Dynamic Ham-Sandwich Cuts for Two Point Sets with Bounded Convex-Hull-Peeling Depth

M. A. Burr* J. Hugg* E. Rafalin* K. Seyboth* D. L. Souvaine*

1 Introduction

A *ham-sandwich cut* (HSC) of two subsets S_1 and S_2 in the plane is a line that simultaneously bisects both sets according to some measure. The static problem of finding a HSC of a point set is well studied and a linear time solution exists [6, 7].

We provide an efficient data structure for dynamically maintaining a HSC of two (possibly overlapping) point sets in the plane, with a bounded number of convex-hull (CH) peeling layers[1]. Our algorithm supports insertion and deletion of vertices in $O(c \log n)$ time, area and perimeter queries in $O(\log n)$ time and vertex-count queries in $O(c^3 \log^3 n)$ time, where n is the total number of points of $S_1 \cup S_2$ and c is a bound on the number of CH peeling layers.

Our algorithm considerably improves previous results [8, 1] as it removes the restrictions based on the convex position and the separation of the points. It solves an open problem about finding the area and perimeter HSCs for overlapping convex point sets in the static setting [8] and can be used to approximate the Tukey median in a dynamic setting [1].

2 The Algorithm

Given two dynamic (possibly overlapping) point sets in the plane S_1, S_2, $|S_1 \cup S_2| = n$, where for every point set the number of CH peeling layers is bounded by a constant c, and where points can be added to or deleted from the sets, we wish to find a HSC that bisects the number of points on each point set.

Data Structure: Let P_i^j be the j^{th} CH peeling layer (represented as a convex polygon) of point set P_i (where $i \in \{1, 2\}$ and P_i^0 is the outermost layer which is the CH of the set). Our data structure is a concatenable queue, a binary search tree which enables efficient searching, splitting and concatenation. A total of $O(4c)$ trees exist: the upper and lower hulls for every CH peeling layer. The leaves of the trees contain the ordered vertices of the CH peeling layers and are linked. Each inner node stores the sum of the vertices in its subtree.

Dynamization is enabled by utilizing the fact that when points are inserted or deleted the CH peeling layers change by transition of entire sections of one layer to the adjacent layer. The algorithm updates the layers incrementally, using binary search on a layer P^k to find the two tangent lines from P^k to the single point or to the convex chain that is inserted into the layer.

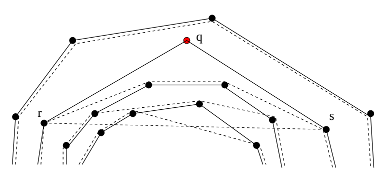

Figure 1: Insertion and deletion of point q and its affect on the CH peeling-layers. The layers with and without point q are drawn as solids and dashed lines respectively. Point q has depth 2, and its insertion or deletion affects layers of depth ≥ 2.

Proposition 1. Given a line l, we can find which edges of P_i^j it intersects in $O(\log n)$.

Proposition 2. Given a line l we can find the number of points of P_i^j above l in $O(\log n)$.

Proposition 3. Given a line l we can find the number of points of S_i above l in $O(c \log n)$.

Lemma 1. Given a point p outside the CH of S_i we can find the bisector of $S_i, i \in \{1, 2\}$ through p in $O(c^2 \log^2 n)$ time.

Proof. Define $F(v)$ as the number of points of S_i above line $\overline{pv}$. As we advance in counter-clockwise order along any CH peeling layer, $F(v)$ is unimodal. $F(v)$ can be computed using the method in Proposition 3. By Fibonacci searching [4] $F(v)$, we can find the edge of P_i^0 that the bisector of S_i through

*Department of Computer Science, Tufts University, Medford, MA 02155. {mburr,jhugg,erafalin,kseyboth,dls}@cs.tufts.edu. Partially supported by NSF grant CCF-0431027. For full version, see [3].

[1]The *convex hull peeling depth* [5, 2] of a point X_k with respect to a data set $\mathcal{S} = \{X_1, \cdots X_n\}$ in $\mathbb{R}^d$ is the level of the convex layer to which X_k belongs. The points on the outer CH of $\mathcal{S}$ are designated layer one and the points on the kth layer are the points on the CH of the set $\mathcal{S}$ after the points on all previous layers were removed.

p crosses in $O(c \log^2 n)$ time. We apply the procedure to each level, finding the edge that the bisector crosses and eventually finding the two points a, b of S_i such that any line between $\overline{p,a}$ and $\overline{p,b}$ is a legal bisector. □

Theorem 1. The HSC of S_1, S_2 can be found in $O(c^3 \log^3 n)$ time.

Proof. Consider l, a line outside the CH of the union of both point sets S_1, S_2 (*a designated line*). For clarity, assume that l is the x-axis and is below both point sets. For any real number x let $f_i(x)$ be the slope of the line passing through point $p = (x, 0)$ on l that bisects S_i. Define $f(x) = f_1(x) - f_2(x)$. Assume h is a HSC of the two point-sets that intersects l in point $p_h = (x_h, 0)$. Then $f_1(x_h) = f_2(x_h)$ (see Fig. 2).

A degenerate line may intersect any number of HSCs, including 0. We guarantee that the algorithm finds at least one intersection point of a HSC with a designated line by considering a set of 3 designated lines l_1, l_2, l_3 that form a triangle enclosing the point sets and together capture every possible slope of the HSC.

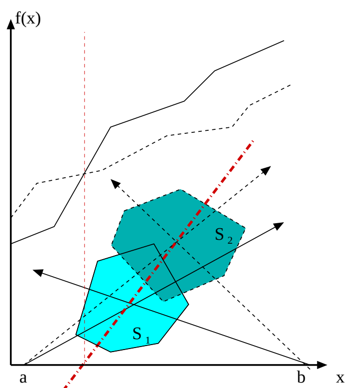

Figure 2: Two convex regions representing two point sets S_1, S_2 and their bisectors, represented using solid and dashed lines respectively. The two monotonically increasing lines represent functions $f_1()$ and $f_2()$, showing the slopes of the bisectors as a function of their intersection point with the x-axis. The two lines intersect in one point, corresponding to the HSC of the two sets.

Given a vertex of S_1 the method in Lemma 1 is used to find a line h through that vertex that bisects S_1 and its intersection point $p = (x, 0)$ with l. Now the method of Lemma 1 is applied again to find the line m passing through p and bisecting S_2 to compute $f(x)$.

Consider the bisectors of set S_i passing through vertices on a CH peeling layer P_i^j. If the vertices are clockwise along P_i^j then the bisectors' intersection points with l have a monotonically decreasing intercept. Since the intersection of two lines that bisect S_i lies inside the CH of S_i (see also [8]) the slopes of the bisectors are increasing monotonically (Fig. 2).

Consider a segment $[a, b]$ on l that intersects a HSC of S_1, S_2. Then $f(a)$ and $f(b)$ change signs. By Fibonacci search on $f()$ along l, we can limit the region of l that intersects the HSC. To bound the running time our algorithm uses the vertices of the CH peeling layers of S_i to constrain the points of l that are visited during the search. The search is conducted for every CH peeling layer separately, total of c times.

The total time complexity of finding the edge on the outermost CH peeling layer of S_1 is therefore $O(c^2 \log^3 n)$. The process is repeated $O(c)$ times to find the correct edges on every CH peeling layer, total of $O(c^3 \log^3 n)$. □

The HSC of the Area and Perimeter [8, 1]): Each vertex v_i in the data structure stores (1) the signed area of the trapezoid defined by edge (v_i, v_{i-1}) and (2) the length of (v_i, v_{i-1}). Each inner node stores the sum of the measures of leaves in its subtree. This enables the computation of the measure associated with each continuous chain in $O(\log n)$ time.

References

[1] T. Abbott, E. D. Demaine, M. L. Demaine, D. Kane, S. Langerman, J. Nelson, and V. Yeung. Dynamic ham-sandwich cuts of convex polygons in the plane. In *Proc. of CCCG'05*, pages 61–64, 2005.

[2] V. Barnett. The ordering of multivariate data. *J. Roy. Stat. Soc. Ser. A*, 139(3):318–355, 1976.

[3] M. Burr, J. Hugg, E. Rafalin, K. Seyboth, and D. L. Souvaine. Dynamic ham-sandwich cuts for two point sets with bounded convex-hull-peeling depth. TUFTS-CS technical report tr-2005-7, Tufts University, 2005.

[4] B. Chazelle and D. P. Dobkin. Intersection of convex objects in two and three dimensions. *J. ACM*, 34(1):1–27, 1987.

[5] W. Eddy. Convex hull peeling. In H. Caussinus, editor, *COMPSTAT*, pages 42–47. Physica-Verlag, Wien, 1982.

[6] H. Edelsbrunner. *Algorithms in combinatorial geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, 1987.

[7] C.-Y. Lo, J. Matoušek, and W. Steiger. Algorithms for ham-sandwich cuts. *Discrete Comput. Geom.*, 11(4):433–452, 1994.

[8] I. Stojmenović. Bisections and ham-sandwich cuts of convex polygons and polyhedra. *Inform. Process. Lett.*, 38(1):15–21, 1991.