Given n sites in a bounding box B, a mixed sequence of m barrier insertions and $O(n)$ site deletions induce $O(m+n)$ combinatorial changes in the our representation of $\mathrm{gh}_D(S)$ as a cyclic alternating sequences of portions of $\mathrm{bh}_D(S)$ and geodesic paths.*

Proof (sketch). A site deletion or barrier insertion may trigger the splitting of a component of $\mathrm{bh}_D(S)$ into several components $O(m)$ times (at most once for each barrier). If the corresponding portion of $\mathrm{bh}_S(D)$ lies in $\mathrm{gh}_D(S)$, then it is replaced by two portions of the resulting components of $\mathrm{bh}_D(S)$ and a geodesic path between them, that is, $O(1)$ combinatorial changes in the representation of $\mathrm{gh}_D(S)$. Besides the effects of splitting the bubble hull into several components, each site deletion or barrier insertion incurs only $O(1)$ change in the affected portion of $\mathrm{gh}_D(S)$ and in the two adjacent portions. □

4 Data structure

Bubble hull data structure. We are given a set of n sites and a subdivision D formed by set of barriers in a bounding box B. Our data structure has three main components: (1) a geometric partition tree T for S, where every node $v \in V(T)$ will store numerous items; (2) plane graphs G_i, $i = 1, 2, \ldots, \log n$, one for each level of T. The vertices of G_i are the sites and the barrier endpoints, the edges are formed by the barriers, all bubble hulls $\mathrm{bh}_{D(v)}(S)$ for the subdivisions $D(v)$ at level i of the T, and an anchor edge between each convex component of $\mathrm{bh}_{D(v)}(S)$ and a nearby barrier endpoint, the faces are simply connected; (3) dynamic data structures of Goodrich and Tamassia [12] for each face of each G_i.

At each node $v \in V(T)$, we store C_v. We store some faces of the subdivision $D(v)$. Let the *parent* of a face $f \in D(v)$ be the face $f' \in D(v')$ such that $v' \in V(T)$ is the parent of v and $f \subseteq f'$. At node v, we store a face $f \in D(v)$ if f or its parent contains a site of S; we also store f if f or its parent is incident to a barrier endpoint. We store some edges of $\mathrm{bh}_{D(v)}(S)$. For a leaf $v \in V(T)$, we have $|S \cap C_v| \leq 1$, and so $\mathrm{bh}_{D(v)}$ has no edges. For a nonleaf node $v \in V(T)$, we store the line h_v that partitions C_v into two cells. We store each segment of h_v clipped in a stored face $f \in D(v)$. We store each component of $\mathrm{bh}_{D(v)}(S)$ in a doubly linked edge list and a binary search tree; and store also the cyclic list of sites along each component.

We store the plane graph G_i, $i = 1, 2, \ldots, \log n$, formed by the barriers and all edges of $\mathrm{bh}_{D(v)}(S)$ for all $v \in V(T)$ at level i of T (see Fig. 3). In addition, for every convex component of $\mathrm{bh}_{D(v)}(S)$, we store an *anchor* edge that connects it to a reflex vertex of D (recall that the components of $\mathrm{bh}_D(S)$ are mutually occluded from each other). The first edge where a component of $\mathrm{bh}_{D(v)}(S)$ diverge from $\mathrm{gh}_{D(v)}(S)$ is a good choice for an anchor. The anchors divide the region $f \setminus \overline{bh}_f(S)$ surrounding the bubble hull into a *simply connected* face (e.g., Fig. 3, right). Let $\Phi(G_i)$ denote the set of faces of G_i. We maintain the dynamic data structure of [12] for each face $\varphi \in \Phi(G_i)$. This completes our data structure for maintaining the bubble hull.

Geodesic hull data structure. In addition to all components of the bubble hull data structure, we store $\mathrm{gh}_{D(v)}(S)$ for every $v \in V(T)$. Here $\mathrm{gh}_{D(v)}(S)$ is represented as an cyclic alternating sequence of paths $(b_1, g_1, b_2, g_2, \ldots, g_k)$, where b_i is a portion of a component of $\mathrm{bh}_{D(v)}(S)$ (represented by the counter-clockwise first and last sites), and g_i is a geodesic path in a face of Φ between two sites of different components of $\mathrm{bh}_{D(v)}(S)$ (represented by the two sites).

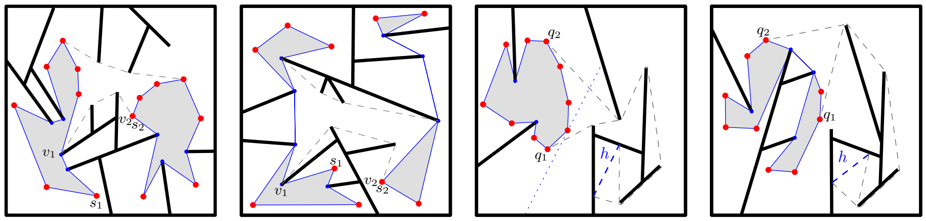
Space requirement. The geometric partition tree T has size $O(n)$. Recall that the cells C_v for nodes $v \in V(T)$ at each level correspond to a partition of the bounding box B. If a face $f \in D(v)$ is incident to m_f barrier endpoints, then it has at most $4 + m_f$ edges. Each barrier endpoint is incident to two faces at each level of T. Hence storing faces of all subdivisions $D(v)$, which are either incident to a barrier endpoint or contain a site, requires $O((n+m)\log n)$ space. The bubble hulls $\mathrm{bh}_{D(v)}(S)$ for all nodes v at a single level of T jointly have $O(n+m)$ edges. Storing $\mathrm{bh}_{D(v)}(S)$ explicitly for all $v \in V(T)$ requires $O((n+m)\log n)$ space. The size of each G_i is $O(n+m)$, and so the dynamic data structure of Goodrich-Tamassia [12] for the all faces of G_i requires $O(n+m)$ space. All graphs G_i, $i = 1, 2, \ldots, , \log n$, jointly use $O((n+m)\log n)$ space. In our representation of geodesic hulls, with selected sites, $\mathrm{gh}_{D(v)}(S)$ for all nodes v at a level of T jointly have at most n edges. Storing $\mathrm{gh}_{D(v)}(S)$ for all $v \in V(T)$ requires $O(n \log n)$ space.

4.1 Updates and queries

Primitives. In the Goodrich-Tamassia data structure [12], each barrier insertion and deletion, point location, or ray shooting query takes $O(\log^2 m)$ time. It reports the shortest path $\pi_f(p_1, p_2)$ between p_1 and p_2 in $O(\log^2 m + k)$ time where k is the size of $\pi_f(p_1, p_2)$. For two points, p_1 and p_2, it can report the first and the last segments, as well as the middle vertex of $\pi_f(p_1, p_2)$ in $O(\log^2 m)$ time. For two points p_1, p_2 and a chord h, it can report the intersection of $\pi_f(p_1, p_2)$ and h in $O(\log^2 m)$ time. Given three points p, q_1, q_2, the first segments where $\pi_f(p, q_1)$ and $\pi_f(p, q_2)$ differ can also be reported in $O(\log^2 m)$ time.

With the queries of the data structure of [12], we can compute the *common tangent geodesics of two disjoint geodesic hulls*: Given two sets of sites, S_1 and S_2 in a face f such that $\mathrm{gh}_f(S_1)$ and $\mathrm{gh}_f(S_2)$ are disjoint, find the pairs of vertices $(v_1, v_2) \in S_1 \times S_2$ such that $\pi_f(v_1, v_2)$ is the shortest path in f tangent to $\mathrm{gh}_f(S_1)$ and $\mathrm{gh}_f(S_2)$ at the endpoints v_1 and v_2, respectively (Fig. 4(ab)). The common tangent geodesics can be used for merging the geodesic hulls into $\mathrm{gh}_f(S_1 \cup S_2)$. Searching for v_1 and v_2 is analogous to finding the common tangents between disjoint convex polygons [18]. With $O(\log n)$ shortest path queries, each in $O(\log^2 m)$ time, we can find pairs of *sites* $(s_1, s_2) \in S_1 \times S_2$ such that $\pi_f(s_1, s_2)$ is tangent to $\mathrm{gh}_f(S_1)$ and $\mathrm{gh}_f(S_2)$ at s_1 and s_2, respectively. If the endpoints s_1 and s_2 are the only common vertices of $\pi_f(s_1, s_2)$ with $\mathrm{gh}_f(S_1)$ and $\mathrm{gh}_f(S_2)$, then $v_1 = s_1$ and $v_2 = s_2$. Otherwise, v_1 and v_2 are the first vertices where $\pi_f(s_1, s_2)$ diverges from $\mathrm{gh}_f(S_1)$ and $\mathrm{gh}_f(S_2)$, resp., and hence v_1 and v_2 can be computed with the above mentioned three-point query of [12] in $O(\log^2 m)$ time. The common tangent geodesics can be computed in $O(\log n \log^2 m)$ time.

The common tangent geodesics can be used to compute the bubble hull from the geodesic hull and the bubble hulls of the two subfaces. Let f be a face of a subdivision $D(v)$, where a line h_v partitions f into some faces $f_1, f_2, \ldots, f_\kappa$. Assume for a moment that we are given $\mathrm{gh}_f(S)$ explicitly and $\mathrm{bh}_{f_k}(S)$ for all $k = 1, 2, \ldots, \kappa$. Then we can compute $\mathrm{bh}_f(S)$ by successively pruning the singularities (i.e., vertices visited twice by $\mathrm{gh}_f(S)$). Refer to Fig. 1(c). If p is a singularity



(a) (b) (c) (d)

Fig. 4. The common tangent geodesics between two geodesic hulls (ab). The common geodesic hulls between a chord h and a geodesic hull (cd).

(which is necessarily a barrier endpoint), then find the two pairs of closest sites, (s_1, s_2) and (s_3, s_4) along $\mathrm{gh}_f(S)$ such that s_1, s_3 are on the same side of p. Here both $\pi_f(s_1, s_2)$ and $\pi_f(s_3, s_4)$ pases through p. Denote by R_j the component of a bubble hull $\mathrm{bh}_{f_k}(S)$ that contains s_j for $j = 1, 2, 3, 4$. Split the $\mathrm{gh}_f(S)$ into two geodesic hulls, by removing $\pi_f(s_1, s_2)$ and $\pi_f(s_3, s_4)$, and inserting instead the common tangent geodesic between R_1, R_3 and the common tangent geodesic between R_2, R_4. The pruning each singularity takes $O(\log n \log^2 m)$ time. If $\mathrm{bh}_f(S)$ has t components, then we can compute it in $O(t \log n \log^2 m)$ time.

Site deletion. Assume we delete site $s \in S$, where s corresponds to a leaf $v_0 \in V(T)$. Update the information at all $\log n$ nodes $v \in V(T)$ where $s \in C_v$. Delete every face $f \in D(v)$ whose parent face contained no other site but s. Update $\mathrm{bh}_{D(v)}(S)$ bottom up. No update is necessary if s is in the interior of $\mathrm{bh}_{D(v)}(S)$. Assume that s is on the vertex of a component $\mathrm{gh}_f(S')$ of $\mathrm{bh}_f(S)$ for some face $f \in D(v)$ and $S' \subseteq S$. If f contains no other site, then $\mathrm{bh}_f(S)$ is deleted. Otherwise assume that s has already been deleted from $\mathrm{bh}_{f_1}(S)$, where f_1 is the child face of f containing s.

Compute $\mathrm{gh}_f(S' \setminus \{s\})$ in $O(\log n \log^2 m)$ time as follows. Let a and b denote the two closest sites of s along $\mathrm{gh}_f(S')$. Let R_a and R_b be the components of the corresponding bubble hulls $\mathrm{bh}_{f_k}(S)$ containing a and b, resp. (which have already been updated). Compute the common geodesic tangent of R_a and R_b. If $\mathrm{gh}_f(S' \setminus \{s\})$ is a simple polygon, then $\mathrm{bh}_f(S' \setminus \{s\}) = \mathrm{gh}_f(S' \setminus \{s\})$ and we are done. However, if $\mathrm{gh}_f(S' \setminus \{s\})$ is not a simple polygon, we need to compute $\mathrm{bh}_f(S \setminus \{s\})$ by successively pruning the singularities as described above (Fig. 1(c)).

Barrier insertion. Let ℓ be a barrier inserted (refer to Fig. 3). Insert ℓ into the Goodrich-Tamassia data structures in $O(\log^2 m)$ time. Since ℓ does not cross other barriers, it intersects at most one face in each subdivision $D(v)$, $v \in V(T)$. In a top-down traversal of T, find all faces $f_v \in D(v)$ in our data structure that intersect ℓ. Update the faces f_v. Some of the faces may be split by ℓ into two faces. If a new face and its parent contain no site, the new face is deleted from our data structure.

Compute all intersection points with $\mathrm{bh}_{D(v)}(S)$ using a ray shooting data structure at $O(\log^2(m+n))$ cost per intersection. Locate all faces $f \in D(v)$ where ℓ intersects $\mathrm{bh}_f(S)$ in a top-down traversal of T. If ℓ is disjoint from $\mathrm{bh}_f(S)$, then it is disjoint from the bubble hull in all descendent faces. Update $\mathrm{bh}_f(S)$ in all faces bottom-up as follows.

Assume that ℓ intersects the component $\mathrm{gh}_f(S')$ of a bubble hull $\mathrm{bh}_f(S)$, for some $f \in D(v)$ and $S' \subseteq S \cap f$. Assume that all descendant faces have already been updated. Distinguish two cases:
Case 1: ℓ intersects $\mathrm{bh}_f(S)$ in exactly one point p. Find the pair of sites, say s_1 and s_2, along $\mathrm{gh}_{f(v)}(S')$ closest to p. Let R_1 and R_2 be the bubble hulls at the children faces of f containing s_1 and s_2, respectively. Compute $\mathrm{gh}_{f\setminus\ell}(S')$ by replacing $\pi_f(s_1, s_2)$ with the common tangent geodesics between R_1 and p, and between p and R_2. If $\mathrm{gh}_{f\setminus\ell}(S')$ is a simple polygon, then we are done; otherwise prune successively the singularities to obtain $\mathrm{bh}_{f\setminus\ell}(S')$.
Case 2: ℓ intersects $\mathrm{bh}_{f(v)}(S)$ in two points. Then ℓ partitions the point set S' into some point sets S_1' and S_2'. Find the pairs of sites, say (s_1, s_2) and (s_3, s_4), along $\mathrm{gh}_f(S')$ closest to the intersection points with ℓ, such that s_1 and s_3 are on the same side of ℓ. Let R_j be the bubble hulls at the children faces of f containing s_j for $j = 1, 2, 3, 4$. Compute $\mathrm{gh}_{f\setminus\ell}(S_1')$ and $\mathrm{gh}_{f\setminus\ell}(S_2')$ by replacing $\pi_f(s_1, s_2)$ and $\pi_f(s_3, s_4)$ with the common tangent geodesics between R_1, R_3 and R_2, R_4 in the face $f \setminus \ell$. If $\mathrm{gh}_{f\setminus\ell}(S_1')$ and $\mathrm{gh}_{f\setminus\ell}(S_2')$ are simple polygons, then we are done; otherwise prune successively the singularities to obtain $\mathrm{bh}_{f\setminus\ell}(S')$. The geodesic hull $\mathrm{gh}_{f\setminus\ell}(S)$ can be updated analogously in a bottom-up traversal.

A single barrier insertion takes $O(\kappa_\ell \text{ polylog}\,(mn))$ time, where κ_ℓ is the number of cells stabbed by ℓ. Here $\kappa_\ell = O(n \log n)$ is a trivial bound (which is tight, e.g., for a partition tree into vertical slabs [18]); and we have $\kappa_\ell = O(m + n^{1/2+\delta})$ for a $\delta > 0$ for a partition tree of low stabbing number [1, 8].

Queries. Here, we discuss query (v) only for space limitations. Refer to the full paper for a detailed description of the remaining queries (i)–(iv).

Query (v) We can check whether the query chord $h = p_1p_2$ is indeed disjoint from $\mathrm{gh}_D(S)$ in $O(\log n \log^2 m)$ time, using the line stabbing query (iii). Then locate the face f of the subdivision D containing h. Compute the common tangent geodesics between h and $\mathrm{gh}_f(S)$, which meet $\mathrm{gh}_f(S)$ at some vertices q_1 and q_2. The chord h, the two common tangent geodesics, and the porion of $\mathrm{gh}_f(S)$ between q_1 and q_2 forms a geodesic quadrilateral (Fig. 4(cd)). Every chord parallel to h that separates h from $\mathrm{gh}_f(S)$ must cross both common tangent geodesics, so the point of tangency along $\mathrm{gh}_f(S)$, if exists, must be between q_1 and q_2. The portion of $\mathrm{gh}_f(S)$ between q_1 and q_2 is a convex polygonal chain, hence it has at most one tangent parallel to h. Since $\mathrm{gh}_f(S)$ is not stored directly, we first find the portion of $\mathrm{gh}_f(S)$ (a geodesic path or a portion of $\mathrm{bh}_f(S)$) that possibly has a tangent parallel to h in a binary search in $O(\log n)$ time. If it is in a portion of a bubble hull, then we can find the tangency point in a binary search in $O(\log n)$ time. If it is in some geodesic path $\pi_f(v_1, v_2)$ along $\mathrm{gh}_f(S)$, then a binary search takes $O(\log m)$ queries with the [12] data structure, in $O(\log^3 m)$ total time. Altogether, query (v) can be answered in $O(\log(mn) \log^2 m)$ time.

References

1. P. K. Agarwal and M. Sharir, Applications of a new space-partitioning technique, *Discrete Comput. Geom.* **9** (1993), 11–38.
2. J. L. Bentley and T. A. Ottmann, Algorithms for reporting and counting geometric intersections, *IEEE Trans. Comput.* **C-28** (9) (1979), 643–647.
3. G. S. Brodal and R. Jacob, Dynamic planar convex hull, in: *Proc. FOCS*, 2002, IEEE, pp. 617–626.
4. B. Chazelle, On the convex layers of a planar set, *IEEE Trans. Inform. Theory* **IT-31** (4) (1985), 509–517.
5. B. Chazelle and H. Edelsbrunner, An optimal algorithm for intersecting line segments in the plane, *J. ACM* **39** (1) (1992), 1–54.
6. B. Chazelle, H. Edelsbrunner, M. Grigni, L. J. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink, Ray shooting in polygons using geodesic triangulations, *Algorithmica* **12** (1994), 54–68.
7. B. Chazelle and B. Rosenberg, Simplex range reporting on a pointer machine, *Comput. Geom. Theory Appl.* **5** (5) (1996), 237–247.
8. B. Chazelle, M. Sharir, and E. Welzl, Quasi-optimal upper bounds for simplex range searching and new zone theorems, *Algorithmica* **8** (1992), 407–429.
9. Y.-J. Chiang, F. P. Preparata, and R. Tamassia, A unified approach to dynamic point location, ray shooting, and shortest paths in planar maps, *SIAM J. Comput.* **25** (1996), 207–233.
10. H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, vol. 10 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1987.
11. S. Ghosh, *Visibility Algorithms in the Plane*. Cambridge University Press, New York, NY, USA, 2007.
12. M. T. Goodrich and R. Tamassia, Dynamic ray shooting and shortest paths in planar subdivisions via balanced geodesic triangulations, *J. Algorithms* **23** (1997), 51–73.
13. R. L. Graham, An efficient algorithm for determining the convex hull of a finite planar set, *Inform. Process. Lett.* **1** (1972), 132–133.
14. J. Hershberger and S. Suri, Applications of a semi-dynamic convex hull algorithm, *BIT* **32** (1992), 249–267.
15. R. Jacob, *Dynamic Planar Convex Hull*, PhD thesis, University of Aarhus, Aarhus, Denmark, 2002.
16. J. Matousek, Geometric range searching, *ACM Comput. Surv.* **26**.
17. J. S. B. Mitchell, Geometric shortest paths and network optimization, in: *Handbook of Computational Geometry*. Elsevier, Amsterdam, 2000.
18. M. H. Overmars and J. van Leeuwen, Maintenance of configurations in the plane, *J. Comput. Syst. Sci.* **23** (1981), 166–204.
19. F. P. Preparata, An optimal real-time algorithm for planar convex hulls, *Commun. ACM* **22** (1979), 402–405.
20. J. Sklansky, R. L. Chazin, and B. J. Hansen, Minimum perimeter polygons of digitized silhouettes, *IEEE Trans. Comput.* **C-21** (1972), 260–268.
21. G. T. Toussaint, Shortest path solves translation separability of polygons, Report SOCS-85.27, School Comput. Sci., McGill Univ., Montreal, PQ, 1985.
22. G. T. Toussaint, An optimal algorithm for computing the relative convex hull of a set of points in a polygon, in: *Signal Processing III: Theories and Applications*. 1986, pp. 853–856.