CMSC 754: Lecture 14
Line Arrangements: Basic Definitions and the Zone Theorem

Reading: Chapter 8 in the 4M's.

Line Arrangements: We have studied a number of the most fundamental structures in computational geometry: convex hulls, Voronoi diagrams and Delaunay triangulations. These are all defined over a finite set of points. As we saw earlier, points and lines in the plane are related to each other through the *dual transformation*. In this lecture, we will studey a fundamental structure defined for a finite set of lines, called a *line arrangement*.

Consider a finite set L of lines in the plane. These lines naturally subdivide the plane into a cell complex, which is called the *arrangement* of L, and is denoted $\mathcal{A}(L)$ (see Fig. 1(a)). The points where two lines intersect form the vertices of the complex, the segments between two consecutive intersection points form its edges, and the polygonal regions between the lines form the faces. Although an arrangement contains unbounded edges and faces, as we did with Voronoi diagrams (from a purely topological perspective) it is possible to add a vertex at infinity and attach all these edges to this vertex to form a proper planar graph (see Fig. 1(b)). An arrangement can be represented using any standard data structure for cell complexes, a DCEL for example.

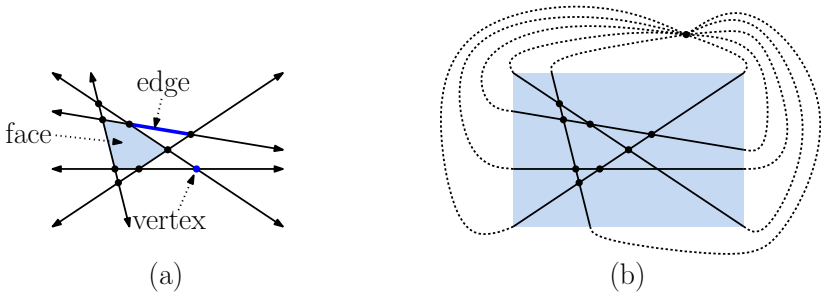

Fig. 1: Arrangement of lines; (a) the basic elements of an arrangement and (b) adding a vertex at infinity to form a proper planar graph.

As we shall see, arrangements have many applications in computational geometry. Through the use of point-line duality, many of these applications involve sets of points. We will begin by discussing the basic geometric and combinatorial properties of arrangements and an algorithm for constructing them. Later we will discuss applications of arrangements to other problems in computational geometry. Although we will not discuss it, line arrangements in $\mathbb{R}^2$ can be generalized to hyperplane arrangements in $\mathbb{R}^d$. In such a case the arrangement is a polyhedral cell complex.

Combinatorial Properties: The *combinatorial complexity* of an arrangement is the total number of vertices, edges, and faces in the arrangement. An arrangement is said to be *simple* if no three lines intersect at a common point. Through our usual general position assumption that no three lines intersect in a single point, it follows that we will be interested only in simple arrangements. We will also assume that no two lines are parallel. The following lemma shows that all of these quantities are $\Theta(n^2)$ for simple planar line arrangements.

Lemma: Let $\mathcal{A}(L)$ be a simple arrangement of n lines L in the plan. Then:

(i) the number of vertices (not counting the vertex at infinity) in $\mathcal{A}(L)$ is $\binom{n}{2} = \frac{1}{2}(n^2-n)$.

(ii) the number of edges in $\mathcal{A}(L)$ is n^2

(iii) the number of faces in $\mathcal{A}(L)$ is $\binom{n}{2} + n + 1 = \frac{1}{2}(n^2+n+2)$.

Proof: The fact that the number of vertices is $\binom{n}{2}$ is clear from the fact that (since no two are parallel) each pair of lines intersects in a single point.

The number of edges follows from the fact that each line contains n lines. This is because each line is cut by each of the other $n-1$ lines (assuming no two parallel lines), which splits the line into n edges.

The number of faces follows from Euler's formula, $v-e+f=2$. To form a cell complex, recall that we added an additional vertex at infinity. Thus, we have $v = 1 + \binom{n}{2}$ and $e = n^2$. Therefore, the number of faces is

$$\begin{aligned} f &= 2 - v + e = 2 - \left(1 + \tbinom{n}{2}\right) + n^2 = 2 - \left(1 + \tfrac{n(n-1)}{2}\right) + n^2 \\ &= 1 + \tfrac{n^2}{2} + \tfrac{n}{2} = 1 + \tfrac{n(n-1)}{2} + n = \tbinom{n}{2} + n + 1, \end{aligned}$$

as desired.

By the way, this generalizes to higher dimensions as well. The combinatorial complexity of an arrangement of n hyperplanes in $\mathbb{R}^d$ is $\Theta(n^d)$. Thus, these structures are only practical in spaces of relatively low dimension when n is not too large.

Incremental Construction: Arrangements are used for solving many problems in computational geometry. But in order to use an arrangement, we first must be able to construct it.[1] We will present a simple incremental algorithm, which builds an arrangement by adding lines one at a time. Unlike the other incremental algorithms we have seen so far, this one is *not randomized*. Its worst-case asymptotic running time, which is $O(n^2)$, holds irrespective of the insertion order. This is asymptotically optimal, since this is the size of the arrangement. The algorithm will also require $O(n^2)$ space, since this is the amount of storage needed to store the final result.

Let $L = \{\ell_1, \ldots, \ell_n\}$ denote the set of lines. We will add lines one by one and update the arrangement after each insertion. We will show that the ith line can be inserted in $O(i)$ time (irrespective of the insertion order). Summing over i, this yields a total running time proportional to $\sum_{i=1}^{n} i = O(n^2)$.

Suppose that the first $i-1$ lines have already been inserted. Consider the insertion of ℓ_i. We start by determining the leftmost (unbounded) face of the arrangement that contains this line. Observe that at $x = \infty$, the lines are sorted from top to bottom in increasing order of their slopes. In time $O(i)$ we can determine where the slope of ℓ_i falls relative to the slopes of the prior $i-1$ lines, and this determines the leftmost face of the arrangement that contains this

[1] This is not quite accurate. For some applications, it suffices to perform a plane-sweep of the arrangement. If we think of each line as an infinitely long line segment, the line segment intersection algorithm that was presented in class leads to an $O(n^2 \log n)$ time and $O(n)$ space solution. There exists a special version of plane sweep for planar line arrangements, called *topological plane sweep*, which runs in $O(n^2)$ time and $O(n)$ space. In spite of its fancy name, topological plane sweep is quite easy to implement.

line. (In fact, we could do this in $O(\log i)$ time by storing the slopes in an ordered dictionary, but this would not improve our overall running time. By our assumption that no two lines are parallel, there are no duplicate slopes.)

The newly inserted line cuts through a sequence of $i-1$ edges and i faces of the existing arrangement. In order to process the insertion, we need to determine which edges are cut by ℓ_i, and then we split each such edge and update the DCEL for the arrangement accordingly.

In order to determine which edges are cut by ℓ_i, we "walk" this line through the current arrangement, from one face to the next. Whenever we enter a face, we need to determine through which edge ℓ_i exits this face. We answer the question by a very simple strategy. We walk along the edges of the face, say in a counterclockwise direction until we find the exit edge, that is, the other edge that ℓ_i intersects. We then jump to the face on the other side of this edge and continue the trace with the neighboring face. This is illustrated in Fig. 2(a). The DCEL data structure supports such local traversals in time linear in the number of edges traversed. (You might wonder why we don't generalize the trapezoidal map algorithm. We could build a trapezoidal map of the arrangement and walk the new segment through a sequence of trapezoids. It turns out that this would be just as efficient.)

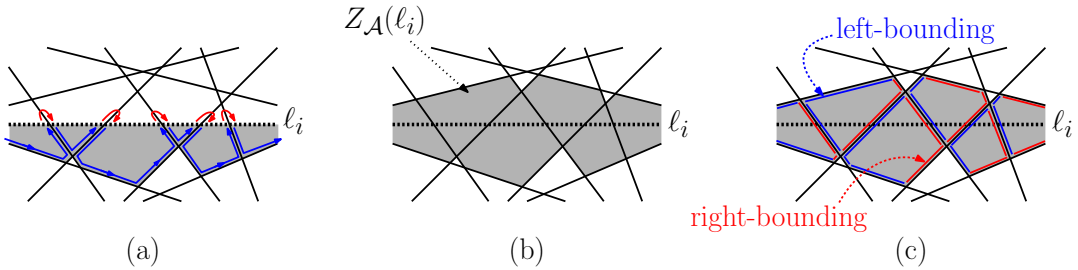

Fig. 2: Adding the line ℓ_i to the arrangement; (a) traversing the arrangement and (b) the zone of a line ℓ_i. (Note that only a portion of the zone is shown in the figure.)

Clearly, the time that it takes to perform the insertion is proportional to the total number of edges that have been traversed in this tracing process. A naive argument says that we encounter $i-1$ lines, and hence pass through i faces (assuming general position). Since each face is bounded by at most i lines, each facial traversal will take $O(i)$ time, and this gives a total $O(i^2)$, which is much higher than the $O(i)$ time that we promised earlier. Why is this wrong? It is based on bound of the total complexity of the faces traversed. To improve this, we need to delve more deeply into a concept of a *zone* of an arrangement.

Zone Theorem: The most important combinatorial property of arrangements (which is critical to their efficient construction) is a rather surprising result called the *zone theorem*. Given an arrangement $\mathcal{A}$ of a set L of n lines, and given a line ℓ that is not in L, the *zone* of ℓ in $\mathcal{A}(L)$, denoted $Z_{\mathcal{A}}(\ell)$, is the set of faces of the arrangement that are intersected by ℓ (shaded in Fig. 2(b)). For the purposes of the above construction, we are only interested in the edges of the zone that lie below ℓ_i, but if we bound the total complexity of the zone, then this will be an upper bound on the number of edges traversed in the above algorithm. The combinatorial complexity of a zone (as argued above) is at most $O(n^2)$. The Zone theorem states that the complexity is actually much smaller, only $O(n)$.

Theorem: (Zone Theorem) Given an arrangement $\mathcal{A}(L)$ of n lines in the plane, and given any line ℓ in the plane, the total number of edges in all the cells of the zone $Z_{\mathcal{A}}(\ell)$ is at most $6n$.

As with many combinatorial proofs, the key is to organize matter so that the counting can be done in an easy way. This is not trivial. We cannot count cell-by-cell, since some cells have high complexity and some low. We also cannot count line-by-line, because some lines contribute many edges to the zone and others just a few. The key in the proof is finding a (clever!) way to add up the edges so that each line appears to induce only a constant number of edges into the zone. (Note that our text counts zone edges a bit differently.)

Proof: The proof is based on a simple inductive argument. For the sake of illustration, let us rotate the plane so that ℓ is horizontal. By general position, we may assume that none of the lines of L are parallel to ℓ. We split the edges of the zone into two groups, those that bound some face from the left side and those that bound some face from the right side. An edge of a face is said to be *left bounding* if the face lies in the right halfplane of the line defining this edge, and a face is *right bounding* if the face lies in the left halfplane of the line defining this edge (see Fig. 2(c)). We will show that there are at most $3n$ left-bounding edges in the zone (highlighted in Fig. 3(a)), and by applying a symmetrical argument to the right-bounding edges, we have a a total of $6n$ edges.

The proof is by induction on n. For the basis case, when $n = 1$, then there is exactly one left-bounding edge in ℓ's zone, and $1 \leq 3 = 3n$. For the induction step, let us assume the induction hypothesis is true for any set of $n-1$ lines, and we will show that it holds for an arrangement of n lines. Consider the rightmost line of the arrangement to intersect ℓ. Call this ℓ_1 (see Fig. 2(c)). Prior to its existence, the induction hypothesis implies that there are at most $3(n-1)$ left-bounding edges in the zone of the remaining $n-1$ lines.

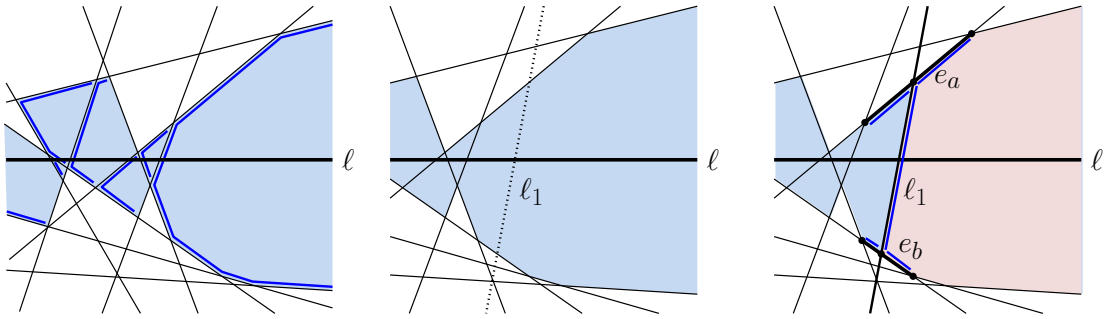

Fig. 3: Proof of the Zone Theorem.

Now, let us add ℓ_1 and see how many more left-bounding edges are generated. Because ℓ_1 is leftmost, it intersects the rightmost face of the zone. Observe that all of the edges of this face are left-bounding edges. By convexity, ℓ_1 intersects the boundary of this face in two edges, denoted e_a and e_b, where e_a is above ℓ, and e_b is below. Its insertion creates a new left-bounding edge running along ℓ_1 between e_a and e_b, and it splits each

of the edges e_a and e_b into two new left-bounding edges. Thus, there is a net increase by three edges, for a total of $3(n-1)+3 = 3n$ edges.

We assert that ℓ_1 cannot contribute any other left-bounding edges to the zone. This is because the lines containing e_a and e_b block any possibility of this. Therefore, there are at most $3n$ left bounding edges, as desired.