CMSC 754: Lecture 8
Halfplane Intersection and Point-Line Duality
Tuesday, Feb 28, 2012

Reading: Chapter 4 in the 4M's, with some elements from Sections 8.2 and 11.4.

Halfplane Intersection: Today we begin studying another very fundamental topic in geometric computing, and along the way we will show a rather surprising connection between this topic and the topic of convex hulls. Any line in the plane splits the plane into two regions, one lying on either side of the line. Each such region is called a *halfplane*. (In d-dimensional space the corresponding notion is a *halfspace*, which consists of the space lying to one side of a $(d-1)$-dimensional hyperplane.) We say that a halfplane is either *closed* or *open* depending on whether or not it contains the line itself. For this lecture we will be dealing entirely with closed halfplanes.

How do we represent lines and halfplanes? For our purposes (since, by general position, we may assume we are dealing only with nonvertical lines), it will suffice to represent lines in the plane using the following equation:

$$y = ax - b,$$

where a denotes the slope and b denotes the negation of the y-intercept. (We will see later why it is convenient to negate the intercept value.) Note that this is not fully general, since it cannot handle vertical lines (which have infinite slope). Each nonvertical line defines two closed *halfplanes*, consisting of the points on or below the line and the points on or above the line:

lower (closed) halfplane: $y \leq ax - b$ upper (closed) halfplane: $y \geq ax - b$.

Halfplane intersection problem: The *halfplane intersection problem* is, given a set of n closed halfplanes, $H = \{h_1, h_2, \ldots, h_n\}$ compute their intersection. A halfplane (closed or open) is a convex set, and hence the intersection of any number of halfplanes is also a convex set. (Fig. 1 illustrates the intersection of a collection of upper halfspaces.) Unlike the convex hull problem, the intersection of n halfplanes may generally be empty or even unbounded. A natural output representation might be to list the lines bounding the intersection in counterclockwise order.

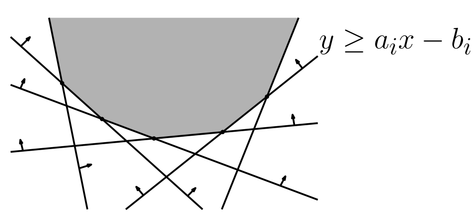

Figure 1: Halfplane intersection.

How many sides can bound the intersection of n halfplanes in the worst case? Observe that by convexity, each of the halfplanes can appear only once as a side, and hence the maximum number of sides is n. How fast can we compute the intersection of halfplanes? As with the convex hull problem, it can be shown, through a suitable reduction from sorting, that the problem has a lower bound of $\Omega(n \log n)$.

Who cares about this problem? Halfplane intersection and halfspace intersection in higher dimensions are also used in generating convex shape approximations. For example, in robotics and computer graphics, rather than computing collisions with a complex shape, it is easier to first check for collisions with a enclosing convex approximation to the shape. Also, many optimization problems can be expressed as minimization problems over a convex domain, and these domains are represented by the intersection of halfspaces.

Solving the halfspace intersection problem in higher dimensions is quite a bit more challenging than in the plane. In general, the worst-case total combinatorial complexity the intersection of n halfspaces in $\mathbb{R}^d$ can be as high as $\Theta(n^{\lfloor d/2 \rfloor})$. For example, the boundary of the intersection of halfspaces in dimension d is a $(d-1)$-dimensional cell complex, and would require an appropriate data structure for storing such objects.

We will discuss two algorithms for the halfplane intersection problem. The first is given in the text, and involves an interesting combination of two techniques we have discussed for geometric problems, geometric divide-and-conquer and plane sweep. For the other, we will consider somewhat simpler problem of computing something called the *lower envelope* of a set of lines, and show that it is closely related to the convex hull problem.

Divide-and-Conquer Algorithm: We begin by sketching a divide-and-conquer algorithm for computing the intersection of halfplanes. The basic approach is very simple:

(1) If $n = 1$, then just return this halfplane as the answer.

(2) Split the n halfplanes of H into subsets H_1 and H_2 of sizes $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$, respectively.

(3) Compute the intersection of H_1 and H_2, each by calling this procedure recursively. Let K_1 and K_2 be the results.

(4) Intersect the convex polygons K_1 and K_2 (which might be unbounded) into a single convex polygon K, and return K.

The running time of the resulting algorithm is most easily described using a *recurrence*, that is, a recursively defined equation. If we ignore constant factors, and assume for simplicity that n is a power of 2, then the running time can be described as:

$$T(n) \;=\; = \begin{cases} 1 & \text{if } n = 1, \\ 2T(n/2) + M(n) & \text{if } n > 1, \end{cases}$$

where $M(n)$ is the time required to merge the two results, that is, to compute the intersection of two convex polygons whose total complexity is n. We will show below that $M(n) = O(n)$, and so it follows by standard results in recurrences that the overall running time $T(n)$ is $O(n \log n)$. (See any standard algorithms textbook.)

Intersecting Two Convex Polygons: The only nontrivial part of the process is implementing an algorithm that intersects two convex polygons, K_1 and K_2, into a single convex polygon. Note that these are somewhat special convex polygons because they may be empty or unbounded.

We know that it is possible to compute the intersection of line segments in $O((n + I) \log n)$ time, where I is the number of intersecting pairs. Two convex polygons cannot intersect in more than $I = O(n)$ pairs. (As an exercise, try to prove this.) This would given $O(n \log n)$ algorithm for computing the intersection. This is too slow, however, and would result in an overall time of $O(n \log^2 n)$ for $T(n)$.

There are two common approaches for intersecting convex polygons. Both essentially involve merging the two boundaries. One works by a plane-sweep approach. The other involves a simultaneous counterclockwise sweep around the two boundaries. The latter algorithm is described in O'Rourke's book. We'll discuss the plane-sweep algorithm.

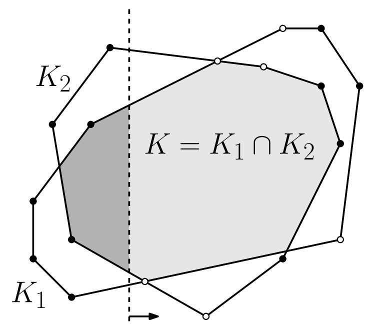

Figure 2: Intersecting two convex polygons by plane sweep.

We perform a left-to-right plane sweep to compute the intersection (see Fig. 2). We begin by breaking the boundaries of the convex polygons into their upper and lower chains. (This can be done in $O(n)$ time.) By convexity, the sweep line intersects the boundary of each convex polygon K_i in at most two points, and hence, there are at most four points in the sweep line status at any time. Thus, we do not need a ordered dictionary for storing the sweep line status—a simple 4-element list suffices. Also, our event queue need only be of constant size. At any point there are at most 8 possible candidates for the next event, namely, the right endpoints of the four edges stabbed by the sweep line and the (up to four) intersection points of these upper and lower edges of K_1 with the upper and lower edges of K_2. Since there are only a constant number of possible events, and each can be handled in $O(1)$ time, the total running time is $O(n)$.

Lower Envelopes and Duality: Next we consider a slight variant of this problem, to demonstrate some connections with convex hulls. These connections are very important to an understanding of computational geometry, and we see more about them in the future. These connections have to do with a concept called *point-line duality*. In a nutshell there is a remarkable similarity between how points interact with each other an how lines interact with each other. Sometimes it is possible to take a problem involving points and map it to an equivalent problem involving lines, and vice versa. In the process, new insights to the problem may become apparent.

The problem to consider is called the *lower envelope* problem, and it is a special case of the halfplane intersection problem. We are given a set of n lines $L = \{\ell_1, \ell_2, \ldots, \ell_n\}$ where ℓ_i is of the form $y = a_i x - b_i$. Think of these lines as defining n halfplanes, $y \leq a_i x - b_i$, each lying *below* one of the lines. The *lower envelope* of L is the boundary of the intersection of these half planes (see Fig. 3). The *upper envelope* is defined symmetrically.

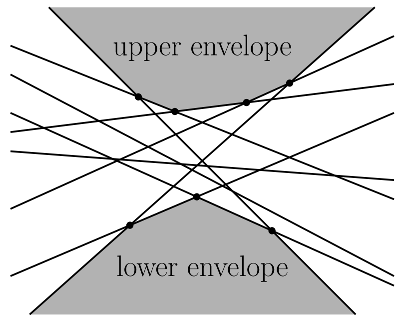

Figure 3: Lower and upper envelopes.

The lower envelope problem is a restriction of the halfplane intersection problem, but it an interesting restriction. Notice that any halfplane intersection problem that does not involve any vertical lines can be rephrased as the intersection of two envelopes, a lower envelope defined by the lower halfplanes and an upper envelope defined by the upward halfplanes.

We will see that solving the lower envelope problem is very similar to solving the upper convex hull problem. In fact, they are so similar that exactly the same algorithm will solve both problems, without changing even a single character of code! All that changes is the way in which you interpret the inputs and the outputs.

Lines, Points, and Incidences: In order to motivate duality, let us discuss the representation of lines in the plane. Each line can be represented in a number of ways, but for now, let us assume the representation $y = ax - b$, for some scalar values a and b. (Why $-b$ rather than $+b$? The distinction is unimportant, but it will simplify some of the notation defined below.) We cannot represent vertical lines in this way, and for now we will just ignore them.

Therefore, in order to describe a line in the plane, you need only give its two coefficients (a, b). Thus, lines in the plane can be thought of as points in a new 2-dimensional space, in which the coordinate axes are labeled (a, b), rather than (x, y). For example, the line $\ell : y = 2x + 1$ corresponds to the point $(2, -1)$ in this space, which we denote by ℓ^*. Conversely, each point $p = (a, b)$ in this space of "lines" corresponds to a nonvertical line, $y = ax - b$ in the original plane, which we denote by p^*. We will call the original (x, y)-plane the *primal plane*, and the new (a, b)-plane the *dual plane*.

This insight would not be of much use unless we could say something about how geometric relationships in one space relate to the other. The connection between the two involves incidences between points and line. Two lines determine a point through intersection. Two points determine a line, by taking their affine combination. Later, we'll show that these relationships are preserved by duality. For example, consider the two lines $\ell_1 : y = 2x + 1$ and the line $\ell_2 : y = -\frac{x}{2} + 6$ (see Fig. 4(a)). These two lines intersect at the point $p = (2, 5)$. The duals of these two lines are $\ell_1^* = (2, -1)$ and $\ell_2^* = \left(-\frac{1}{2}, -6\right)$. The line in the (a, b) dual plane passing through these two points is easily verified to be $b = 2a - 5$. Observe that this is exactly the dual of the point p (see Fig. 4(b)). (As an exercise, prove this for two general lines.)

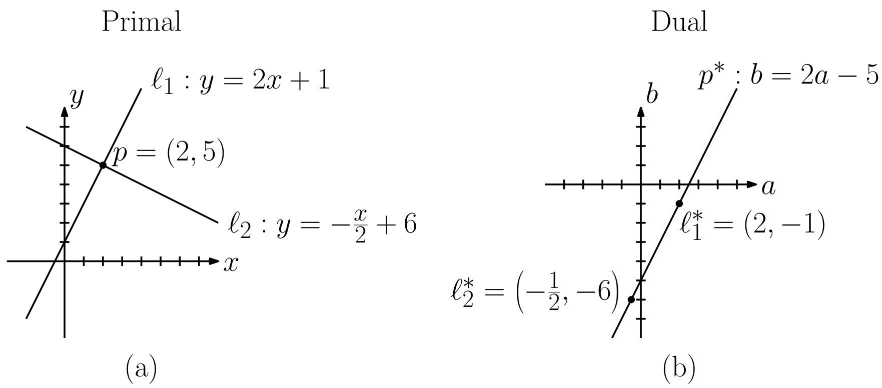

Figure 4: The primal and dual planes.

Point-Line Duality: Let us explore this dual transform more formally. Duality (or more specifically *point-line duality*) is a transformation that maps points in the plane to lines and lines to point. (More generally, it maps points in d-space to hyperplanes dimension d.) We denote this transformation using a asterisk ($*$) as a superscript. Thus, given point p and line ℓ in the primal plane we define ℓ^* and p^* to be a point and line, respectively, in the dual plane defined as follows.[1]

$$\begin{array}{lll} \ell : y = \ell_a x - \ell_b & \Rightarrow & \ell^* = (\ell_a, \ell_b) \\ p = (p_x, p_y) & \Rightarrow & p^* : b = p_x a - p_y. \end{array}$$

It is convenient to define the dual transformation so that it is its own inverse (that is, it is an involution). In particular, it maps points in the dual plane to lines in the primal, and vice versa. For example, given a point $p = (p_a, p_b)$ in the dual plane, its dual is the line $y = p_a x - p_b$ in the primal plane, and is denoted by p^*. It follows that $p^{**} = p$ and $\ell^{**} = \ell$.

Properties of Point-Line Duality: Duality has a number of interesting properties, each of which is easy to verify by substituting the definition and a little algebra.

Self Inverse: $p^{**} = p$.

Order reversing: Point p is above/on/below line ℓ in the primal plane if and only if line p^* is below/on/above point ℓ^* in the dual plane, respectively (see Fig. 5).

Intersection preserving: Lines ℓ_1 and ℓ_2 intersect at point p if and only if the dual line p^* passes through points ℓ_1^* and ℓ_2^*.

Collinearity/Coincidence: Three points are collinear in the primal plane if and only if their dual lines intersect in a common point.

[1]Duality can be generalized to higher dimensions as well. In $\mathbb{R}^d$, let us identify the y axis with the d-th coordinate vector, so that an arbitrary point can be written as $p = (x_1, \ldots, x_{d-1}, y)$ and a $(d-1)$-dimensional hyperplane can be written as $h : y = \sum_{i=1}^{d-1} a_i x_i - b$. The dual of this hyperplane is $h^* = (a_1, \ldots, a_{d-1}, -b)$ and the dual of the point p is $p^* : b = \sum_{i=1}^{d-1} x_i a_i - y$. All the properties defined for point-line relationships generalize naturally to point-hyperplane relationships.

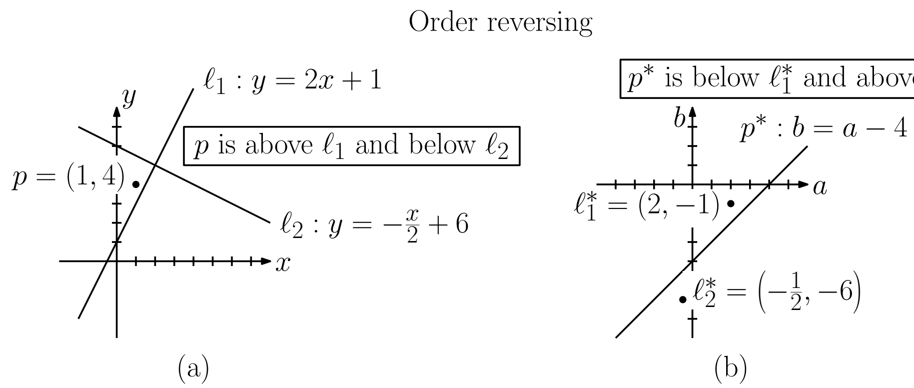

Figure 5: The order-reversing property.

The self inverse property was already established (essentially by definition). To verify the order reversing property, consider any point p and any line ℓ.

$$p \text{ is on or above } \ell \iff p_y \geq \ell_a p_x - \ell_b \iff \ell_b \geq p_x \ell_a - p_y \iff p^* \text{ is on or below } \ell^*$$

(From this is should be apparent why we chose to negate the y-intercept when dualizing points to lines.) The other two properties (intersection preservation and collinearity/coincidence are direct consequences of the order reversing property.)

Convex Hulls and Envelopes: Let us return now to the question of the relationship between convex hulls and the lower/upper envelopes of a collection of lines in the plane. The following lemma demonstrates the, under the duality transformation, the convex hull problem is dually equivalent to the problem of computing lower and upper envelopes.

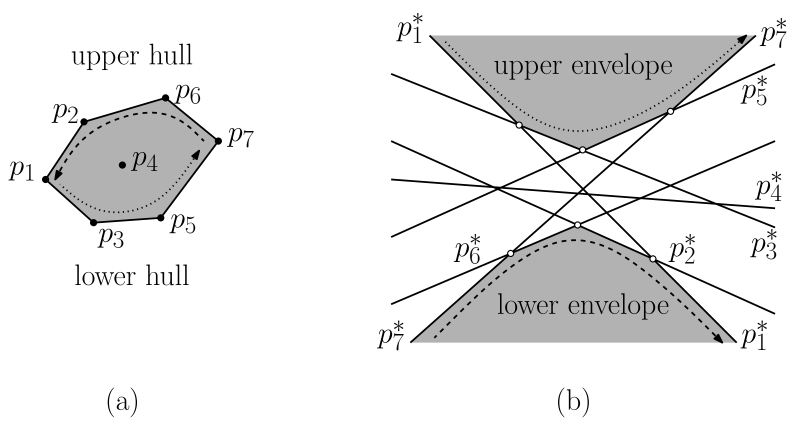

Figure 6: Equivalence of hulls and envelopes.

Lemma: Let P be a set of points in the plane. The counterclockwise order of the points along the upper (lower) convex hull of P (see Fig. 6(a)), is equal to the left-to-right order of the sequence of lines on the lower (upper) envelope of the dual P^* (see Fig. 6(b)).

Proof: We will prove the result just for the upper hull and lower envelope, since the other case is symmetrical. For simplicity, let us assume that no three points are collinear.

Consider a pair of points p_i and p_j that are consecutive vertices on the upper convex hull. This is equivalent to saying that all the other points of P lie beneath the line ℓ_{ij} that passes through both of these points.

Consider the dual lines p_i^* and p_j^*. By the incidence preserving property, the dual point ℓ_{ij}^* is the intersection point of these two lines. (By general position, we may assume that the two points have different x-coordinates, and hence the lines have different slopes. Therefore, they are not parallel, and the intersection point exists.)

By the order reversing property, all the dual lines of P^* pass above point ℓ_{ij}^*. This is equivalent to saying the ℓ_{ij}^* lies on the lower envelope of P^*.

To see how the order of points along the hulls are represented along the lower envelope, observe that as we move counterclockwise along the upper hull (from right to left), the slopes of the edges increase monotonically. Since the slope of a line in the primal plane is the a-coordinate of the dual point, it follows that as we move counterclockwise along the upper hull, we visit the lower envelope from left to right.

One rather cryptic feature of this proof is that, although the upper and lower hulls appear to be connected, the upper and lower envelopes of a set of lines appears to consist of two disconnected sets. To make sense of this, we should interpret the primal and dual planes from the perspective of projective geometry, and think of the rightmost line of the lower envelope as "wrapping around" to the leftmost line of the upper envelope, and vice versa. The places where the two envelopes wraps around correspond to the vertical lines (having infinite slope) passing through the left and right endpoints of the hull. (As an exercise, can you see which is which?)