# Line Arrangements

Michael Hoffmann <hoffmann@inf.ethz.ch>

3rd December 2007

## 1 Arrangements

The subdivision of the plane induced by a finite set $L$ of lines is called the **arrangement** $\mathcal{A}(L)$. A line arrangement is **simple** if no two lines are parallel and no three lines meet in a point. Although lines are unbounded, we can regard a line arrangement as a bounded object by (conceptually) putting a sufficiently large box around that contains all vertices. Such a box can be constructed in $O(n \log n)$ time for $n$ lines. (Exercise) Moreover, we can view a line arrangement as a planar graph by adding an additional vertex at "infinity", that is incident to all rays which leave this bounding box.

**Theorem 1** *A simple arrangement $\mathcal{A}(L)$ of $n$ lines in $\mathbb{R}^2$ has $\binom{n}{2}$ vertices, $n^2$ edges, and $\binom{n}{2} + n + 1$ faces/cells.*

**Proof.** Since all lines intersect and all intersection points are pairwise distinct, there are $\binom{n}{2}$ vertices.

The number of edges we prove by induction on $n$. For $n = 1$ we have $1^2 = 1$ edge. By adding one line to an arrangement of $n-1$ lines we split $n-1$ existing edges into two and introduce $n$ new edges along the newly inserted line. Thus, there are in total $(n-1)^2 + 2n - 1 = n^2 - 2n + 1 + 2n - 1 = n^2$ edges.

The number $f$ of faces can now be obtained from Euler's formula $v - e + f = 2$, where $v$ and $e$ denote the number of vertices and edges, respectively. However, in order to apply Euler's formula we need to consider $\mathcal{A}(L)$ as a planar graph and take the symbolic "infinite" vertex into account. Therefore,

$$f = 2 - \left(\binom{n}{2} + 1\right) + n^2 = 1 + \frac{1}{2}(2n^2 - n(n-1)) = 1 + \frac{1}{2}(n^2 + n) = 1 + \binom{n}{2} + n\,.$$

□

The *complexity* of an arrangement is simply the total number of vertices, edges, and faces (in general, cells of all dimension).

## 2 Representation as a DCEL

We need to have a suitable representation for line arrangements that supports a variety of local operations. More specifically, it should allow to access or iterate over ...

- the (two) vertices/faces incident to an edge,
- the edges/faces incident to a vertex,
- the vertices adjacent to a vertex,
- the faces adjacent to a face.

It should also be possible to modify the structure, for instance, to

- insert a new vertex along an edge (*edge split*),
- contract or remove an edge,
- insert a new edge connecting two vertices incident to the same face (*face split*).

We use a general purpose data structure to represent planar subdivisions[1], known as DCEL (doubly connected edge list) or halfedge data structure. It consists of three containers for the three entities of interest: vertices, edges, and faces. The information on how these objects are interconnected are stored at the edges mostly.

Each edge appears actually twice in the data structure, one *halfedge* for each incident face. The pair of halfedges representing the same edge of the original graph are called *twins*. Halfedges are considered oriented such that the incident face is to the left.

Each vertex and each face store a pointer to some incident halfedge. (They may also store/refer to some geometric information, e.g., coordinates of a vertex.)

Each halfedge $e$ stores ...

1. a pointer $e$.target to its target vertex,
2. a pointer $e$.face to the incident face,
3. a pointer $e$.twin to its twin,
4. pointers $e$.prev and $e$.next to the previous and next halfedge in the circular sequence of halfedges around $e$.face (oriented anticlockwise),
5. and possibly additional information. In the case of line arrangements, one may want to store a pointer to the underlying line, for example.

How can one iterate around a face or vertex? (Needs both prev and next. The source, on the other hand, can be obtained as the twin's target in constant time.)

[1] More precisely, orientable (inside/outside/left/right are well defined, no Moebius-strips) 2-manifolds (objects that look like a Euclidean disk locally). It is possible to extend this notion to objects with boundaries.

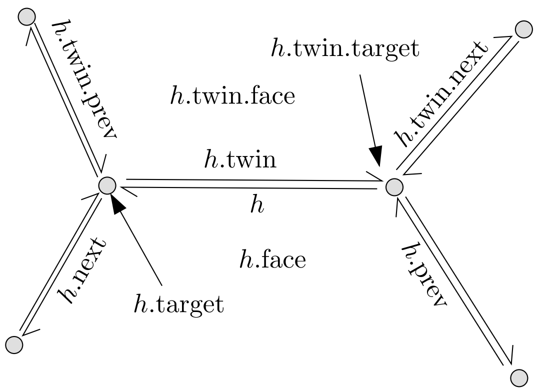


**Figure 1:** *The data associated to a halfedge* $h$.

# 3 Construction

As the complexity of a line arrangement is quadratic, there is no need to look for a subquadratic algorithm to construct it. We will simply construct it incrementally, inserting the lines one by one. Let $\ell_1, \ldots, \ell_n$ be the order of insertion.

At Step $i$ of the construction, locate $\ell_i$ in the leftmost cell of $\mathcal{A}(\{\ell_1, \ldots, \ell_{i-1}\})$ it intersects. (The halfedges leaving the infinite vertex are ordered by slope.) This takes $O(i)$ time. Then traverse the boundary of the face $F$ found until the halfedge $h$ is found where $\ell_i$ leaves $F$. Insert a new vertex at this point, splitting $F$ and $h$ and continue in the same way with the face on the other side of $h$.

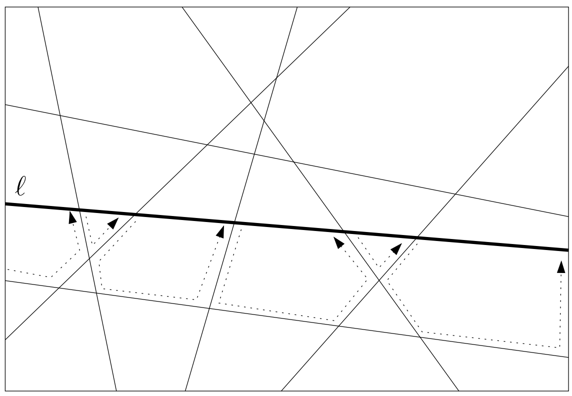


**Figure 2:** *Incremental construction: Insertion of a line* $\ell$.

What is the time needed for this traversal? The complexity of $\mathcal{A}(\{\ell_1, \ldots, \ell_{i-1}\})$ is $\Theta(i^2)$, but we will see that the region traversed by a single line has linear complexity only.

## 4 Zone Theorem

For line $\ell$ and an arrangement $\mathcal{A}(L)$ the **zone** of $\ell$ in $\mathcal{A}(L)$ is the set of faces from $\mathcal{A}(L)$ whose closure intersects $\ell$.

**Theorem 2** *Given an arrangement $\mathcal{A}(L)$ of $n$ lines in $\mathbb{R}^2$ and a line $\ell$ (not necessarily from $L$), the total number of edges in all cells of the zone $Z_{\mathcal{A}(L)}(\ell)$ is at most $6n$.*

**Proof.** Without loss of generality suppose that $\ell$ is horizontal and that none of the lines from $L$ is horizontal. Split the edges of $Z_{\mathcal{A}(L)}(\ell)$ into two groups, left-bounding and right-bounding edges (split each cell at its topmost and at its bottommost vertex). We will show that there are at most $3n$ left-bounding edges by induction on $n$.

For $n = 1$, there is exactly one left-bounding edge in $Z_{\mathcal{A}(L)}(\ell)$ and $1 \leq 3n = 3$. Assume the statement is true for $n - 1$.

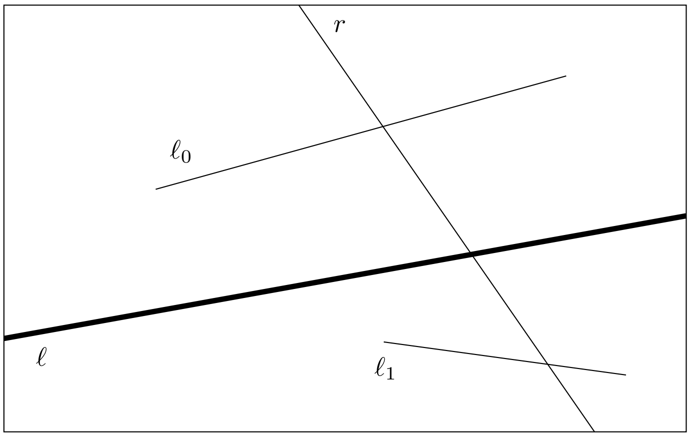


**Figure 3:** *At most three new left-bounding edges are created by adding $r$ to $\mathcal{A}(L \setminus \{r\})$.*

Consider the rightmost line $r$ from $L$ intersecting $\ell$ and the arrangement $\mathcal{A}(L \setminus \{r\})$. By the induction hypothesis there are at most $3n-3$ left-bounding edges in $Z_{\mathcal{A}(L\setminus\{r\})}(\ell)$. Adding $r$ back adds at most three new left-bounding edges: At most two existing left-bounding edges (call them $\ell_0$ and $\ell_1$) of the rightmost cell of the zone are intersected by $r$ and thereby split in two, and $r$ itself contributes one more edge to that cell. The line $r$ cannot contribute a left-bounding edge to any cell other than the rightmost: to the left of $r$, the edges induced by $r$ form right-bounding edges only and to the right of $r$ all other cells touched by $r$ (if any) are shielded away from $\ell$ by one of $\ell_0$ or $\ell_1$. Therefore, the total number of edges in $Z_{\mathcal{A}(L)}(\ell)$ is bounded from above by $3 + 3n - 3 = 3n$. □

**Corollary 3** *The arrangement of $n$ lines in $\mathbb{R}^2$ can be constructed in $O(n^2)$ time and this is optimal.*

Corresponding bounds in $\mathbb{R}^d$: Complexity of arrangements in $\Theta(n^d)$, zone of a hyperplane is $O(n^{d-1})$.

# 5 The Power of Duality

The real beauty and power of line arrangements becomes apparent in context of point $\leftrightarrow$ line duality.

Recall the standard duality transform that maps a point $p = (p_x, p_y)$ to the line $p^* : y = p_x x - p_y$ and a line $g : y = mx + b$ to the point $g^* = (m, -b)$. It is ...

- Incidence preserving: $p \in g \iff g^* \in p^*$.
- Order preserving: $p$ is above $g \iff g^*$ is above $p^*$.
- Another way to think of duality is in terms of the parabola $\mathcal{P} : y = \frac{1}{2}x^2$. For a point $p$ on $\mathcal{P}$, the dual line $p^*$ is the tangent to $\mathcal{P}$ at $p$. For a point $p$ not on $\mathcal{P}$, consider the vertical projection $p'$ of $p$ onto $\mathcal{P}$: the slopes of $p^*$ and $p'^*$ are the same, just $p^*$ is shifted up the difference in $y$-coordinates.

The following problems can all be solved in $O(n^2)$ time and space by constructing the dual arrangement.

**General position test.** Given $n$ points in $\mathbb{R}^2$, are any three of them collinear? (Dual: do three lines meet in a point?)

**Minimum area triangle.** Given $n$ points in $\mathbb{R}^2$, what is the minimum area triangle spanned by any three of them? For any vertex of the dual arrangement (primal: line through two points $p$ and $q$) find the closest point vertically above/below through which an input line passes (primal: closest line above/below parallel to the line through $p$ and $q$ that passes through an input point).

**Ham-sandwich cuts.** Given two finite sets $R$ and $B$ of points, construct a line that bisects both sets, that is, in either open halfplane defined by the line there are no more than $|R|/2$ points from $R$ and no more than $|B|/2$ points from $B$.

This needs the concept of $k$-levels in arrangements (a point is at level $k$ if there are at most $k-1$ lines below and at most $n-k$ lines above it). Suppose wlog that no two points have the same $x$-coordinate. (Otherwise, rotate the plane infinitesimally.) Also suppose wlog that both $|R|$ and $|B|$ are odd. (Otherwise, just remove one point from the set and observe that the bisection as defined above remains valid when adding the point back.)

The median level of $\mathcal{A}(B)$ defines the bisecting lines for $B$ and since $|B|$ is odd, both the leftmost and the rightmost piece of this level are defined by the same line from $B$, the one with median slope. Similarly for $\mathcal{A}(R)$. Since both lines intersect (no two points have the same $x$-coordinate, so no two lines have the same slope), the median levels must intersect as well, at a point whose dual corresponds to a line that bisects both $R$ and $B$.

**Sorting all Angular Sequences.**

**Theorem 4** *Consider a set* $P$ *of* $n$ *points in the plane. For a point* $q \in P$ *let* $c_P(q)$ *denote the circular sequence of points from* $S \setminus \{q\}$ *ordered counterclockwise around* $q$ *(in order as they would be encountered by a ray sweeping around* $q$*). All* $c_P(q)$, $q \in P$*, collectively can be obtained in* $O(n^2)$ *time.*

**Proof.** Consider the projective dual $P^*$ of $P$. An angular sweep around a point $q \in P$ corresponds there to a traversal of the line $q^*$ from left to right. (A collection of lines through a single point $q$ corresponds to a collection of points on a single line $q^*$ and slope corresponds to $x$-coordinate.) Clearly, the sequence of intersection points along all lines in $P^*$ can be obtained by constructing the arrangement in $O(n^2)$ time. In the primal plane, any such sequence corresponds to an order of the remaining points according to the slope of the connecting line; to construct the circular sequence of points as they are encountered around $q$, we have to split the sequence obtained from the dual into those points that are to the left of $q$ and those that are to the right of $q$; concatenating both yields the desired sequence. □

# 6 Segment Endpoint Visibility Graphs

A fundamental problem in motion planning is to find a short(est) path between two given positions in some domain, subject to certain constraints. As an example, suppose we are given two points $p, q \in \mathbb{R}^2$ and a set $S \subset \mathbb{R}^2$ of obstacles. What is the shortest path between $p$ and $q$ that avoids $S$?

**Observation 5** *The shortest path between two points that does not cross a set of polygonal obstacles (if it exists) is a polygonal path whose interior vertices are obstacle vertices.*

One of the simplest type of obstacle conceivable is a line segment. In general the plane may be disconnect with respect to the obstacles, for instance, if they form a closed curve. However, if we restrict the obstacles to pairwise disjoint line segments then there is always a free path between any two given points. Apart from start and goal position, by the above observation we may restrict our attention concerning shortest paths to straight line edges connecting obstacle vertices, in this case, segment endpoints.

**Definition 6** *Consider a set* $S$ *of* $n$ *disjoint line segments in* $\mathbb{R}^2$*. The* **segment endpoint visibility graph** $\mathcal{V}(S)$ *is a plane straight line graph defined on the segments endpoints. Two segment endpoints* $p$ *and* $q$ *are connected in* $\mathcal{V}(S)$ *if and only if*

- *the line segment* $\overline{pq}$ *is in* $S$ *or*
- $\overline{pq} \cap s \subseteq \{p, q\}$ *for every segment* $s \in S$.

If all segments are on the convex hull, the visibility graph is complete. If they form parallel chords of a convex polygon, the visibility graph consists of copies of $K_4$, glued together along opposite edges and the total number of edges is linear only. These graphs are Hamiltonian. :-)

Constructing $\mathcal{V}(S)$ for a given set $S$ of disjoint segments in a brute force way takes $O(n^3)$ time. (Take all pairs of endpoints and check all other segments for obstruction.)

**Theorem 7 (Welzl 1985)** *The segment endpoint visibility graph of $n$ disjoint line segments can be constructed in worst case optimal $O(n^2)$ time.*

**Proof.** We have seen above how all sorted angular sequences can be obtained from the dual line arragement in $O(n^2)$ time. Topologically sweep the arrangement from left to right (corresponds to changing the slope of the primal rays from $-\infty$ to $+\infty$) while maintaining for each segment endpoint $p$ the segment $s(p)$ it currently "sees" (if any). Initialize by brute force in $O(n^2)$ time (direction vertically downwards). Each intersection of two lines corresponds to two segment endpoints "seeing" each other along the primal line whose dual is the point of intersection. In order to process an intersection, we only need that all preceding (located to the left) intersections of the two lines involved have already been processed. This order corresponds to a topological sort of the arrangement graph where all edges are directed from left to right. A topological sort can be obtained, for instance, via (reversed) post order DFS in linear time.

When processing an intersection, there are four cases. Let $p$ and $q$ be the two points involved such that $p$ is to the left of $q$.

1. The two points belong to the same input segment $\rightarrow$ output the edge $pq$, no change otherwise.
2. $q$ is obscured from $p$ by $s(p)$ $\rightarrow$ no change.
3. $q$ is endpoint of $s(p)$ $\rightarrow$ output $pq$ and update $s(p)$ to $s(q)$.
4. Otherwise $q$ is endpoint of a segment $t$ that now obscures $s(p)$ $\rightarrow$ output $pq$ and update $s(p)$ to $t$.

Thus any intersection can be processed in constant time and the overall runtime of this algorithm is quadratic. □

# 7 3-Sum

The 3-Sum problem is the following: Given a set $S$ of $n$ integers, does there exist a three-tuple of elements from $S$ that sum up to zero? By testing all three-tuples this can obviously be solved in $O(n^3)$ time. If the tuples to be tested are picked a bit more cleverly, we obtain an $O(n^2)$ algorithm.

Let $(s_1, \ldots, s_n)$ be the sequence of elements from $S$ in increasing order. Then we test the tuples as follows.

```
For i = 1,...,n-2 {
    j = i+1, k = n.
    While k > j {
        If s_i + s_j + s_k = 0 then exit with triple s_i, s_j, s_k.
        If s_i + s_j + s_k > 0 then k = k-1 else j = j+1.
    }
}
```

The runtime is clearly quadratic (initial sorting can be done in $O(n \log n)$ time). Regarding the correctness observe that the following is an invariant that holds at begin of the inner loop: There exists no suitable triple that contains $s_i$ and $s_\ell$ for any $\ell < j$ or $\ell > k$.