Sweep algorithm (Fortune’s algorithm)

- Objective: generating a Voronoi diagram from a set of points in a plane using $O(n \log n)$ time and $O(n)$ space.


- Naïve approach is very expensive. ($O(n^2)$ - Bisector by bisector).

- Previous approaches: Incremental algorithm, Divide & Conquer - both with the same complexity.

- A survey by Bern and Eppstein shows that "D&C" runs the fastest, “sweep” a bit behind and “incremental” the slowest by far. In case of sites more general than points and distance measures other Euclidean, sweeping can be more efficient.

- The approach is based on the line sweep algorithm by Bentley and Ottmann for line segments intersection
  - A vertical line sweeps the plane.
  - Two data-structures are being maintained:
    * $X$: list of sweet-spots (starts with the lines endpoints)
    * $Y$: list of the lines which intersect with the sweeping line sorted by Y value
  - Checks only new neighbours.

- Algorithm
  - A sweep-line ($H$) crosses from left to right.
  - A wavefront ($W$), also known as beach-line, is a connected chain of parabola segments (the ends tend to infinity). Each parabola is the bisector of a point and $H$, $D(p_i, H)$.
  - During the sweep there are two types of events:
    * new wave appears: happens when $H$ reaches a new site (starts from horizontal line and expands).
    * old wave disappears: happens when $H$ reaches the intersection of two adjacent spikes.
  - After all site have been scanned, $V(S)$ is obtained by removing the wavefront and extending all spikes to infinity.
Note: Delaunay Triangulations can be created from $V(S)$ in linear time.

- Lemma 3.4: The size of the wavefront is $O(n)$.

- Proof: Since any two parabolic bisectors $B(p, H), B(q, H)$ can cross at most twice, the size of the wavefront is bounded by $\lambda_2(n) = 2n-1$, where $\lambda$ denotes the maximum length of a Davenport–Schinzel sequence.

- a Davenport–Schinzel sequence is a sequence of symbols in which the number of times any two symbols may appear in alternation is limited. The length of a Davenport–Schinzel sequence is bounded by the number of its distinct symbols multiplied by a small but nonconstant factor that depends on the number of alternations that are allowed.

- Theorem 3.4: Correctness & Complexity

- Proof:

  - At any moment, the diagram to the left of the beachline is correct, as it represent the distances which ignoring the sites right to the sweepline.
  - The sweepline itself is considered as a site, therefore the parabola are created (distance between a line and a point).
  - The wavefront is implemented by a balanced tree, thus inserting/removing is being done in $O(\log n)$.
  - Initially the event queue is populated by the sites in increasing x-coordinate order: $O(n \log n)$.
  - After each update of the wavefront, newly adjacent spikes are tested for intersection. If so the intersection point is added.
  - Since the intersection is a Voronoi vertex, there are only $O(n)$ spike intersections. In addition each site causes an event.
  - For each active spike we need to store only it’s first intersection event, thus the size of the queue never exceeds $O(n)$.
  - Overall we’ve got $O(n)$ operations of $O(\log n)$, thus the complexity is $O(n \log n)$.

Characterization of Voronoi diagrams

- Introduction

  - We shall now address the inverse problem: Given a planer-partition into $n$ convex regions - is it a Voronoi diagram of some sites?
  - Given the sites, the problem is easy to verify (By symmetry in Voronoi, and empty circumcircles in Delaunay).
This problem arises in recognition of biological growth models and the gerrymander problem.

- Lemma 4.1: A convex partition $R_1, \ldots, R_n$ of the plane defines a Voronoi diagram iff there exists a point $p_i$ for each region $R_i$ s.t. $p_i \in R_i$ (Containment), and $\sigma_{ij}(p_i) = p_j$ (Reflection). (Where if $h_{ij}$ is the line containing the common edge, $\sigma_{ij}(p_i)$ is the reflection at $h_{ij}$.)

- Proof:
  - If the partition is a Voronoi diagram then Containment & Reflection follows immediately.
  - Assume that points $p_1, \ldots, p_n$ fulfilling both conditions exist. Take any region $R_i$ and any point $x$ therein.
  - Assume minimum $j$ s.t. $\exists i \neq j, d(x, p_j) < d(x, p_i)$ in order to get a contradiction.
  - Let $R_k$ be the neighbour of $R_j$ which intersects $p_j$; $k = i$ possible.
  - By Convexity $+ (1)$ $h_{jk}$ separates $x$ from $p_j$, thus by (2) we get $d(x, p_k) < d(x, p_j)$ in contradiction.

- Using Lemma 4.1 we reduce the problem to linear programming:
  - Reflection at a line is an affine transformation, so we may write $\sigma_{ij}(x)$ as $A_{ij}x + b_{ij}$, for appropriate matrix $A_{ij}$ and vector $b_{ij}$.
  - Let $R_1, \ldots, R_n$ be a permutation of the regions such that $R_i$ and $R_{i+1}$ are adjacent.
  - Represent each point as $A_{ij}(C_i x + d_i) + b_{ij} = C_j x + d_j$:
    - $p_1 = x$
    - $p_2 = A_{12}x + b_{12} := C_2 x + d_2$
    - $p_3 = A_{23}(A_{12}x + b_{12}) + b_{23} := C_3 x + d_3$
  - This system has at most $2n-2$ equations by lemma 2.3 ($v \leq 2n-2$ by Euler’s formula, or $x$, $y$ for each point expect the first)
  - If there is no solution, construction of a Voronoi diagram is impossible.
  - If there is a unique solution, we get the coordinates of the first candidate site $p_i = x$. The corresponding other sites are obtained simply by reflection. It remains to test these sites for containment in their regions (Checking if each point is in the correct sides of the lines).
  - Setting up the system, solving it, and testing for containment can be accomplished in time $O(n)$ by standard methods.
  - The solution space of the linear system above may have dimension 1 or 2, if so we should apply LP.
Consider each region $R_i$ as the intersection of all half planes bounded by the lines $h_{ij}$. Then $p_i \in R_i$ gives a set of inequalities of the form $p_i^T h_{ij} \leq a_{ij}$.

- plugging in $p_i = C_i x + d_i$ yields $(C_i^T h_{ij}) x \leq a_{ij} - d_i^T h_{ij}$.
- Since we deal with a linear program with $O(n)$ constraints and of constant dimension (actually two), also only linear time.

Theorem 4.1: Let $C$ be a partition of 2-space into $n$ convex regions, given by half-planes supporting the regions and by adjacencies among regions. $O(n)$ time suffices for deciding whether $C$ is a Voronoi diagram, and also for restoring a suitable set of sites in case of its existence.

Optimization properties of Delaunay triangulations

- Let’s look at $DT(S)$ as a triangulation per se and concentrate on parameters which are optimized by $DT(S)$ over all possible triangulations of the point set $S$.
- Recall that a triangulation $T$ of $S$ is a maximal set of non-crossing line segments spanned by the sites in $S$.
- $T$ is locally Delaunay if for each of its convex quadrilaterals $Q$, the corresponding two triangles have circumcircles empty of vertices of $Q$.
- $DT(S)$ is locally Delaunay because all circumcircles for its triangles are empty of sites.
- Theorem 4.2: If a triangulation of $S$ is locally Delaunay then it equals $DT(S)$.
- Proof:
  - Let $T$ be a triangulation of $S$ and assume that $T$ is locally Delaunay.
  - We show that, for each triangle $\triangle$ of $T$, its circumcircle $C(\triangle)$ is empty of sites in $S$.
  - Assuming the contrary, let $s \in C(\triangle)$ for some $s \in S$ and some $\triangle \in T$.
  - Observe $s \in \triangle$, Let $e$ be the edge of $\triangle$ closest to $s$.
  - Suppose, w.l.o.g., that $(\triangle, s, e)$ maximizes the angle at $s$ spanned by $e$, for all such triples.
  - Let triangle $\triangle'$ be adjacent to $\triangle$ at $e$, and let $s'$ be the third vertex of $\triangle'$.
  - As $T$ is locally Delaunay, $s' \notin C(\triangle)$, hence $s \neq s'$.
  - Observe $s \in C(\triangle')$, and let $e'$ be the edge of $\triangle'$ closest to $s$. The angle at $s$ spanned by $e'$ is larger than that spanned by $e$. 


• An edge flip is called good if – after the flip – the triangulation inside the quadrilateral is locally Delaunay.

• Repeated exchange of diagonals of the same quadrilateral always produces an alternating sequence of good and not-good flips.

• Theorem 4.2 now can be used to prove that $DT(S)$ optimizes various quality measures, by showing that each good flip increases quality. Any sequence of good flips then terminates at the global optimum, the Delaunay triangulation.

• Equiangularity is a prominent quality measure:
  - defined as the sorted list of angles $(\alpha_1, \ldots, \alpha_{3t})$.
  - A triangulation is called equiangular if it possesses lexicographically largest equiangularity among all possible triangulations (MaxMin of the angles).
  - Every good flip increases equiangularity.
  - Lawson called a triangulation locally equiangular if no flip can increase equiangularity. Locally equiangular thus is equivalent to locally Delaunay.
  - Sibson first proved theorem 4.2, showing that locally equiangular triangulations are Delaunay and hence unique.
  - Edelsbrunner observed that $DT(S)$ is equiangular (in the global sense) as the global property implies the local one.
  - Mount and Saalfeld showed that $DT(S)$ can be completed by retaining local equiangularity, in $O(n \log n)$ time.

• Theorem 4.3: Let $S$ be a finite set of sites in 2-space. A triangulation of $S$ is equiangular only if it is a completion of $DT(S)$.

• The equiangular triangulation obviously maximizes the minimum angle of all triangles.

• Recently, it has been observed that several other parameters are optimized by $DT(S)$.

• All the properties listed below can be proved by observing that every good edge flip locally optimizes the respective parameter:
  - Coarseness
    - Measured by the largest minimum-enclosing-circle for each triangle.
    - $DT(S)$ minimizes coarseness among all possible triangulations for $S$.
  - Fatness
* Defined as the sum of inradii (The radius of a triangle’s incircle) of the triangles in $S$.
* Lambert showed that $DT(S)$ maximizes fatness, or equivalently, the mean inradius.

- **Roughness**
  * Given an individual function value (height) $h(p)$ for each site $p \in S$, every triangulation $T$ of $S$ defines a triangular surface in 3-space.
  * Define roughness as follows: $\sum_{\Delta \in T} |\Delta| (\alpha^2 + \beta^2)$, with $|\Delta|$ being the area of $\Delta$, and $\alpha, \beta$ being the slopes of the corresponding triangle in 3-space (In other words, roughness is the integral of the squared gradients).
  * Rippa has shown that roughness is minimum for the surface obtained from $DT(S)$.

- On the negative side, $DT(S)$ in general fails to fulfill optimization criteria similar to those mentioned above, such as minimizing the maximum angle, or minimizing the longest edge.

- The Delaunay triangulation avoids an undesirable property that might be shared by other triangulations:
  - Fix a point $v$ in the plane, called the viewpoint. For two triangles $\Delta$ and $\Delta'$ in a given triangulation, write $\Delta < \Delta'$ if $\Delta$ fully or partially hides $\Delta'$ as seen from $v$ (the in-front/before relation).
  - De Floriani et al. observed that this relation is acyclic if the triangulation is Delaunay.

**Voronoi diagrams and Delaunay triangulations in 3-space**

- **Definitions:**
  - Let $S$ be a set of $n$ point sites in 3-space.
  - The bisector of two sites $p, q \in S$ is the perpendicular plane through the midpoint of the line segment $\overline{pq}$.
  - The region $VR(p, S)$ of a site $p \in S$ is the intersection of half spaces bounded by bisectors, and thus is a 3-dimensional convex polyhedron.
  - The boundary of $VR(p, S)$ consists of facets (maximal subsets within the same bisector), of edges (maximal line segments in the boundary of facets), and of vertices (endpoints of edges).
  - The regions, facets, edges, and vertices of $V(S)$ define a cell complex in 3-space. This cell complex is face-to-face: if two regions have a non-empty intersection $f$, then $f$ is a face (facet, edge, or vertex) of both regions.
• As an appropriate data structure for storing a 3-dimensional cell complex we mention the facet-edge structure in Dobkin and Laszlo.

• Voronoi in 3-space
  - The number of facets of $VR(p, S)$ is at most $n - 1$, at most one for each site $q \in S \setminus p$.
  - Hence, by the Eulerian polyhedron formula, the number of edges and vertices of $VR(p, S)$ is $O(n)$. This shows that the total number of components of the diagram $V(S)$ in 3-space is $O(n^2)$.
  - In fact, there are configurations $S$ that force each pair of regions of $V(S)$ to share a facet, thus achieving their maximum possible number of $\binom{n}{2}$ (Dewdney and Vranch).
  - This fact sometimes makes Voronoi diagrams in 3-space less useful compared to 2-space.
  - On the other hand, Dwyer showed that the expected size of $V(S)$ in d-space is only $O(n)$, provided $S$ is drawn uniformly at random in the unit ball. This result indicates that high-dimensional Voronoi diagrams will be small in many practical situations.

• Delaunay in 3-space
  - In analogy to the 2-dimensional case, the Delaunay tessellation $DT(S)$ in 3-space is defined as the geometric dual of $V(S)$.
  - It contains a tetrahedron for each vertex, a triangle for each edge, and an edge for each facet, of $V(S)$.
  - Equivalently, DT(S) may be defined using the empty sphere property.
  - DT(S) is a partition of the convex hull of S into tetrahedra.

• Among the various proposed methods for constructing $V(S)$ in 3-space, incremental insertion is most intuitive and easy to implement.

• two different techniques for integrating a new site $p$ into $V(S)$:
  - The more obvious method first determines all facets of the region of $p$ in the new diagram, $V(S \cup p)$, and then deletes the parts of $V(S)$ interior to this region.
  - In the dual environment, this amounts to detecting and removing all tetrahedra of $DT(S)$ whose circumspheres contain $p$, and then filling the 'hole' with empty-sphere tetrahedra with $p$ as apex, to obtain $DT(S \cup \{p\})$.
  - A different and numerically more stable approach is similarly to the planar case, after having added a site to the current Delaunay tessellation, certain flips changing the local tetrahedral structure are performed in order to achieve local Delaunayhood.
The existence of such a sequence of flips is less trivial, however. No flipping sequence might exist that turns an arbitrary tetrahedral tessellation for $S$ into $DT(S)$.

Like in 2-space, there are also two ways of tetrahedralizing five sites in convex position, and a flip per definition exchanges them. However, that the flip will replace two tetrahedra by three or vice versa.

This indicates an important difference between triangulations in 2-space and tetrahedral tessellations in 3-space: The number of tetrahedra does depend on the way of tetrahedralizing $S$. It may vary from $\Theta(n)$ to $\Theta(n^2)$.

After having added a site $p \in S$ to the current Delaunay tessellation, the tetrahedron containing $p$ is split into four tetrahedra with apex $p$, in the obvious way. The algorithm first considers the four triangles opposite to $p$, that is, the bases of the tetrahedra with apex $p$.

Generally, each triangle $\Delta$ of the tessellation is shared by two tetrahedra $T$ and $T'$ which, in turn, are spanned by five sites. Three of them span $\Delta$, and the remaining two sites $q$ and $q'$ belong to $T$ and $T'$.

A third tetrahedron might be spanned by these five sites. $\Delta$ is called flippable if the union of these two or three tetrahedra is convex.

For each triangle $\Delta$ opposite to $p$, the algorithm now performs the flip that involves the five sites corresponding to $\Delta$, provided $\Delta$ is flippable and not locally Delaunay (defined like in 2-space).

Thereby, new triangles become opposite to $p$ and possibly have to be flipped, too.

This sequence of flips terminates in the Delaunay tessellation that includes $p$.

The runtime is $O(n^2)$ which is optimal in the worst case.

Still, it might construct quadratically large intermediate tessellations, in spite of the possibly linear size of the final tessellation.