Entropy-Bounded Computational Geometry Made Easier and Sensitive to Sortedness

David Eppstein, Michael T. Goodrich, Abraham M. Illickan, Claire A. To

University of California, Irvine

The 37th Canadian Conference on Computational Geometry
August 2025

Themes

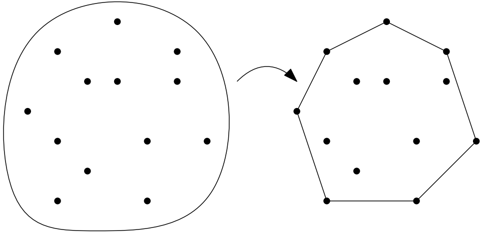
1. Beyond worst-case analysis
2. Adaptive algorithms
3. Entropy bounds
4. Sorting

applied to computational geometry problems

2D Convex Hull

Definition

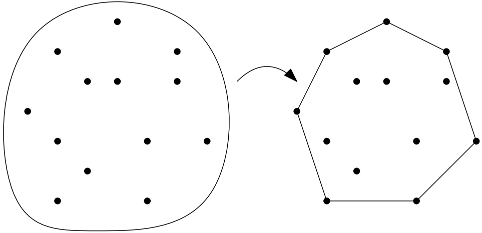
The **convex hull** of a set of points is the smallest convex polygon that contains the set.



Worst Case

Theorem

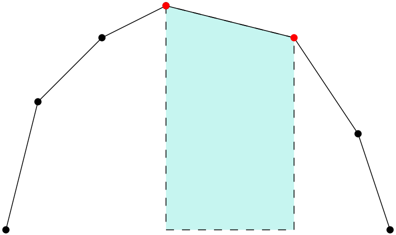
The 2D convex hull of n points can be computed in $\Theta(n \log n)$ time in the worst case.



Output Sensitive

Theorem (Kirkpatrick and Seidel [1], Chan [2])

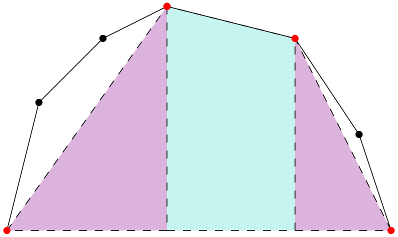
The 2D convex hull of n points can be computed in $\Theta(n \log h)$ time in the worst case, where h is the number of points on the hull.



Entropy-Bounded

Theorem (Afshani, Barbay, Chan [3])

The 2D convex hull of a set S of n points can be computed in $\Theta(n(\mathscr{H}(S)+1))$ time in the worst case, where $\mathscr{H}(S)$ is the structural entropy of S.

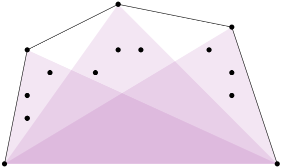


Respectful Partition

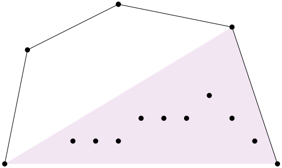
Definition (Afshani, Barbay, Chan [3])

A **partition** Π of S is *respectful* if, for every disjoint subset $S_k \in \Pi$, S_k is either

1. a singleton
2. or can be enclosed by triangle Δ_k whose interior lies under the convex hull of S.



(a) Harder point set.



(b) Easier point set.

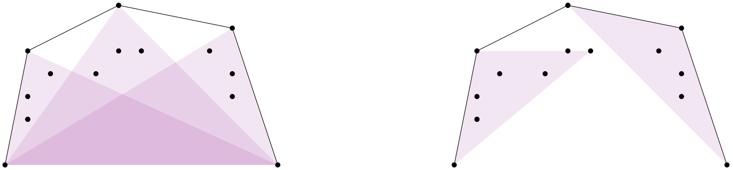
Structural Entropy

Definition

The **entropy** $\mathscr{H}(\Pi)$ of a partition Π is $-\sum \frac{|S_k|}{n} \log \frac{|S_k|}{n}$.

Definition

The **structural entropy** $\mathscr{H}(S)$ of a set S is the minimum entropy over all respectful partitions.



(a) Higher-entropy partition.

(b) Lower-entropy partition.

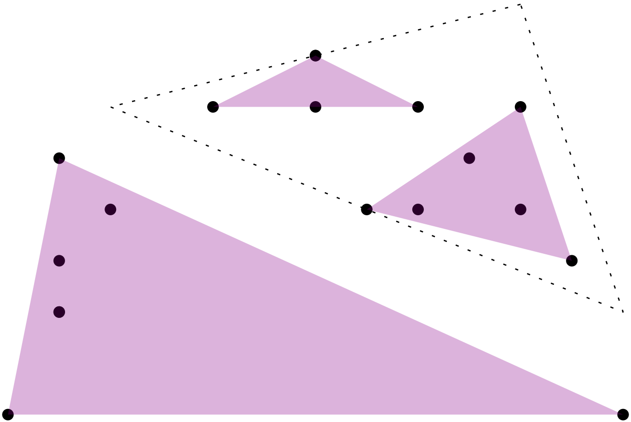
Our Definition

Definition

A **partition** $\Pi = \{(S_1, R_1), \ldots, (S_t, R_t)\}$ of S, where each R_i is a geometric range (<u>triangle</u> for convex hulls), is *respectful* if

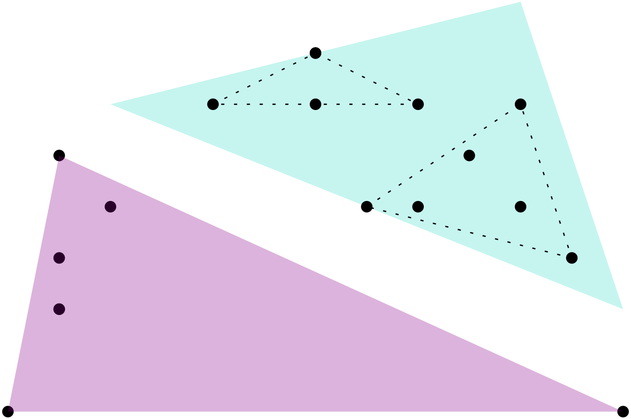
1. for each (S_i, R_i), S_i is contained in R_i, and
    1. S_i forms a sorted subsequence in S,
    2. or R_i lies under the hull of S,
2. and for $i, j = 1, 2, \ldots, t$, if R_i does not lie under the hull of S (which means it is sorted in S), then it will not intersect another range, R_j, where $j \neq i$.

Our Definition (contd.)



Respectful partition without sorted subsets.

Our Definition (contd.)

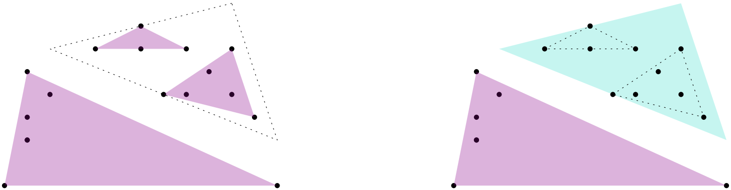


Respectful partition with sorted subsets.

Range-Partition Entropy

Definition

The **range-partition entropy** $\mathscr{H}(S)$ of a set S is the minimum entropy over all respectful partitions.



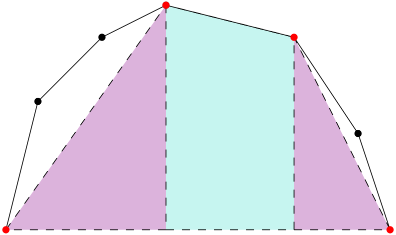
(a) Higher-entropy partition.

(b) Lower-entropy partition.

Our Result

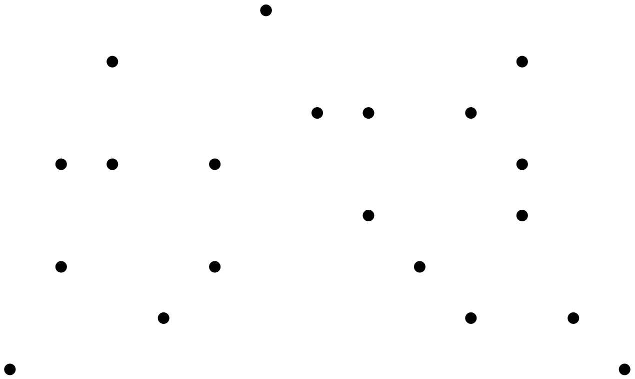
Theorem

The convex hull of a set S of n points can be computed in $\mathcal{O}(n(\mathscr{H}(S)+1))$ time in the worst case, where $\mathscr{H}(S)$ is the range-partition entropy of S.



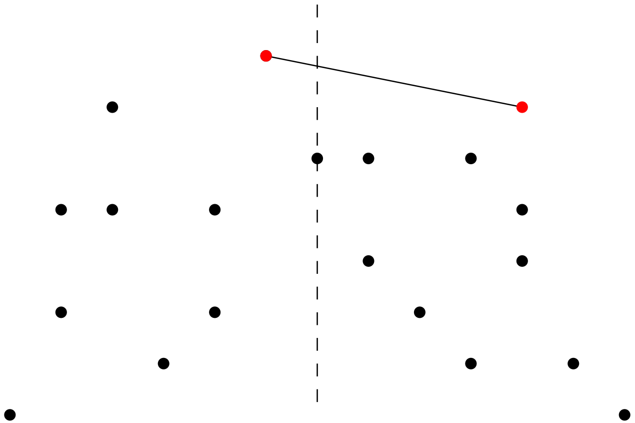
Our Algorithm

Compute the upper hull.



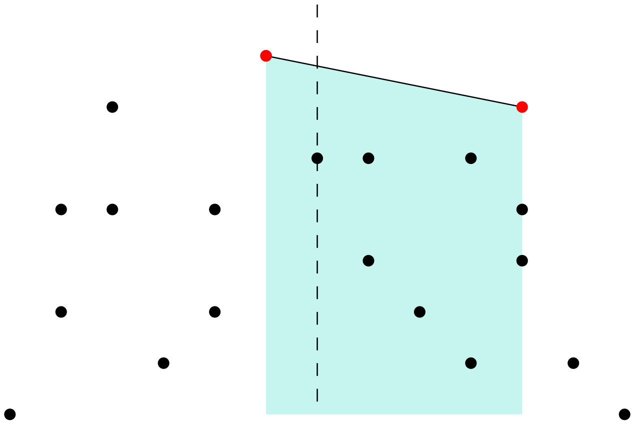
Our Algorithm (contd.)

Divide the points into right and left subsets at the median.
Find the bridge edge intersecting the median line.



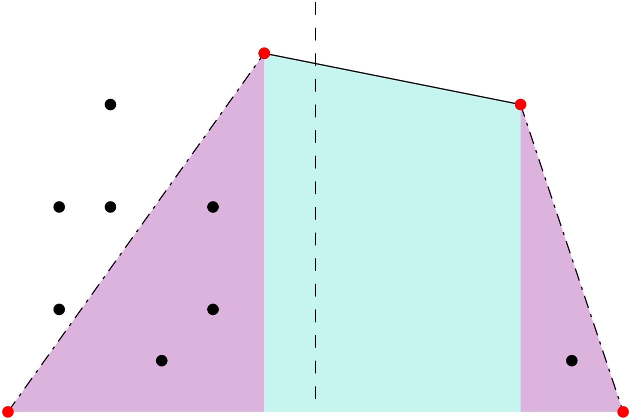
Our Algorithm (contd.)

The points below the bridge edge cannot be in the upper hull.
Prune the points below the edge.



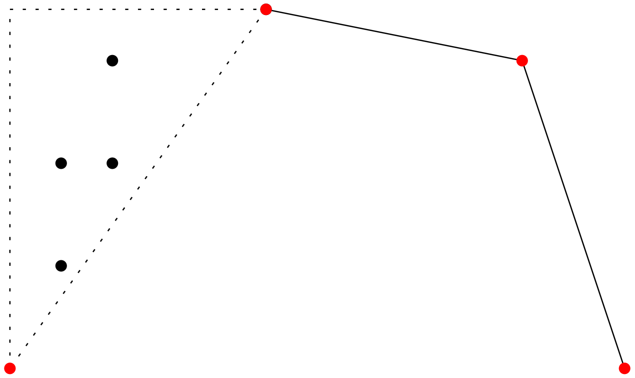
Our Algorithm (contd.)

The points below the edge connecting the right (left)-most point to the bridge's right (left) endpoint cannot be in the upper hull.



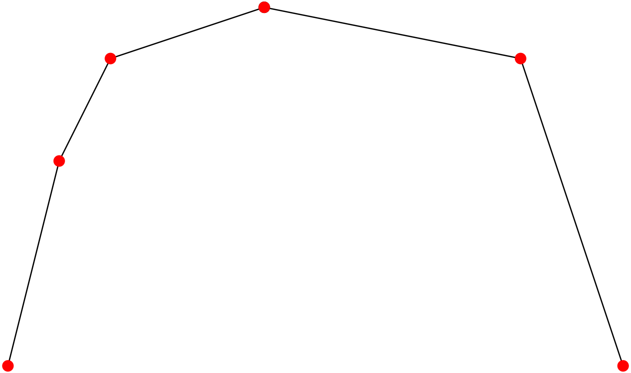
Our Algorithm (contd.)

Check if the left and right subsets are sorted.



Our Algorithm (contd.)

If a subset is sorted, compute the upper hull in linear time [4].
Otherwise, solve that subset recursively.

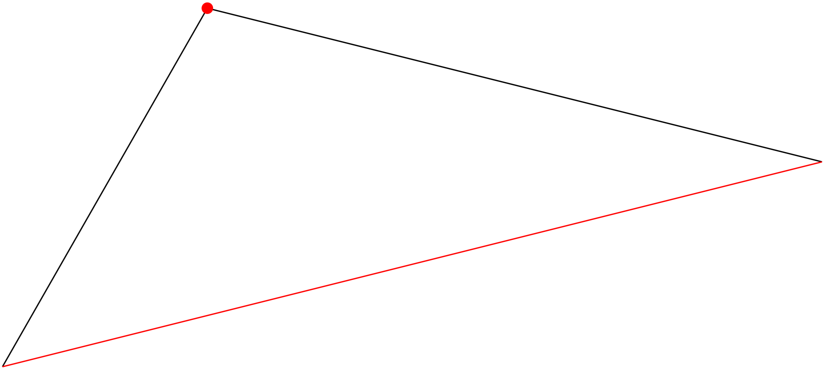


Proof

- Accounting argument.
- Show $S_k \in \Pi$ contributes at most $\mathcal{O}(|S_k|\log(n/|S_k|)+1)$ cyber-dollars.
- Recursion tree T, where each node v is a recursive call.
- Node v contains the x-interval $I_v = [a_v, b_v]$.
- S_k *covers* v if its x-range spans I_v but not $I_{\text{parent}(v)}$.
    1. Case 1: S_k is unsorted.
    2. Case 2: S_k is sorted.
- Cost proportional to $\sum_{j=0}^{\lceil \log n \rceil} n_j$, where n_j total points survive to level j in T.

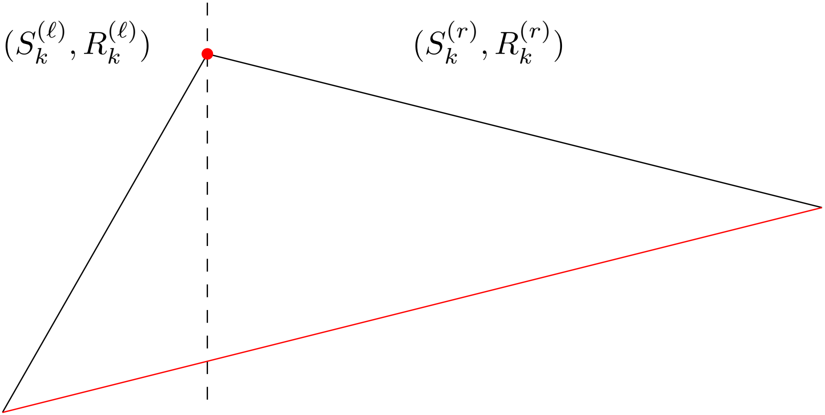
Proof: S_k is Unsorted

Divide S_k at the point across from the lowest edge.



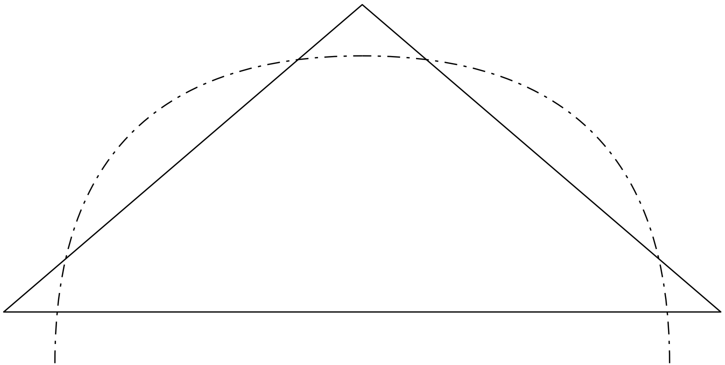
Proof: S_k is Unsorted (contd.)

Both $R_k^{\ell/r}$ are strictly below the hull.
If a node v covers $S_k^{\ell/r}$, a hull edge prunes them.



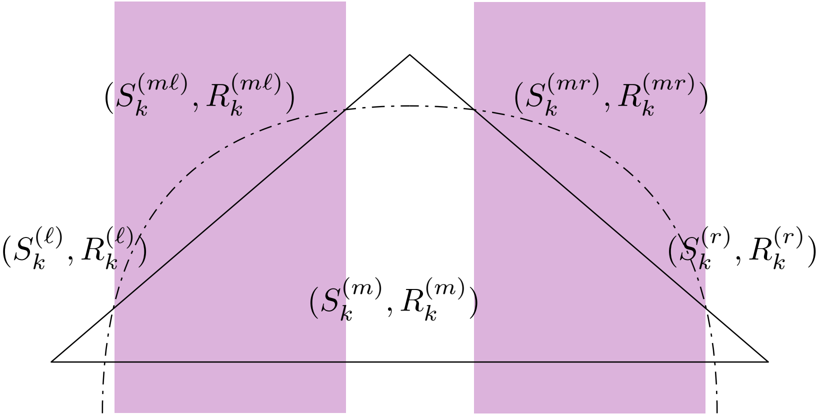
Proof: S_k is Sorted

At most, two sections of the hull can be above R_k.
Divide S_k into five parts at hull-edge intersections.



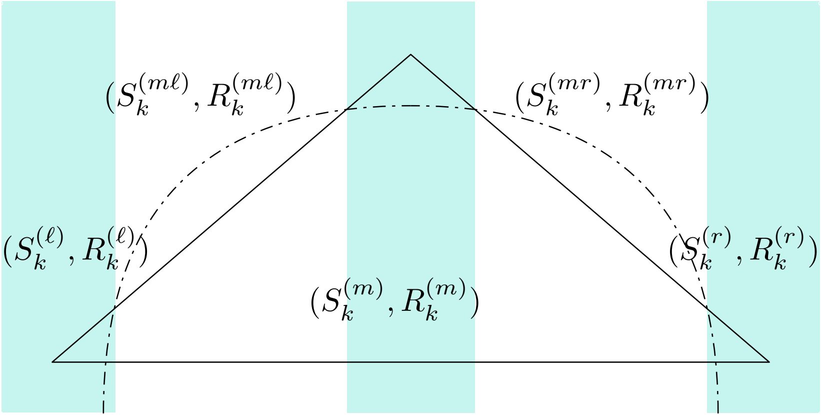
Proof: S_k is Sorted (contd.)

If a node v covers $S_k^{(m\ell)/(mr)}$, a hull edge prunes them.

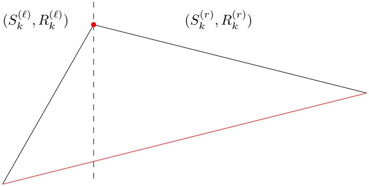

$(S_k^{(m\ell)}, R_k^{(m\ell)})$
$(S_k^{(mr)}, R_k^{(mr)})$
$(S_k^{(\ell)}, R_k^{(\ell)})$
$(S_k^{(r)}, R_k^{(r)})$
$(S_k^{(m)}, R_k^{(m)})$

Proof: S_k is Sorted (contd.)

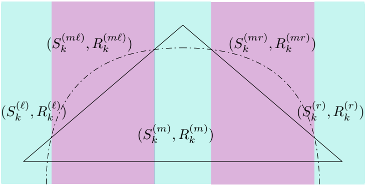
If a node v covers $S_k^{(\ell)/(m)/(r)}$, no points lie above or below. The subsets above the hull are solved in linear time.


$(S_k^{(m\ell)}, R_k^{(m\ell)})$
$(S_k^{(mr)}, R_k^{(mr)})$
$(S_k^{(\ell)}, R_k^{(\ell)})$
$(S_k^{(r)}, R_k^{(r)})$
$(S_k^{(m)}, R_k^{(m)})$

Proof



(a) Case 1: S_k is unsorted.


$(S_k^{(m\ell)}, R_k^{(m\ell)})$
$(S_k^{(mr)}, R_k^{(mr)})$
$(S_k^{(\ell)}, R_k^{(\ell)})$
$(S_k^{(r)}, R_k^{(r)})$
$(S_k^{(m)}, R_k^{(m)})$

(b) Case 2: S_k is sorted.

Proof by counting argument on the recursion tree.
The points in S_k that survive to level j is $\mathcal{O}(\min\{|S_k|, \lceil \frac{n}{2^j} \rceil\})$.

$$\sum_{j=0}^{\lceil \log n \rceil} n_j \le \sum_{j=0}^{\lceil \log n \rceil} \sum_{k} \min\left\{|S_k|, \mathcal{O}\left(\frac{n}{2^j}\right)\right\}$$

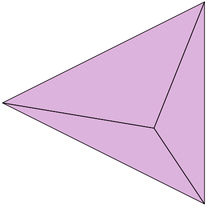
$$\in \mathcal{O}(n(\mathscr{H}(\Pi) + 1))$$

3D Convex Hull

There is no notion of sortedness in 3D, so the range-partition entropy is equivalent to the structural entropy.

Definition

A **partition** Π of S is *respectful* if for each $S_k \in \Pi$, there is a tetrahedron Δ_k that contains all the points of S_k and lies under the 3D convex hull of S.



Previous Result

Theorem (Afshani, Barbay, Chan [3])

The 3D convex hull of a set S of n points can be computed in $\Theta(n(\mathscr{H}(S)+1))$ time in the worst case, where $\mathscr{H}(S)$ is the structural entropy of S.

Theorem (Matoušek [5])

Matoušek's partition theorem runs in $\mathcal{O}(n)$ time.

Theorem (Yao, Dobkin, Edelsbrunner, Paterson [6])

Recursive use of eight-sectioning theorem runs in $\mathcal{O}(n^6 \log n)$ time.

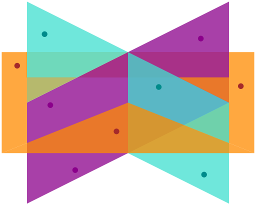
Our Result

Theorem (Afshani, Barbay, Chan [3])

The 3D convex hull of a set S of n points can be computed in $\Theta(n(\mathscr{H}(S)+1))$ time in the worst case, where $\mathscr{H}(S)$ is the range-partition entropy of S.

Lemma

We can find a partition points into eight regions, called octants, containing roughly the same number of points in $\mathcal{O}(n)$.

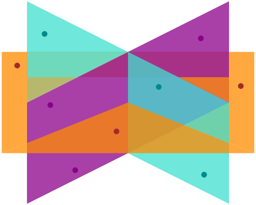


Our Subroutine

We simplify the recursive eight-sectioning subroutine.

1. Randomly sample $n^{\frac{1}{10}}$ points.
2. For every 3 points, define a plane $\Rightarrow \mathcal{O}(n^{\frac{3}{10}})$ planes.
3. For every 3 planes, define a partition $\Rightarrow \mathcal{O}(n^{\frac{9}{10}})$ partitions.
4. Searching $\mathcal{O}(n^{\frac{9}{10}})$ partitions, each $\mathcal{O}(n^{\frac{1}{10}}) \Rightarrow \mathcal{O}(n)$ time.

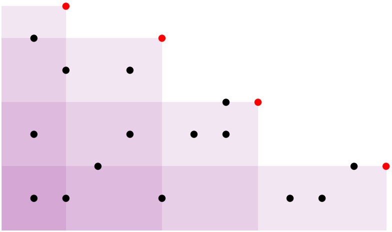
Achieves $\Theta(n(\mathscr{H}(S)+1))$ time to compute the 3D convex hull.



Maxima Set

Definition

The **maxima set** are the points that are not *dominated* (greater x- and y-coordinates) by any other point.

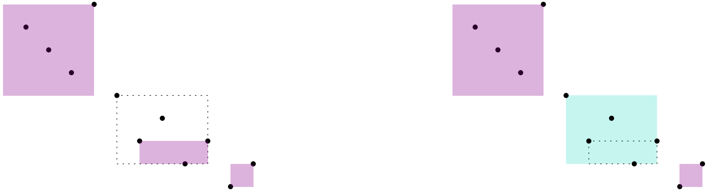


The red points are the maxima set.

Our Result

Theorem

The maxima set of a set S of n points can be computed in $\mathscr{O}(n(\mathscr{H}(S)+1))$ time in the worst case, where $\mathscr{H}(S)$ is the range-partition entropy of S.



(a) Ignores sorted subsets.

(b) Leverages sorted subsets.

Subsets are enclosed in axis-aligned rectangles.

Sorting Entropy

Definition

The **entropy** $\mathscr{H}(S)$ of a sequence S of n elements is

$$-\sum \frac{|S_i|}{n} \log \frac{|S_i|}{n},$$

where each S_i is the i-th monotone run decomposition of S.

$$S = (\underbrace{12, 10, 6, 4,}_{\text{first run}} \ \underbrace{7, 9, 14}_{\text{second run}}, \ \underbrace{0, 3, 5, 11, 16,}_{\text{third run}} \ \underbrace{15, 13, 8, 2, 1}_{\text{fourth run}})$$

$$\mathscr{H}(S) = -(\tfrac{4}{17} \log \tfrac{4}{17} + \tfrac{3}{17} \log \tfrac{3}{17} + \tfrac{5}{17} \log \tfrac{5}{17} + \tfrac{5}{17} \log \tfrac{5}{17})$$

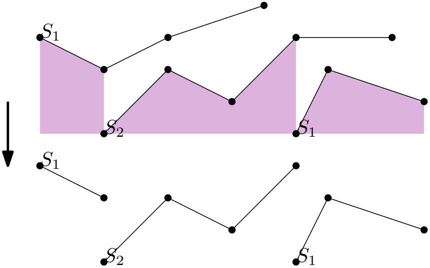
Sorting has a lower bound of $\Omega(n(\mathscr{H}(S) + 1))$.
Range-partition entropy subsumes sorting entropy.

Lower Envelope

Definition

The **lower envelope** is the point-wise minimum of a finite set of functions.

We focus on non-crossing x-monotone chains.

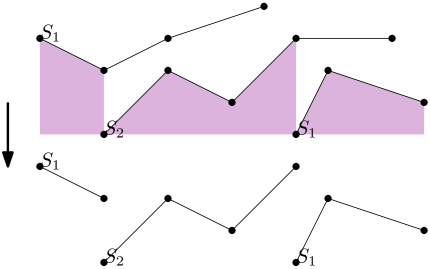


Merging two lower envelopes.

Our Result

Theorem

The lower envelope of a set S of n line segments partitioned into disjoint monotone chains can be computed in $\mathcal{O}(n(\mathscr{H}(S)+1))$ time in the worst-case, where $\mathscr{H}(S)$ is the range-partition entropy of S.



Merging two lower envelopes.

Our Algorithm

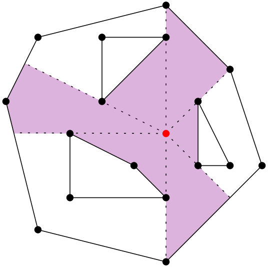
Adapted from stack-based mergesort algorithm, TimSort [7] [8].

- Maintain a stack of maximal non-decreasing runs.
- Merge consecutive sequences by linear scan like mergesort.
- Merge conditions based on their *weight* (total size) leads to a complexity of $\mathcal{O}(n(\mathcal{H}(S) + 1))$ time.

Visibility Polygon

Definition

The **visibility polygon** is the region of a plane that can be seen from a specific point, given a set of obstacles.

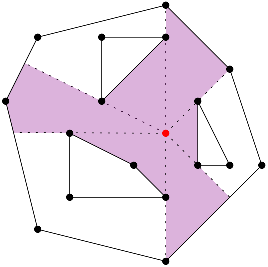


Visibility polygon of convex obstacles.

Our Result

Theorem

Given a collection S of disjoint convex chains of n vertices, representing obstacles, the visibility polygon can be computed in $O(n(\mathscr{H}(S)+1))$ time in the worst case, where $\mathscr{H}(S)$ is the range-partition entropy of S.



Visibility polygon of convex obstacles.

Summary

Range-partition entropy extends the existing measure of structural entropy to capture both structure and sortedness.

New entropy-bounded algorithms:

1. Convex hull
2. Maxima set
3. Lower envelope
4. Visibility polygon

Our framework is a more powerful and general approach to adaptive geometric algorithms.

References

[1] D. G. Kirkpatrick and R. Seidel, “The ultimate planar convex hull algorithm?,” SIAM Journal on Computing, vol. 15, no. 1, pp. 287–299, 1986.

[2] T. M. Chan, “Optimal output-sensitive convex hull algorithms in two and three dimensions,” Discrete & Computational Geometry, vol. 16, no. 4, pp. 361–368, 1996.

[3] P. Afshani, J. Barbay, and T. M. Chan, “Instance-optimal geometric algorithms,” Journal of the ACM, vol. 64, pp. A3:1–A3:38, Mar. 2017.

[4] R. L. Graham, “An efficient algorith for determining the convex hull of a finite planar set,” Information Processing Letters, vol. 1, no. 4, pp. 132–133, 1972.

References (contd.)

[5] J. Matoušek, "Efficient partition trees.," Discrete & Computational Geometry, vol. 8, pp. 315–334, 1992.

[6] F. F. Yao, D. P. Dobkin, H. Edelsbrunner, and M. S. Paterson, "Partitioning space for range queries," SIAM Journal on Computing, vol. 18, no. 2, pp. 371–384, 1989.

[7] T. Peters, "listsort.txt," March 21, 2024.

[8] N. Auger, V. Jugé, C. Nicaud, and C. Pivoteau, "On the worst-case complexity of TimSort," 2019.
Previously announced at ESA 2018.

$$+h\{a_n\}^k\ \gamma 0\cup$$