Incremental (Combinatorial) Voronoi Diagrams

Boris Zolotov, Mag. 1

Modern Methods in Computer Science

February 11, 2020

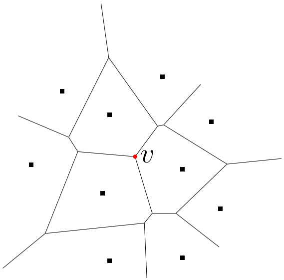
Voronoi Diagrams

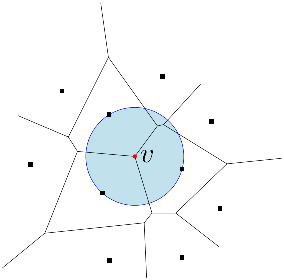
Definition

Voronoi diagram of set $S \subset \mathbb{R}^2$ of n points is the subdivision of the plane into n *cells*, one for each site in S, with the property that a point q lies in the cell corresponding to a site s_i if and only if

$$\mathrm{dist}(q, s_i) < \mathrm{dist}(q, s_j)$$

for each $s_j \in S$ with $j \neq i$.





Algorithms for Voronoi Diagram

1. Calculate whole Voronoi diagram — $O(n \log n)$: sweep line, divide-and-conquer.
2. (And no better: sorting reduces to Voronoi.)
3. Update the diagram when a new site came — $O(n)$ — *literally any* possible way.
4. (And no better: *should draw en example here.* There can be that many changes.)
5. But what if we consider *the graph* of a diagram...

Results today

1. $O(\sqrt{n})$, $\Omega(\sqrt{n})$ edge insertions / removals — even if the diagram is a tree.

   This is in contrast to $O(n)$ geometric changes and $O(\log n)$ (amortised, existential) combinatorial changes in case of inserting in clockwise order.

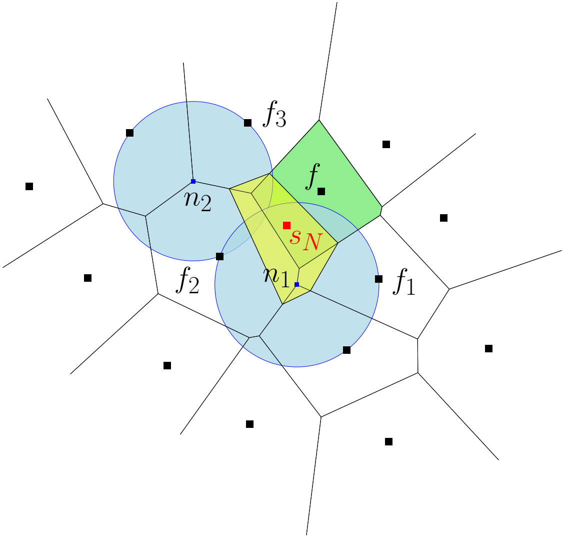
2. *Algorithm* for insertion of sites in convex position, in arbitrary order — $O(\sqrt{n}$ polylog).

Links and Cuts

Link is the addition of an edge, *cut* is the removal of an edge. We count only them. All other operations are thought to have no cost. For example, addition of a vertex.

How many links / cuts are there in our example?

Geometric vs Combinatorial Changes



Chan's structure

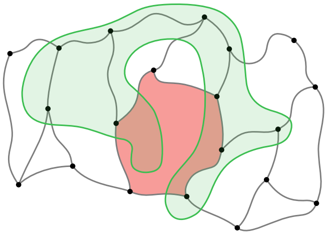
Can be used for:

1. Search of nearest neighbor;
2. Reporting extreme point in given direction.

Makes use of *shallow cuttings.* (Was presented at CG seminar.) Time complexity:

1. Insertion: $O(\log^3 n)$;
2. Deletion: $O(\log^7 n)$ (improved to ^{5});
3. Query: $O(\log^2 n)$.

Flarb



(a) This curve is not flarbable (b) This curve is flarbable (c) Result of applying flarb operation

Fleeq-edges: those intersecting $\mathcal{C}$. Note that in a real Voronoi diagram there can be no such cells as g.

Flarb: Results

Theorem

$\mathcal{G}(G, \mathcal{C})$ has at most two more vertices than G (3-regular graph inside).

Theorem

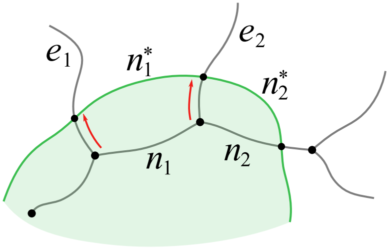
$\mathcal{G}(G, \mathcal{C})$ contains one more face than G (if it comes to Voronoi diagrams).

Theorem

For any new point, there exists a curve such that the changes the graph of V. D. undergoes can be represented by Flarb.

Preserving operation

Sometimes no links and cuts are needed.



$\mathcal{P}(G, \mathcal{C})$ — the set of *preserved* faces. $\mathcal{B}(G, \mathcal{C})$ — faces wholly inside $\mathcal{C}$.
$\mathcal{A}(G, \mathcal{C})$ — augmented, $\mathcal{S}(G, \mathcal{C})$ — shrinking.

Combinatorial Cost of Flarb

$\textsc{cost}$ is how many links / cuts are to be made.

Lemma 2.6 *For a flarbable curve $\mathcal{C}$,*

$$\begin{aligned}(|\mathcal{E}_{\mathcal{C}}| + |\mathcal{B}(G,\mathcal{C})| - |\mathcal{P}(G,\mathcal{C})|)/2 &\le \textsc{cost}(G,\mathcal{C}) \\ &\le 4|\mathcal{E}_{\mathcal{C}}| + 3|\mathcal{B}(G,\mathcal{C})| - 4|\mathcal{P}(G,\mathcal{C})|.\end{aligned}$$

Corollary 2.8 *For a flarbable curve $\mathcal{C}$, it holds that*

$$\textsc{cost}(G,\mathcal{C}) \le 12|\mathcal{S}(G,\mathcal{C})| + 3|\mathcal{B}(G,\mathcal{C})| + O(1).$$

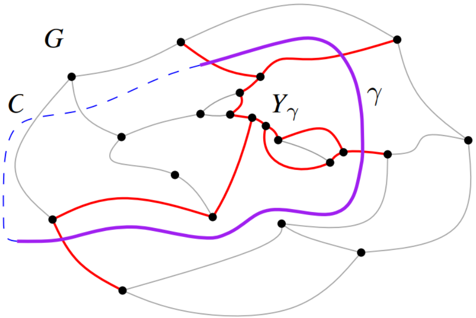
Potential functions

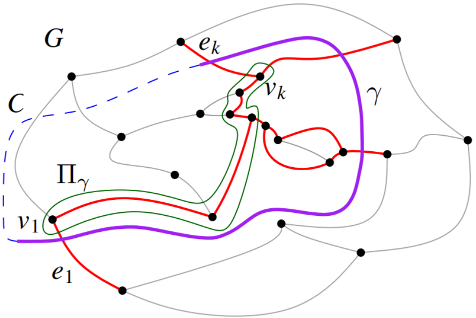
Amortized complexity analysis requires a potential function *and maybe a couple examples.*

1. Local:
$$\mu(f) = \min\left\{\left\lceil\sqrt{|V|}\right\rceil, |f|\right\}.$$
2. Global:
$$\Phi = \lambda \cdot \sum_{f} \mu(f).$$

Really big faces don't change potential.
λ will be later set to 24.

Flarable Sub-Curves





Corollary 3.3 *The graph Y_γ consists of exactly $2k + \delta_2 + 3|H_\gamma| - 3$ edges.*

Lemma 3.4 *The path Π_γ has length at most $k + 3|H_\gamma| + \delta_2 - a - s_c$.*

Shrinking Faces

It turns out that the variation in size of the $\mathcal{C}$-faces after a flarb depends only on the number of shrinking faces.

Theorem 3.5 *Given a flarbable curve $\mathcal{C}$ on G and a flarbable sub-curve γ crossing the fleeq-edges $\epsilon = e_1, \ldots, e_k$, let $f_1, \ldots, f_k$ be the sequence of γ-faces and let $f_1', \ldots, f_k'$ be their corresponding modified faces after the flarb $\mathcal{F}(G, \gamma)$. Then*

$$\sum_{i=1}^{k}(|f_i| - |f_i'|) \geq |\mathcal{S}(G, \gamma)|/2. \tag{1}$$

Flarbable Sequences

Theorem 3.6 *For a 3-regular planar graph $G = (V, E)$ and some flarbable sequence $\mathscr{C} = \mathcal{C}_1, \ldots, \mathcal{C}_N$ of flarbable fleeqs, for all $i \in [N]$,*

$$\textsc{cost}(\mathcal{G}^{i-1}, \mathcal{C}_i) + \Phi(\mathcal{G}^i) - \Phi(\mathcal{G}^{i-1}) \leq O(\sqrt{|V_i|}),$$

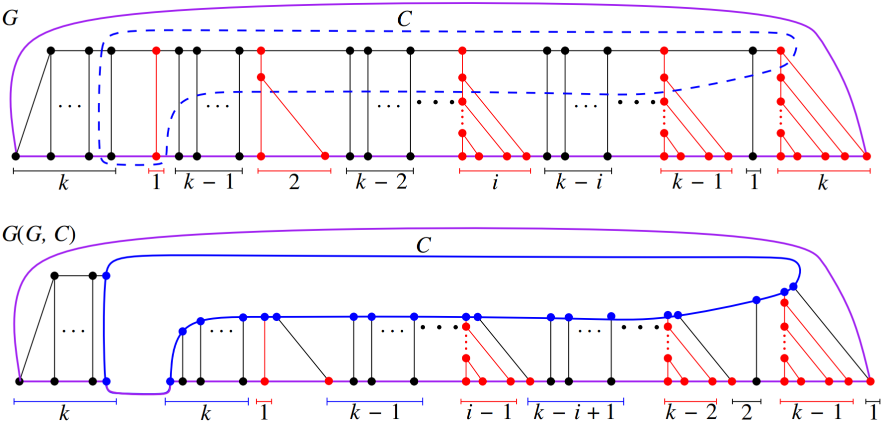
where V_i is the set of vertices of $\mathcal{G}^i$.

Corollary 3.7 *Let G be a 3-regular plane graph with v vertices. For a sequence $\mathscr{C} = \mathcal{C}_1, \ldots, \mathcal{C}_N$ of flarbable fleeqs for graph $G = (V, E)$ where $v = |V|$,*

$$\sum_{i=1}^{N} \textsc{cost}(\mathcal{G}^{i-1}, \mathcal{C}_i) = O(v + N\sqrt{v + N}).$$

So this is really the amortized bound we needed.

Flarbable Sequences



Flarb produces a graph isomorphic to G and has cost $\Omega(\nu)$.

Grappa Tree

A structure designed to store a forest supporting the following operations:
Make-Tree, Link, Cut, Evert, Left-Mark, Right-Mark, Oracle-Search.
Each of them is performed in $O(\log n)$.

Uses heavy-path decomposition.

Voronoi Diagram is stored as a Grappa tree with face-markers
on each edge: cells of which sites are to the right / left from it.

Procedure

1. Evert the tree so that the root is at infinity,
2. Identify portion of each heavy path inside $\textsc{cell}(q, S_{\mathsf{new}})$,
3. Within each path find non-preserved edges,
4. Remove all non-preserved edges,
5. Link back the resulting components in the right order.

Dynamic Circle Reporting Structure

We need to report all the circles from the set containing a given point.
Lift the circles to planes

$$(x, y) \mapsto (x, y, x^2 + y^2).$$

Using point-plane duality in $\mathbb{R}^3$ the circle-reporting can be reduced to extreme-point query that Chan's structure is originally designed for.

The DCR has far-going applications. We store roots of heavy paths Ψ in this structure.

Finding Non-Preserved Edges

For each root $r \in \Psi_q$ (returned by the DCR) *transition edge* h_r is computed in $O(\log n)$.

Observation 5.7 *Given a 3-regular graph G and a flarbable curve $\mathcal{C}$, if we can test whether a point is enclosed by $\mathcal{C}$ in $O(1)$ time, then we can test whether an edge is preserved in $O(1)$ time.*

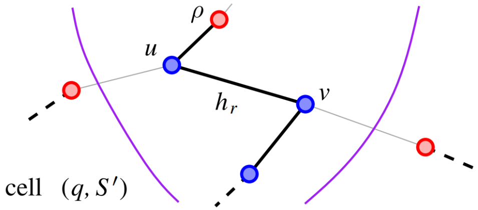
We can now proceed to finding *all* non-preserved edges.

Shadow edges. Bent edges.

$\mathcal{V}_q(S)$ — all the edges of the diagram that intersect the face $\text{CELL}(q, S_{\text{new}})$ of new site.

Shadow edge is an edge among these that is not preserved. We can mark all shadow edges in $O(|\Psi_q| \log n)$. There are also *bent edges* that can't be preserved. They need to be marked.

Path h_r contains two adjacent vertices u and v such that the light edge of u is a left edge while the light edge of v is a right edge. The edge uv cannot be preserved

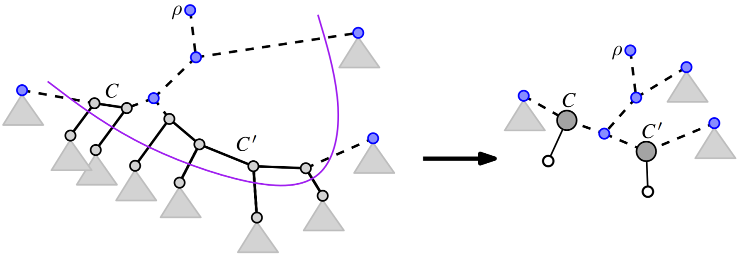


After all, the edge is preserved iff it is not marked as shadow.

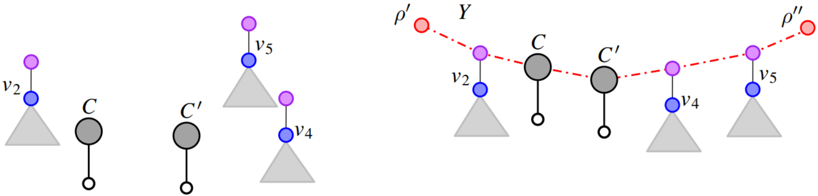
Compressed Tree

$\mathcal{F}$ — forest obtained after *(virtually)* removing all shadow edges. Each component of it is a *comb.* Combs can be large, but we want to have a tour on the graph to see how to re-link it. So we compress each comb into a supernode.

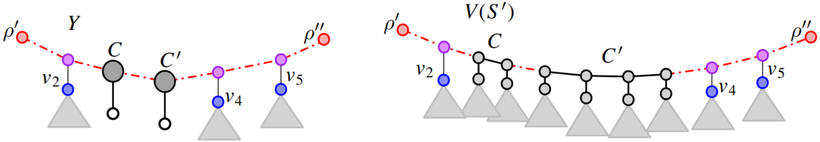
If σ is a number of shadow edges, the compressed tree has $O(\sigma)$ vertices / edges and can be obtained in $O(\sigma \log \sigma)$.



Decompression



Left: an anchor node is created for each isolated leaf of $\mathcal{V}_q(S)$ and attached as its parent. Other isolated nodes are ignored. Right: a super comb is created connecting two new leaves ρ' and ρ'' through a path. This path connects anchor and super-nodes in the order retrieved by the Eulerian tour around the compressed tree



The tree $\mathcal{V}(S')$ achieved after the decompression

Thank you for your attention!

Main points: