Stabbing and ray shooting in 3 dimensional space[1]

Marco Pellegrini

Department of Computer Science
Courant Institute of Mathematical Sciences
New York University

May 25, 1994

---

[1] Research partially supported by Eni and Enidata within the AXL project, and by NSF grant CCR-8901484.

Abstract

In this paper we consider the following problems: given a set T of triangles in 3-space, with $|T| = n$,

a) answer the query " given a line l, does l stab the set of triangles?" (*query problem*).

b) find whether a stabbing line exists for the set of triangles (*existence problem*).

c) Given a ray ρ, which is the first triangle in T hit by ρ?

The following results are shown.

1. There is an $\Omega(n^3)$ lower bound on the descriptive complexity of the set of all stabbers for a set of triangles.
2. The existence problem for triangles on a set of planes with g different plane inclinations can be solved in $O(g^2 n^2 \log n)$ time (Theorem 2).
3. The query problem is solvable in quasi-quadratic $O(n^{2+\epsilon})$ preprocessing and storage and logarithmic $O(\log n)$ query time (Theorem 4).
4. All stabbing results for triangles extend, with the same asymptotic bounds, to sets of convex polyhedra with total complexity n.
5. Using $O(n^{3+\epsilon})$ preprocessing time and storage we can determine the first triangle hit in a set of disjoint triangles by any given ray ρ in $O(\sqrt{n} \log n)$ worst case query time (Theorem 8).
6. If we are given m rays we can answer ray shooting queries in

$$O(m^{5/6-\delta} n^{5/6+5\delta} \log^2 n + m \log^2 n + n \log n \log m)$$

randomized expected time and $O(m + n)$ space (Theorem 9).
7. In time $O((n + m)^{5/3+4\delta})$ it is possible to decide whether two non convex polyhedra of complexity m and n intersect (Corollary 1).
8. Given m rays and n axis-oriented boxes we can answer ray shooting queries in randomized expected time $O(m^{3/4-\delta} n^{3/4+3\delta} \log^4 n + m \log^4 n + n \log n \log m)$ and $O(m + n)$ space (Theorem 10).

1 Introduction

Avis and Wenger [AW87] first presented algorithms for line stabbing of several kinds of objects in 3 dimensional space. When the stabbed objects are lines, $O(n)$ time suffices to find a stabbing line, for line segments the time is $O(n \log n)$, while for a set of polyhedra they gave an $O(n^5)$ time algorithm which was later improved to $O(n^4 \log n)$.

New results on the problem of stabbing objects in space appeared recently; in particular, McKenna and O'Rourke [MO88] give an $O(n^4 \alpha(n))$ time, $O(n^2)$ space algorithm for finding a maximal stabbing line for a set of polyhedra; using $O(n^4 \alpha(n))$ time they can construct a more general data structure to answer several problems, like enumerating pairs of visible faces, and determining if 2 lines are in the same class (i.e. they can be deformed continuously one into the other without crossing the edges of the polyhedra).

Jaromczyk and Kowaluk [JK88] claim an $O(n^3 2^{\alpha(n)} \log n)$ algorithm for detecting a stabbing line for a family of convex polyhedra with total combinatorial complexity n; a characterization of a subclass of all stabbing lines (those passing through at least two edges of the polyhedra) requires $O(n^3 2^{\alpha(n)} \log n)$ time and $O(n^3 2^{\alpha(n)})$ space. Unfortunately there are cases in which the approach in [JK88] does not attain the desired complexity (Section 2 provides more details on this issue).

The bounds in [MO88] on the stabbing and visibility problems in 3 dimensions are clearly not satisfactory for most practical applications and in this paper we try to get better bounds on a simplified version of the problem. We consider plane polygons in E^3, and in particular triangles, as a simple case. Then the results for triangles are extended to other plane convex polygons and to convex polyhedra (see Section 5).

In this paper we consider the special case of polyhedra with facets on a constant number of plane orientations (e.g., isothetic boxes), obtaining a bound of $O(n^2)$ on the complexity of the set of stabbers. We address also the query problem; using $O(n^{2+\epsilon})$ preprocessing and storage we determine whether a line is a stabber in $O(\log n)$ time. The techniques used in [JK88] and [MO88] do not provide any simple way to deal with query problems and no previous method to solve query stabbing problems in 3 dimension is known.

The stabbing problem suggests that we look at more general problems concerning lines in 3-space. Lines in 3-space lack many properties which are useful for deriving algorithms. First of all, a line does not have any separation property with respect to the space; second, duality cannot be applied directly since the dual of a line is still a line; third, a line is characterized by 4 parameters, therefore we are led to work in higher dimensional parametric spaces.

In the context of line problems we deal also with visibility problems. Using the technique of Plücker coordinates we can solve ray shooting query problems (Section 6).

The ray shooting problem is widely studied in computational geometry; usually the problem is stated with some restrictions on the rays or on the polygonal scene. Also we distinguish the on-line case from the batched case when all rays are known in advance. When the triangles form a terrain, on-line ray queries can be answered in $O(\log^2 n)$ time with $O(n^{2+\epsilon})$ preprocessing time and storage [CEGS89]. If we force the source of the ray to lay on a vertical line the method of Cole and Sharir [CS86] allows us to answer on-line queries in $O(\log^2 n)$ time after $O(n\lambda_4(n) \log n)$ preprocessing, using $O(n\lambda_4(n))$ storage.

When the source of the rays is on a (non-vertical) line (also called *flightpath*) and for any polygonal scene, Bern, Dobkin Eppstein and Grossman [BDEG90] build in $O((n^2 + k)\log^2 n + p \log n)$ time a data structure of size $O(n^2 + k)$ that answers ray-queries in $O(\log^2 n)$; where k is the number of opaque topological changes of the scene along the flightpath, p is the number of transparent topological changes, and, in general, $0 \le k \le p \le n^3/3$. In the case of vertical flightpath and polygonal terrain Bern et al. [BDEG90] improve the query time of the Cole and Sharir method to $O(\log n)$.

In the batched case, when the rays have a unique source, Overmars and Sharir

[OS89] solve the problem in time $O(n^{2/3}m^{2/3}\log^{\gamma} n)$ for n triangles, m rays (m and n sufficiently large) and a small constant γ. On-line single source ray shooting queries can be solved in time $O(\sqrt{n}\log n)$ using $O(n\log n)$ space and $O(n^{3/2}\log^{\gamma} n)$ preprocessing by modifying the implicit point location method of Agarwal [Aga89a] [Aga89b].

Schmitt, Müller and Leister [SML88] consider the case when the objects are isoriented rectangles, in the batched case they use $O(m^{0.82}n^{0.59}\log^{c} n)$ time and $O(m\log^{c} m + n)$ storage.

For the problem involving any polygonal scene and any set of rays only the trivial $O(mn)$ bound is known. Ray shooting (also called ray tracing) is widely used in graphics to simulate the effect of a light source on surfaces, and to simulate reflection and refraction of light. Ray tracing is an expensive computation and most of the method used in graphics to speed up the computation [Gla89] [SML88] [AK87] do not perform well in the worst case. The result presented in this paper (Sections 7 and 8, Theorem 9), although not yet practical because of the huge constants involved, is the first examples of ray shooting algorithm with a sublinear cost per ray, for any set of triangles and rays.

Moreover, the general ray shooting algorithm is modified to solve the problem of detecting whether two non-convex polyhedra intersect, in subquadratic time.

In Section 3, we present the basic theory of Plücker coordinates which we will exploit to characterize sets of stabbing lines, and obtain results in some restricted cases.

Section 4 deals with the query stabbing problem (i.e., preprocess the set of objects in order to answer quickly the question whether a query line is a stabber); this turns out to be easier than the more classical problem that asks whether there exists a stabber.

Sections 6 deals with on-line ray-shooting. Section 7 deals with batched ray-shooting. In Section 8 we improve slightly the results of Sections 6 and 7.

2 The skew projection technique

Jaromczyk and Kowaluk [JK88] describe a technique called *skew projection* to solve the existence problem. In this section we describe this method and we show how the time bound analysis presented in [JK88] is correct only for a set of triangles (or polyhedra) satisfying additional and rather unnatural conditions.

Given 2 skew lines l_1, l_2 in R^3, there exists a plane h_{12} parallel to both, unique up to a translation, which is called *screen*, if it does not contain l_1 or l_2. Let h_1 (resp. h_2) be the plane parallel to the screen h_{12} and containing l_1 (resp. l_2).

We define a map *sp* as follows $sp : R^3 - (h_1 \cup h_2) \rightarrow h_{12}$. Given a point $p \in R^3 - (h_1 \cup h_2)$ and 2 skewed lines l_1 and l_2 there is a unique line l' passing through p, l_1 and l_2; we take as $sp(p)$ the intersection of l' with the screen h_{12}: the mapping *sp* is the *skew projection.*

The skew projection has the following properties:

1. It is continuous.
2. $\partial sp(P) \subset sp(\partial P)$, for any convex set P, where ∂ is the relative boundary operator.
3. The projection of a line is a hyperbola, a parabola or 2 lines.

The skew projection is applied to the triangle stabbing problem in the following way: assume T to be a family of triangles, we choose 2 edges in T and take the 2 lines supporting the edges as the lines l_1, l_2 of the skew projection (repeating the projection for all $O(n^2)$ possible choices of 2 edges). For any $t \in T$, $sp(t)$ is a region bounded by arcs of hyperbola or parabola or line segments; define

$$sp_T = \bigcap_{t \in T} sp(t)$$

If sp_T is not empty for some choice of the base lines used in the projection (note that there are $O(n^2)$ possible pairs of base lines) then there is a stabber; conversely if there is a stabber, some sp_T among the $O(n^2)$ possible projections is not empty.

The core of the method lies in the way sp_T is computed. Projecting the edges in T, an arrangement of $O(n)$ hyperbolic arcs is obtained; by computing this arrangement in time roughly $O(n^2)$, sp_T is also determined[2]. The region sp_T can be computed more efficiently as a region between a lower envelope and an upper envelope of the curves corresponding to single edges of T *only* when each triangle in T maps onto the screen h_{12} in a *monotone*[3] region. The skew projection cannot guarantee this property in general.

When a triangle t intersects h_1 (and it does not lie on a plane parallel to l_1 or l_2), the region $sp(t)$ is monotone in one direction on the screen (see Figure 1). When the triangle t intersects h_2, the region $sp(t)$ is monotone in a *different* direction on the screen.

In the argument of [JK88], therefore, it is not correct to use the results from Hart and Sharir [HS86] on lower envelopes of curves to derive an almost linear $O(n2^{\alpha(n)})$ complexity bound for sp_T. Moreover it seems hard to find a trivial way to improve the proof in [JK88].

3 Plücker Coordinates of lines

Triangles in 3 dimensional space are plane figures bounded by edges, therefore we need to be able to efficiently represent and manipulate segments and lines in E^3.

We use a representation for lines called *Plücker coordinates of the line* [Som51]. Two points $x = (x_0, x_1, x_2, x_3)$ $y = (y_0, y_1, y_2, y_3)$ in 3-dimensional homogeneous coordinates define a line l in 3-space. The six quantities[4]:

$$\xi_{ij} = x_i y_j - x_j y_i \tag{1}$$

$$ij = 01, 02, 03, 12, 23, 31 \tag{2}$$

are called *Plücker coordinates* of the line l (oriented from x to y). They correspond to the 2 by 2 minors of the 2 by 4 matrix formed by the coordinates of the point x (on the first row) and y (on the second row).

---

[2] The total cost would be roughly $O(n^4)$ at this point of the analysis

[3] A region on the plane is monotone if its intersection with any vertical line is connected.

[4] The indices of the six quantities are chosen so as to obtain only positive signs in the following formulas.

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \end{pmatrix} \tag{3}$$

The six parameters are not independent; they must satisfy the following equation (whose solution constitute the Plücker hypersuperface or Klein quadric or Grassman manifold $\mathcal{F}_4^2$ [Sto89] [HP52]):

$$\Pi \; : \; \xi_{01}\xi_{23} + \xi_{02}\xi_{31} + \xi_{03}\xi_{12} = 0 \tag{4}$$

The incidence relation between two lines l and l' can be expressed using the Plücker coordinates of l and l'. Let a_1, b_1 (resp. a_2, b_2) be two points on l (resp. l') oriented as l (resp. l'). The incidence between l and l' is expressed as the vanishing of the determinant of a 4 by 4 matrix whose rows are the coordinates of a_1, b_1, a_2, b_2 in this order from top to bottom.[5]

$$\begin{vmatrix} a_{10} & a_{11} & a_{12} & a_{13} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ b_{20} & b_{21} & b_{22} & b_{23} \end{vmatrix} = 0 \tag{5}$$

If we expand the determinant according to the 2 by 2 minors of the two submatrices formed by the first two rows and from the last two rows we obtain the following equation in which only Plücker coordinates are involved:

$$\xi_{01}\xi'_{23} + \xi_{02}\xi'_{31} + \xi_{03}\xi'_{12} + \xi'_{01}\xi_{23} + \xi'_{02}\xi_{31} + \xi'_{03}\xi_{12} = 0 \tag{6}$$

In this paper we use the notation $D(a, b, c, d)$ for the determinant formed by the coordinates of the points a, b, c, d placed on the rows in this order from top to bottom.

Let us introduce 2 mappings: $\pi : l \rightarrow \pi_l$ maps a line in R^3 to an hyperplane in R^6 whose plane coordinates are the Plücker coordinates of l appropriately reordered. $p : l \rightarrow p_l$ maps a line in R^3 to a point in R^6 whose coordinates are the Plücker coordinates of the line.

The incidence relation between the two lines l, l' (expressed by equation 6) can be reformulated as an incidence relation between points and hyperplanes in R^6. Equation (6) can be rewritten in the form $\pi_l(p_{l'}) = 0$, which is equivalent to to requiring point $p_{l'}$ to belong to hyperplane π_l. Equation (6) is homogeneous; therefore we reduce the number of coordinates to 5, modifying accordingly the mappings π and p. This reduction of space dimension is important for the complexity analysis of the algorithms presented in this paper.

3.1 Characterization of stabbing lines using Plücker coordinates

---

[5] The absolute value of this determinant represents the volume of the tetrahedron whose vertices are a_1, b_1, a_2 and b_2. The sign of the determinant is positive if the quadruple of points has the same orientation as the reference frame chosen in R^3. When the volume vanishes the 4 points are coplanar, therefore the 2 lines intersect (or are parallel).

Definition 1 *Given the point* a *and the triangle* t *in 3-space the cone* $\mathcal{C}_{a,t}$ *is the set of rays from* a *intersecting* t. *(see Figure 3)*

A family of triangles T and a point a define a family of cones $\mathcal{C}_{a,T} = \{\mathcal{C}_{a,t} \mid t \in T\}$. We say that that the family of cones $\mathcal{C}_{a,T}$ is *based* on T with *apex a*. In the following we restrict our attention to stabbing lines *intersecting a reference plane* $P : z = -1$ and we assume without loss of generality that the triangles in T are all above P. It is easy to prove the following lemma:

Lemma 1 *There exists a stabbing line for* T *if and only if there is a point* q *on* P *such that* $\bigcap \mathcal{C}_{q,T} \neq \emptyset$.

The next step is the characterization of the cones as the intersection of three halfspaces. Let $\mathcal{C}_{q,t}$ be the cone based on triangle $t = (p_1, p_2, p_3)$ and with apex $q \in P$. Consider the three planes spanned by two vertices of t and the apex q; the halfspace containing the third vertex of t is called *positive*. When q belongs to the plane spanned by t the cone degenerates in a 2-dimensional object, but for simplicity of exposition we ignore degenerate cases.

It is easy to check that the set of points in the rays belonging to the cone $\mathcal{C}_{q,t}$ is the intersection of the 3 positive halfspaces determined by q and t (see Figure (3)). [6]

If t is parallel to the reference plane P, then the variable point Q in the cone $\mathcal{C}_{q,t}$ satisfies the following system of linear inequalities:

$$\begin{aligned} D(q, p_1, p_2, Q) &\geq 0 \\ D(q, p_2, p_3, Q) &\geq 0 \\ D(q, p_3, p_1, Q) &\geq 0 \end{aligned} \tag{7}$$

If necessary relabel the vertices of t to ensure the inequalities all have the same sign.

When t is not parallel to P we must consider the position of apex q with respect to the line l_t, which is the intersection of P with the spanning plane of t (we will denote this plane with *aff*(t)). The cone is then defined by the two systems of linear inequalities (8) and (9), according to the position of q relative to l_t:

$$\begin{aligned} D(q, p_1, p_2, Q) &\geq 0 \\ D(q, p_2, p_3, Q) &\geq 0 \\ D(q, p_3, p_1, Q) &\geq 0 \end{aligned} \tag{8}$$

or

$$\begin{aligned} D(q, p_1, p_2, Q) &\leq 0 \\ D(q, p_2, p_3, Q) &\leq 0 \\ D(q, p_3, p_1, Q) &\leq 0 \end{aligned} \tag{9}$$

---

[6] Note that the intersection defines exactly the cone and its central symmetric image through the apex q. This fact does not impair the argument.

We need two systems of inequalities because when the apex q crosses l_t the three positive halfspaces switch with the non-positive ones. In Figure (4), system (8) is valid for apex q, system (9) for apex q'.

Using the *row exchange rule* of determinants and changing the sign of the inequalities accordingly we can put all the determinants in the systems (7), (8) and (9) in the form $D(p_1, p_2, q, Q)$, $D(p_2, p_3, q, Q)$, and $D(p_3, p_1, q, Q)$. Then we expand those determinants according to minors of the first 2 columns and of the last two, and obtain a linear expression in terms of Plücker coordinates. Note that the final systems involve the Plücker coordinates of lines supporting edges of triangles in T and the (variable) line passing through q and Q (see Figure (2)).

The above discussion shows how to characterize stabbing lines intersecting a reference plane P. Now, let us consider 3 mutually orthogonal planes P_1, P_2, P_3; every line in R^3 must intersect at least one of them. Therefore, characterizing the stabbing lines through P_1, P_2 and P_3 in turn, we are able to find any stabbing line. From the algorithmic point of view this implies that we have always to solve 3 instances of the same problem, each with a different reference plane, to obtain a correct answer. The asymptotic complexity of the whole algorithm is the same as for one of the three restricted cases; it is possible, however, to gain something in the last phase. Since a line parallel to P_1 and P_2 must be perpendicular to P_3 the problem reduces to a planar problem in which we project T orthogonally onto P_3 and ask whether there exists a point in the intersection of all the projected triangles.

3.2 Special forms of the existence problem

Each triangle in $t \in T$ lies on a spanning plane *aff*(t), which intersects the reference plane in a (proper or improper) line l_t. Let us call $\mathcal{M}$ the planar arrangement on P generated by the lines l_t for $t \in T$.

A special case in which it is possible to express the intersection of all the cones $\mathcal{C}_{q,T}$ using only one set of inequalities is when the family of planes $aff(T) = \{aff(t) \mid t \in T\}$ is a family of parallel planes. We choose the reference plane P parallel to the family of supporting planes. The map $\mathcal{M}$ has only one region (the whole plane) and there is only one possible system of inequalities generated by concatenating the equations of the single cones in one system.

Theorem 1 *The existence problem for a set of triangles lying on a family of parallel planes can be solved in $O(n^2 \log n)$ time.*

Proof. Using Seidel's convex hull algorithm [Sei86] we compute the *Plücker polytope* R corresponding to the solution of the unique system of linear inequalities in $O(n^2 \log n)$. The second step is to check that the Plücker hypersuperface Π intersects the polytope R, $R \cap \Pi \neq \emptyset$. The test proceeds as follows[7]:

1. Compute the sign of Π at any vertex of R. If there are two vertices with different signs, by continuity we conclude that there is a point where the curve Π enters the polytope R.

---

[7] An alternative is to divide R into simplices and check each one for intersection with Π. We can partition R in $O(n^2)$ simplices, and each test uses constant time.

2. Intersect Π with any line supporting a 1-face of R (the 1-face can lie on Π or) there are at most 2 intersection points; we can check if they lie on the edge by comparing these points to the endpoints of the 1-face.
3. Intersect Π with each 2-face f; the result is a second degree curve Π' that lies on the affine spanning space of f, which does not cross the 1-faces bounding f. Here we need checking one point in each component of Π' against the 1-faces of f and determine if one of the components is in the interior of f.
4. Intersect Π with any 3-face f; we obtain a second degree surface Π'' that does not intersect any of the 2-faces of f; by checking one point in each component of Π'' against the planes supporting the 2-faces of f we can determine if one component of Π'' is in the interior of f.

The total complexity of the test is linear in the size of the adjacency graph for the polytope R, therefore it is $O(n^2)$ (see [Ede87] chapter 6, Theorem 6.12). ∎

The second special case we consider is when the spanning planes for T belong to g families of parallel planes. We choose our reference plane P at infinity (i.e., we consider only the direction of the lines). The arrangement $\mathcal{M}$ has $O(g^2)$ faces and for each face $\sigma \in \mathcal{M}$ the cones with apex in σ are defined by one system. We can solve $O(g^2)$ subproblems in essentially the same way as in the proof of Theorem (1), and derive:

Theorem 2 *The existence problem for a family of n triangles lying on g different plane directions can be solved in $O(g^2 n^2 \log n)$ time.*

In the general case is when our triangles lie on $g = O(n)$ different families of parallel planes. In this case we do not gain anything using a reference plane P at infinity. The arrangement $\mathcal{M}$ has complexity $O(n^2)$. For each face $\sigma \in \mathcal{M}$ we use the same method described in the proof of Theorem (1) and obtain an overall $O(n^4 \log n)$ complexity, which is no better than the algorithm by McKenna and O'Rourke [MO88].

A further special case is when the stabbers are constrained to pass through a line $\mathcal{L}$. Our method works in this case if we assume that there is a plane in the pencil of planes containing $\mathcal{L}$ that does not intersect any triangle in T. And we choose this plane as P.

Now we need to consider only the regions of $\mathcal{M}$ intersecting $\mathcal{L}$.[8] Since we have only a linear number of subproblems the total time complexity is $O(n^3 \log n)$.

3.3 A Lower bound for the set of stabbers

Let us consider the arrangement of hyperplanes in 5 dimensions generated by mapping lines supporting triangles into Plücker hyperplanes. Intersecting this arrangement with the Plücker hypersuperface Π, we obtain a 4 dimensional arrangement of hyperbolic surfaces and a decomposition of the 4-space into cells. The set of Plücker points corresponding to stabbing lines is the union of some of these cells. We are interested in finding lower and upper bounds to the combinatorial complexity of these cells. For this purpose it is enough to bound the number of "corners" (i.e. 0-dimensional faces) of the cells.[9] The number of corners corresponds to the number of combinatorially different ways of choosing 4 edges of the triangles such that there is a stabber touching those edges.

---

[8] A second consequence is that the number of Plücker coordinates of our systems is now 4, since we force the apex q to belong to $\mathcal{L}$. This does not change the asymptotic complexity of the construction of the Plücker polytopes.

The paper [CEGS89] (Theorem 2) describes a lower bound to the complexity of the upper envelope of a set of lines which can easily be transformed into a construction of triangles, giving an $\Omega(n^3)$ bound on the descriptive complexity of the set of stabbers. The construction comprises three sets of lines such that, for any choice of three lines, one from each set, there is a fourth line (not in the original sets) that touches them and is above all the lines in the three sets.

We generate the same three sets of lines as in [CEGS89] and then we complete them into triangles by adding 2 steep almost vertical lines above each original line, making sure that all the features of the upper envelope of lines are kept within the triangles. This bound is for general non-disjoint triangle; no better bound is known for disjoint triangles. In the following of the section we give a second lower bound construction.

A (non horizontal) line l_i in 3-space is given by the equations:

$$x = a_i z + b_i \tag{10}$$

$$y = c_i z + d_i \tag{11}$$

Equation 10 is the projection of the line l on the xz-plane, equation 11 is the projection on the yz-plane.

On the xz-plane we can create a set of $O(n)$ segments S_{xz} such that the set of the stabbing lines for S_{xz} has n components. Using the dual method described in [EMP$^+$82] we draw on the dual plane n double wedges[10] with the apices $(w, 0)$, with $w = 1, \ldots, n$ and slopes -1 and $+1$. A second set of double wedges has apices $(w, -1)$ for $w = 1, \ldots, n$ and slopes -100 and $+100$.

The intersection of the double wedges, representing the dual set of stabbing lines has n disjoint components. Let $\{C_j\}_{j=1,\ldots,n}$ be the set of such component on the dual xy-plane. Similarly, we have a set S_{yz} and n components $\{C'_h\}_{h=1,\ldots,n}$ on the dual yz-plane.

Our goal is to find n^2 lines $\{l_i\}$ in 3D with the property that each line, when projected, maps in a different couple of components (C_j, C'_h).

The sequence of coefficients $\{a_i\}_{i=0,\ldots n^2-1}$ is an increasing sequence of n^2 numbers. We associate with the coefficient a_i the coefficient $c_i = a_u$, where $u = (i \bmod n)n + \lfloor i/n \rfloor$. The coefficients b_i and d_i will be chosen later inside an arbitrary interval. We choose the segments S_{xz} so that n dual points (a_i, b_i) are in each class C_j. Similarly we choose the segments S_{yz}. It is easy to check that the goal is reached by this construction.

---

[9] This can easily be proved by considering the complexity of the cells as the sum of the complexity of the 3-dimensional faces on each hypersurface. In 3-D the complexity of the cell is proportional to the number of vertices. Assuming general position, each vertex belongs to only 4 hypersurfaces.

[10] A double wedge is the region defined by two lines through a point called *apex* and not containing any vertical line.

Now for each of the n^2 lines in $\{l_i\}$ generated at the previous step we build n other lines $\{l'_{ik}\}_{k=1,\dots,n}$ with the same coefficients $a_{ik} = a_i$ and $c_{ik} = c_i$ but with different coefficients b_{ik} and d_{ik}.

The pair b_{ik}, d_{ik} is the intersection point of the line $\{l_{ik}\}$ with the plane $z = 0$. On the plane $z = 0$ we draw a convex polygon P with n edges, labeled with the integers $k = 1, \dots, n$; on each edge we select n^2 points, labeling them with the index $i = 0, \dots, n^2 - 1$. The coefficients b_{ik} and d_{ik} are the coordinates of the point labeled (i, k).

We extend orthogonally the segments S_{xz} and S_{yz} to infinite strips in 3-space and we move each of the n^3 lines $\{l'_{ik}\}$, keeping contact with the polygon P, until it touches 2 edges of the infinite strips. The fact that the generating lines $\{l_i\}$ are in separate components ensures that each the lines l'_{ik} touch $\Omega(n^3)$ different triplets of edges in our construction.

At the end we substitute the polygon P with n triangles whose intersection is P. Each strip is substituted by 2 triangles containing all the contact points between the lines l'_{ik} and the strips. In total there are $O(n)$ triangles.

Theorem 3 *Given n triangles in 3 space there is an $\Omega(n^3)$ lower bound on the number of extremal stabbing lines.*

Note that the triangles in our construction are not in general position.

A trivial upper bound on the number of extremal stabbing lines is given by the number of vertices in the 4-dimensional arrangement of surfaces, that is $O(n^4)$.

4 General query problem

Let us again consider the construction used in Section 3. We notice that the arrangement $\mathcal{M}$, induced on the reference plane P by the set *aff*(T) of planes spanning triangles in T, partitions P into regions σ, and that the intersection of the family of cones based on T and with apex in σ is given by one system of linear inequalities.

Let us take a random sample of the set *aff*(T); this sample induces a planar arrangement A_R on P. We triangulate A_R, obtaining the triangulated map $\triangle(A_R)$. From random sampling theory [Cla87] we know that each simplex $\sigma \in \triangle(A_R)$ is intersected by no more than $(n/r)\log r$ supporting planes. This means that for each point in σ we can define the stabbing conditions using a unique system of inequalities for a set T_σ of at least $(1 - \frac{\log r}{r})n$ triangles. Then we consider the Plücker polytope C_σ relative to the region σ and the set of triangles T_σ and preprocess it so as to be able to quickly deciding if a given point belongs to the polytope.

We recurse the construction on each simplex in $\triangle(A_R)$ for the lines defined by planes $aff(t)$ passing through the simplices.

The procedure for point location in a polytope[11] is a modification of Clarkson's random sampling method for performing point location in an arrangement of n hyperplanes in E^d ([Cla87]) (see also [CEGS89]). This method selects a random sample G of r (r constant) hyperplanes, considers the arrangement $A(G)$ generated by G and produces its triangulation $\triangle(A(G))$. This construction is then carried on recursively on the set of hyperplanes intersecting each simplex of the triangulation. The triangulation step has a complexity proportional to the number of vertices in $A(G)$.

---

[11] In the query problem we do not have to consider the Π-condition since the query point lies on Π.

We use the same general approach, however the recursive step is not on the whole arrangement $A(G)$ but only on one cell corresponding to the intersection of the half-spaces forming our polytope. The complexity of the triangulation, the time and storage needed to construct it are proportional to the number of vertices of the polytope [Cla87] that is $O(r^{\lfloor d/2 \rfloor}) = O(r^2)$ for the upper bound theorem (see [Ede87] Chapter 6, Theorem 6.12). The function $T(n)$ which is the maximum storage and preprocessing time satisfies:

$$T(n) \leq O(r^2) \times T((n \log r)/r) + O(r) \tag{12}$$

The solution is $O(n^{2+\epsilon})$. We then have logarithmic query time using almost quadratic storage and preprocessing.

Coming back to the stabbing problem, and denoting by $S(n)$ the total preprocessing space (and time) for the stabbing query, we obtain the following recurrence equation:

$$S(n) = r^2 S((n/r) \log r) + O(r^2 n^{2+\epsilon}) \tag{13}$$

whose solution is $S(n) = O(r^2 n^{2+\epsilon})$, for a different, slightly greater, ϵ.

The query algorithm is the following: given the line l, consider the point $q = l \cap P$ and locate it in a simplex $\sigma \in \triangle(A_R)$; then locate the Plücker point p_l in the associated Plücker polytope C_σ. If the point p_l is external to $\mathcal{C}_\sigma$ the line l is not a stabber; otherwise, we recurse the query on the data structure associated with the triangles whose spanning planes cross σ. The depth of the recursion is at most $\log_r n$; choosing $r = O(n^\nu)$, the depth is constant. The preprocessing time (and space) bound is the same with a slightly greater value for ϵ. We can summarize the above discussion in the following Theorem:

Theorem 4 *There exists an algorithm for answering line stabbing queries on a set of triangles in 3 dimensions with $O(n^{2+\epsilon})$ expected randomized preprocessing time and space and $O(\log n)$ worst case query time.*

4.1 Remarks on stabbing

The stabbing query algorithm can overcome the lower bound on the complexity of the set of stabbers because the set is not computed explicitly, but it is expressed as the intersection of sets with small complexity. The stabbing query is a *decomposable* problem according to the definition of Mehlhorn ([Meh84], page 4), since l is a stabber for the set $T = F_1 \cup F_2$ if and only if it is a stabber for F_1 and F_2. This fact allows us to use an implicit representation of the solution space in order to correctly answer the query.

5 Stabbing of Polyhedra

Definition 2 *Given a point a and a convex polyhedron B in 3-space the cone $\mathcal{C}_{a,B}$ is the set of rays from a intersecting B.*

A family $\mathcal{B}$ of polyhedra and a point a define a family of cones $\mathcal{C}_{a,\mathcal{B}} = \{\mathcal{C}_{a,B} | B \in \mathcal{B}\}$. We again assume that the reference plane P leaves all polyhedra in $\mathcal{B}$ on one side. It is easy to prove the following lemma:

Lemma 2 *There exists a stabbing line for $\mathcal{B}$ if and only if there is a point q on P such that $\bigcap \mathcal{C}_{q,\mathcal{B}} \neq \emptyset$.*

A point q external to the polyhedron B can see only a connected subset of the faces of B. Let us call the *silhouette* of B from q ($sil(q, B)$) the set of edges of B adjacent to a face visible from q and to a face not visible from q. It is easy to show that $sil(q, B)$ is a connected closed polygonal chain in 3-space. A plane defined by q and an edge in $sil(q, B)$ leaves B in one of the two halfspaces (which we call *positive*). It is easy to show that the set of points on rays in $\mathcal{C}_{q,\mathcal{B}}$ is the intersection of the positive halfspaces determined by q and $sil(q, B)$. Now we allow q to vary on P; the silhouette $sil(q, B)$ can change only when q crosses a line l_f that is the intersection of P with the plane spanning a facet f of B (which we denote with $aff(f)$).

On the plane P we consider the planar map $\mathcal{M}$ induced by the set of planes $aff(\mathcal{B}) = \{aff(f) \mid f \textit{ facet in } \mathcal{B}\}$. Clearly, for a fixed B, in each region $\sigma \in \mathcal{M}$ each $q \in \sigma$ sees the same silhouette $sil(q, B)$; therefore we can express the stabbing condition for all $q \in \sigma$ as the intersection of a set of positive halfspaces, i.e. as the solution to a system of inequalities. As in Section 3.1 these inequalities can be expressed as linear inequalities in the Plücker coordinates of lines spanning edges in $\mathcal{B}$ and of the (variable) line through q. Note that we have to spend some extra time in order to compute incrementally the silhouettes, but this time is asymptotically dominated by the time to create the map $\mathcal{M}$. It is now easy to extend Theorem 2 for polyhedra.

Theorem 5 *The existence problem for a set of polyhedra of total complexity n whose facets lay on g different plane directions can be solved in $O(g^2 n^2 \log n)$ time.*

Note that the case of isothetic boxes is a specialization of Theorem 5 with $g = 3$.

The algorithm for the query problem presented in Section 4 is also extended to deal with polyhedra. We add a preliminary phase in which we compute the the silhouettes relative to every region $\sigma \in \mathcal{M}$. Then we consider only a random sample R of the facets of the polyhedra and the induced arrangement A_R on the plane P. For each simplex $\sigma \in \triangle(A_R)$ we know for all edges except at most $O(n \log r/r)$ of them if they belong to a silhouette $sil(q, B)$ for some $B \in \mathcal{B}$ and any $q \in \sigma$.

As in Section 4 we can construct the Plücker polytope that express the stabbing condition for the edges in the silhouettes and recurse in each region σ. We conclude:

Theorem 6 *There exists an algorithm for answering line stabbing queries on a set of polyhedra in 3 dimensions with total complexity n, that requires $O(n^{2+\epsilon})$ expected randomized preprocessing time and space, with $O(\log n)$ worst case query time.*

6 Ray shooting

A generalization of the stabbing query problem follows. Preprocess the set T so as to quickly answer queries of the form: given a line l report the number of triangles intersected by l. In [MO88] this problem is solved with $O(n^4\alpha(n))$ preprocessing and $O(n)$ query time. For our purposes this method is equivalent to the trivial solution of checking each triangle in turn against the query line.

The fact that all the information we need is contained in a 5 dimensional arrangement suggests the following algorithm. Given a set H of n hyperplanes in R^5, we use the method of Edelsbrunner, O'Rourke and Seidel [EOS86] [Ede87] to generate the facial lattice of the arrangement $A(H)$ in $O(n^5)$ time and space. Within the same time bound we can check which cells intersect the Plücker Hypersuperface Π (these are the *useful cells*), and record one Plücker point in each useful cell.

Then we consider the graph G whose nodes are the useful cells of $A(H)$; two nodes labeled with adjacent cells (i.e., they share a common 4-face) are connected with an edge. We generate a spanning tree of G and an Eulerian tour of the spanning tree. Following the schedule given by the tour we visit the cells and keep an auxiliary dictionary holding the set of the triangles intersected by a line whose Plücker point lies in the cell. Moving to the next cell implies inserting or deleting at most one triangle; this operation takes at most $O(\log n)$ time for each update. For each cell we record the number of elements in the dictionary.

It is possible to show that the number of useful cells is $O(n^4)$ and therefore this phase requires $O(n^4 \log n)$ time. The same arrangement $A(H)$ is now preprocessed for point location using Clarkson's technique in $O(n^{5+\epsilon})$ randomized expected time [Cla87]. Then we associate the informations stored in $A(H)$ during the first phases with the representation of $A(H)$ for point location. The discussion above proves the following lemma:

Lemma 3 *Given n triangles in R^3 one can preprocess them in $O(n^{5+\epsilon})$ randomized expected time into a data structure of size $O(n^{5+\epsilon})$, so that given any line l one can determine in $O(\log n)$ worst case time which is the number of triangles hit by l.*

Now, we trade off preprocessing and query time. We divide T into $n^{1-\beta}$ sets of size n^β. We build the 5-dimensional arrangement for each set; we can perform the query in each one in logarithmic time. The preprocessing time is $O(n^{1+4\beta+\epsilon})$ to construct the 5-dimensional arrangement and store the relevant information in every cell; the query time is $O(n^{1-\beta}\log n)$. For $\beta = 0.5$ we obtain an $O(n^{3+\epsilon})$ preprocessing and $O(\sqrt{n}\log n)$ query method. Therefore the following theorem holds:

Theorem 7 *Given n triangles in R^3 one can preprocess them in $O(n^{3+\epsilon})$ randomized expected time into a data structure of size $O(n^{3+\epsilon})$, so that given any line l, the number of triangles intersecting l can be computed in $O(\sqrt{n}\log n)$ worst case query time.*

Constructing 5-dimensional arrangements of hyperplanes takes time $O(n^5)$. We noticed above that the number of useful cells is $O(n^4)$. Therefore we can store in each cell a list of the triangles stabbed by lines whose Plücker points are in the cell. When the triangles in T are *disjoint* there is a unique order in which the triangles can be stabbed by a point (i.e., line) in each cell. Each list will store the triangles in the proper stabbing order. The total time and space is still $O(n^5)$. Given a query ray ρ from the point p supported by the line l_ρ, we locate the line l_ρ in one cell of the arrangement, then with a simple binary search we compare the point p with the intersections of l_ρ

and the triangles in the list associated with the cell. This method suffices to detect the first triangle hit by the ray.

Lemma 4 *Given n disjoint triangles in R^3 one can preprocess them in $O(n^{5+\epsilon})$ randomized expected time into a data structure of size $O(n^{5+\epsilon})$, so that given any query ray ρ one can determine in $O(\log n)$ worst case time which is the first triangle hit by ρ.*

We can reduce the space used, even if not asymptotically, using persistent trees [ST86] or the similar list technique of Cole [Col86]. Trading preprocessing and query time we obtain an $O(n^{3+\epsilon})$ preprocessing time and space, $O(\sqrt{n}\log n)$ query time.

Theorem 8 *Given n disjoint triangles in R^3 one can preprocess them in $O(n^{3+\epsilon})$ randomized expected time into a data structure of size $O(n^{3+\epsilon})$, so that given any query ray ρ one can determine in $O(\sqrt{n}\log n)$ worst case time which is the first triangle hit by ρ.*

7 Batched ray shooting

If the rays are given in advance we solve the ray shooting queries more efficiently using a batched technique [GOS89]. Suppose we have n triangles and m rays; for $m \geq n^5$, we use the on-line query method at an overall cost $O(m \log n)$. When $m < n^5$ we recast the problem in a dual space. Each triangle is mapped into 3 points in 5-space and a line supporting a ray is mapped into an hyperplane. [12] We select a sample of r hyperplanes, produce the 5-dimensional arrangement and triangulate it. We obtain $M = O(r^5)$ cells. Let us consider now a triple of cells $\sigma = (\sigma_1, \sigma_2, \sigma_3)$. From the randon sampling theory [Cla87], at least $m(1 - c\log r/r)$ hyperplanes do not intersect σ; therefore they have a definite position with respect to any point in σ.

There are only a constant number of combinatorially different relative positions of an hyperplane and 3 regions σ_1, σ_2 and σ_3. Some of these positions describe the fact that the corresponding line hits a triangle that maps into points in σ (stabbing position). We consider a function $F_1(h)$ that for every hyperplane h gives us the triples σ in stabbing position with h; and a function $F_2(t)$ that for every triangle t gives us the triple σ in which it is mapped[13]. We find the inverse function $F_2^{-1}(\sigma)$ that for every triple σ produces the list of triangles mapped into σ in stabbing order (this order is unique and can be precomputed in overall $O(n \log n)$ time). The functions F_1 and F_2^{-1} are built using $O(n+m)$ space and $O(m + n\log n)$ time. With this data structure we access, for each ray, the lists of triangles hit by the supporting line. For each ray we perform a logarithmic search of the source of the ray over the lists of the triangles to locate the first one hit by the ray. This phase takes overall $O(m \log n)$ time.

Than we recurse in each cell on the hyperplanes cutting into the cell and on the points that define triangles with at least one point in the cell.

---

[12] This passage to dual space is necessary to exploit the first case as base of the recursive algorithm. It is not clear how to obtain a similar result in primal space.

[13] If t is mapped into two or one regions we add couple and singletons in the domain of the function F_1 and F_2.

The total space used by the algorithm is $O(n+m)$ since the functions F_1 and F_2^{-1} use linear space and we just need to record the closest triangle for each ray when we pass to a different level in the recursion.

The time complexity satisfies the following recursive equation:

$$T(m,n) \le \begin{cases} am\log n & \text{for } m \ge n^5 \\ \Sigma_{i=1}^{M} T(n_i, m_i) + cM^3 m \log n + bn\log n & \text{for } m < n^5 \end{cases} \tag{14}$$

The additional constraints are:

$$\begin{aligned} &\Sigma_{i=1}^{M} n_i \le 3n \\ &m_i \le c' m \log r / r \text{ for some constant } c' \end{aligned} \tag{15}$$

Theorem 9 *One can determine the first triangles hit by m rays in a set of n disjoint triangles using a randomized algorithm whose expected running time is bounded by*

$$[Dm^{5/6-\delta} n^{5/6+5\delta} \log^2 n + Am\log^2 n + Bn\log n \log m]$$

for any $\delta > 0$, where the coefficients A, B and D depend on δ. The storage is bounded by $O(m+n)$.

Proof. The proof is follows a schema in [EGS88b].

Fix δ and choose $r = r(\delta)$ to be sufficiently large (how large will be determined later in the proof). If $m \ge n^5$ then $T(m,n) \le am\log^2 n$ satisfies the bound assuming $A \ge a$. Suppose $m \le n^5$. In this case

$$m = m^{5/6-\delta} m^{1/6+\delta} \le m^{5/6-\delta} n^{5/6+5\delta}$$

First notice that at each level of the recursion the third term in (14) contributes $O(n\log n)$ on one level of the recursion; and there are at most $O(\log m)$ levels. The overall contribution of this term is $O(n\log n\log m)$. It is sufficient to drop this term from the recursion (14) and prove the modified disequation satisfies the bound

$$T(m,n) \le Dm^{5/6-\delta} n^{5/6+5\delta} \log^2 n + Am\log^2 n$$

.

By induction hypothesis we have

$$T(m,n) \le \Sigma_{i=1}^{M} [Dm_i^{5/6-\delta} n_i^{5/6+5\delta} + Am_i] \log^2 n_i + cM^3 m\log n$$

But $\Sigma_{i=1,M} m_i \log^2 n_i \le [Mc' \log r/r] m \log^2 n$, therefore we can sum up two terms in m, put $d = AMc'\log r/r + cM^3$, and use the inequality 15, obtain

$$T(m,n) \le [D\Sigma_{i=1}^{M} m_i^{5/6-\delta} n_i^{5/6+5\delta} + dm^{5/6-\delta} n^{5/6+5\delta}] \log^2 n$$

Now we eliminate the summation using first the bound on m_i and then the Hölder-Minkowsky inequality [Mit70].

$$\Sigma_{i=1}^{M} m_i^{5/6-\delta} n_i^{5/6+5\delta} \le [(c'\log r/r)m]^{5/6-\delta} \Sigma_{i=1}^{M} n_i^{5/6+5\delta} \le [c'\log r/r]^{5/6-\delta} m^{5/6-\delta} M^{1/6-5\delta} n^{5/6+5\delta}$$

Then we obtain, for some constant c''

$$T(m,n) \leq [Dc'' \log^{5/6-\delta} r / r^{24\delta} + d] m^{5/6-\delta} n^{5/6+5\delta} \log^2 n \qquad (16)$$

If we choose r sufficiently large so that the constant in the equation (16) is less than $[D/2 + d]$, and we choose $D = 2d$, we obtain

$$T(m,n) \leq D m^{5/6-\delta} n^{5/6+5\delta} \log^2 n$$

which proofs the asserted inequality. ∎

Using this result, when $m = n$ we can answer the ray shooting queries at a randomized expected cost $O(n^{2/3+4\delta})$ per ray. When $m = n^2$ the time cost is $O(n^{1/2+4\delta})$ per ray.

This batched technique provides a solution for the problem of deciding whether two non-convex polyhedra intersect. Two non convex polyhedra A and B intersect if and only if an edge of A intersects a facet of B or an edge of B intersects a facet of A or $A \subset B$ or $B \subset A$.

Modifying the batched ray shooting technique in order to deal with segment queries (instead of ray queries) we decide the first two cases; afterward, the inclusions can be decided in linear time.

Corollary 1 *Given two (non-convex) polyhedra of complexity m and n, there is an algorithm to decide whether they intersect that uses $O((n+m)^{5/3+4\delta})$ randomized expected time.*

When we solve a batched ray shooting problem for a set of n axis-oriented boxes we can obtain a better bound by using as base case in the recursion an $O(n^3)$ time/space $O(\log^3 n)$ query time method reported in [SML88]. Following the same proof schema we obtain:

Theorem 10 *One can determine the first box hit by m rays in a set of n disjoint axis-oriented boxes using a randomized algorithm whose expected running time is bounded by*

$$D m^{3/4-\delta} n^{3/4+3\delta} \log^4 n + A m \log^4 n + B n \log n \log m$$

for any $1/28 < \delta < 1/9$, where the coefficients A, B and D depend on δ. The storage is bounded by $O(n+m)$.

The proof follows the schema of the proof of Theorem 9. The only difference is that δ must be chosen between the values $1/28$ and $1/9$ and cannot be chose arbitrarily. The details of the proof are omitted.

8 Improved Batched Ray-Shooting

As noted in Section 6 the number of useful cells in an arrangement of n hyperplanes in R^5 is only $O(n^4)$. This fact can be exploited to obtain an improved on-line ray-shooting algorithm. If we can prove that the total complexity (number of faces of any dimensions) of $m = n^4$ cells is $O(n^\gamma)$ then it is possible to construct the data structure of Section 6 in $O(n^{\gamma+\epsilon})$ time and space. Note that now we *must* use a similar lists method to store the lists of triangles associated with the cells.

To derive such bound on the complexity of many cells we use bounds in [EGS88a] on the number of *facets* and we use the Minkowsky-Hölder inequality [Mit70] to apply that bound recursively in lower dimensional spaces. We get $\gamma = 4 + 2/3 + 1/27$ which improves the previous trivial bound $O(n^5)$.

We will use the following results from [EGS88a]:

Theorem 11 ([EGS88a]) *The maximum number of* faces *bounding m cells in an arrangement of n planes in R^3 is $O(m^{2/3} n \log n + n^2)$.*

Theorem 12 ([EGS88a]) *The maximum number of* facets *bounding m cells in an arrangement of n hyperplanes in R^d, $d > 3$ is $O(m^{2/3} n^{d/3} \log n + n^{d-1})$*

The main combinatorial result is the following theorem:

Theorem 13 *Given a set M of $m \geq n^{d-3/2}$ cells in an arrangement of n hyperplanes in R^d the total number of faces of any dimension bounding the cells in M is $O((m^{2/3} n^{d/3})^{(2/3)^{d-3}} n^{d-d(2/3)^{d-3}})$.*

Proof. Let $K(k)$ be the number of faces of dimension $k-1$ in the boundary of cells in M.

The faces of dimension k are facets of faces of dimension $k+1$, which are cells of a $k+1$-dimensional subspace of R^d.

Denoting with $C(k)$ the set of subspaces of dimension k in the arrangement and with m_s for $s \in C(k)$ the number of cells in the boundary of M and belonging to s, the following relation holds:

$$\Sigma_{s \in C(k)} m_s = O(K(k+1))$$

From Theorem 12 applied on all subspaces in $C(k)$ we obtain the following recursion:

$$K(k) = \Sigma_{s \in C(k)} (m_s^{2/3} n^{k/3} + n^{k-1}) \tag{17}$$

Using the Minkowsky-Hölder inequality[14] and the bound $|C(k)| = O(n^{d-k})$ we transform equation 17 and obtain:

$$K(k) = n^{d-1} + n^{d/3} [K(k+1)]^{2/3}$$

The total number of faces in the boundary of the m cells, which we want to bound, is $O(K(3))$. We compute $K(3)$ by telescoping:

$$K(3) = n^{d-1} + n^{d/3} [\underbrace{n^{d-1} + n^{d/3} [\ldots K(d)^{2/3}]^{2/3}}_{K(4)}]^{2/3}$$

If $m \geq n^{d-3/2}$ than the following holds:

---

[14] $\Sigma_i a_i^p b_i^q \leq (\Sigma_i a_i)^p (\Sigma_i b_i)^q$ with $p + q = 1$.

1. $K(d) = O(m^{2/3}n^{d/3})$
2. $K(d)^{2/3}n^{d/3} \geq n^{d-1}$
3. For every k,$d \leq k \leq 3$, $K(k)^{2/3}n^{d/3} \geq n^{d-1}$.

Therefore the telescoping formula can be simplified:

$$K(3) = n^{d/3}\underbrace{[n^{d/3}[\dots K(d)^{2/3}]^{2/3}]}_{K(4)}{}^{2/3}$$

Collecting the factors:

$$K(3) = K(d)^{(2/3)^{d-3}}\Pi_{i=0}^{d-4}(n^{d/3})^{(2/3)^i}$$

Appling the logarithm to the product and computing the resulting geometric series:

$$K(3) = K(d)^{(2/3)^{d-3}}n^{d-d(2/3)^{d-3}}$$

And since from Theorem 12:

$$K(d) = m^{2/3}n^{d/3} + n^{d-1}$$

we have

$$K(3) = (m^{2/3}n^{d/3})^{(2/3)^{d-3}}n^{d-d(2/3)^{d-3}}$$

■

Note that, for $m = n^d$, $K(3) = O(n^d)$ and ,for $m = n^h$ with $d > h > d - 3/2$, we have $K(3) = o(n^d)$. Recent results from Aronov [Aro] that eliminate log factors in the bounds of [EGS88a] can be used to eliminate the log factor in Theorem 13.

For the useful cells the following corollary is easily proved:

Corollary 2 *The complexity (number of faces) of n^4 cells in an arrangement of n hyperplanes in R^5 is $O(n^\beta \log^{4/9} n)$, where $\beta = 4 + 2/3 + 1/27$*

From Corollary 2 and the discussions of Sections 6 and 7 the following theorems hold:

Theorem 14 *Given n disjoint triangles in R^3 one can preprocess them in $O(n^{\beta+\epsilon})$ randomized expected time into a data structure of size $O(n^{\beta+\epsilon})$, so that given any query ray ρ one can determine in $O(\log n)$ worst case time which is the first triangle hit by ρ. Where $\beta = 4 + 2/3 + 1/27$.*

Theorem 15 *One can determine the first triangles hit by m rays in a set of n disjoint triangles using a randomized algorithm whose expected running time is bounded by*

$$[Dm^{\frac{\beta}{\beta+1}-\delta}n^{\frac{\beta}{\beta+1}+5\delta}\log^2 n + Am\log^2 n + Bn\log n\log m]$$

for any $\delta > 0$ and $\beta = 4 + 2/3 + 1/27$, where the coefficients A, B and D depend on δ. The storage is bounded by $O(m + n)$.

As a consequence of the last theorem $m = n$ implies a cost for ray $O(n^{0.6492})$, which improves the previous $O(n^{0.666})$ bound.

9 Acknowledgments

I wish to thank Prof. Micha Sharir, Prof. Richard Pollack, Dr. Peter Shor and Prof. Richard Cole for providing invaluable suggestions, discussions and improvements. I am indebted to Boris Aronov for many useful conversations.

References

[Aga89a] P.K. Agarwal. A deterministic algorithm for partitioning arrangements of lines and its applications. In *Proceedings of the 5th ACM Symposium on Computational Geometry*, pages 11–22, 1989.

[Aga89b] P.K. Agarwal. Ray shooting and other applications of spanning trees with low stabbing number. In *Proceedings of the 5th ACM Symposium on Computational Geometry*, pages 315–325, 1989.

[AK87] J. Arvo and D. Kirk. Fast ray tracing by ray classification. In M. C. Stone, editor, *SIGGRAPH '87 Conference Proceedings*, pages 55–63, 1987.

[Aro] B. Aronov. New bounds on the number of facets. Personal comunication.

[AW87] D. Avis and R. Wenger. Algorithms for line transversals in space. In *Proceedings of the 3rd annual Symposium on Computational Geometry*, pages 300–307, 1987.

[BDEG90] M. Bern, D. Dobkin, D. Eppstein, and R. Grossman. Visibility with a moving point. In *Proceedings of the 1st ACM-SIAM Symposium on Discrete Algorithms*, 1990.

[CEGS89] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Lines in space: combinatorics, algorithms and applications. In *Proc. of the 21st Symposium on Theory of Computing*, pages 382–393, 1989.

[Cla87] K.L Clarkson. New applications of random sampling in computational geometry. *Discrete Computational Geometry*, pages 195–222, 1987.

[Col86] R. Cole. Searching and storing similar lists. *Journal of Algorithms*, (7):202–220, 1986.

[CS86] R. Cole and M. Sharir. Visibility problems for polyhedral terrains. Technical Report 266, CIMS, December 1986.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer Verlag, 1987.

[EGS88a] H. Edelsbrunner, L. Guibas, and M. Sharir. The complexity of many cells in arrangemets of planes and related problems. Technical Report 184, Courant Institute-Robotics Lab., December 1988.

[EGS88b] H. Edelsbrunner, L. Guibas, and M. Sharir. The complexity of many faces in arrangements of lines and segments. In *Proceedings of the 4th ACM Symposium on Computational Geometry*, pages 44–55, 1988.

[EMP$^+$82] H. Edelsbrunner, H. Mauer, F. Preparata, E. Welzl, and D. Wood. Stabbing line segments. *BIT*, (22):274–281, 1982.

[EOS86] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM Journal of Computing*, pages 341–363, 1986.

[Gla89] A.S. Glassner, editor. *Ray tracing.* Academic Press, 1989.

[GOS89] L. Guibas, M. Overmars, and M. Sharir. Ray shooting, implicit point location and related queries in arrangements of segments. Technical Report TR433, Courant Institute, 1989.

[HP52] W.V.D. Hodge and D. Pedoe. *Methods of algebraic geometry.* Cambridge University press, 1952.

[HS86] S. Hart and M. Sharir. Non-linearity of davenport-schinzel sequences and of generalized path compression schemes. *Combinatorica*, pages 151–177, 1986.

[JK88] J. W. Jaromczyk and M. Kowaluk. Skewed projections with an application to line stabbing in r^3. In *Proceedings of the 4th annual Symposium on Computational Geometry*, pages 362–370, 1988.

[Meh84] K. Mehlhorn. *Multidimensional Searching and Computational Geometry.* Springer Verlag, 1984.

[Mit70] D.S. Mitrinovic. *Analytic inequalities.* Springer Verlag, 1970.

[MO88] M. McKenna and J. O'Rourke. Arrangements of lines in 3-space: A data structure with applications. In *Proceedings of the 4th annual Symposium on Computational Geometry*, pages 371–380, 1988.

[OS89] M. Overmars and M. Sharir. Output-sensitive hidden surface removal. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, 1989.

[Sei86] R. Seidel. Constructing higher-dimensional convex hulls at logarithmic cost per face. In *Proceedings of the 18th annual Symposium on Theory of Computing*, pages 404–413, 1986.

[SML88] A. Schmitt, H. Muller, and W. Leister. Ray tracing algorithms - theory and practice. In R.A. Earnshaw, editor, *Theoretical Foundations of Computer Graphics and CAD*, pages 997–1030. NATO ASI, Springer Verlag, 1988.

[Som51] D. M. H. Sommerville. *Analytical geometry of three dimensions.* Cambridge, 1951.

[ST86] N. Sarnak and R.E. Tarjan. Planar point location using persistent search trees. *Comm. ACM*, (29):669–679, 1986.

[Sto89] J. Stolfi. Primitives for computational geometry. Technical Report 36, Digital SRC, 1989.

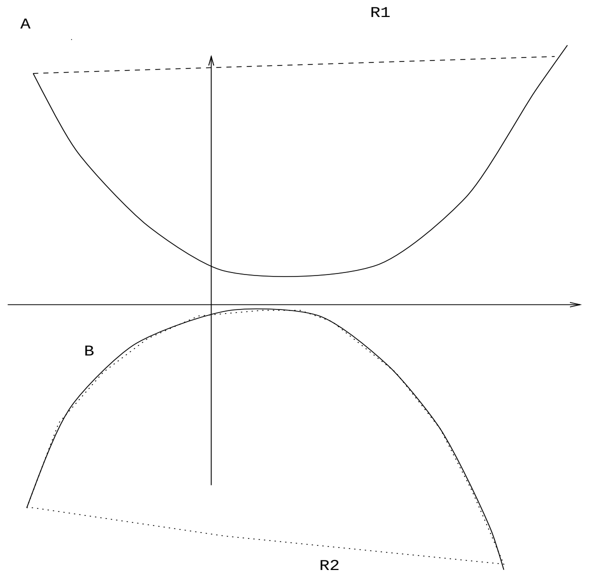

Figure 1: Regions not in between an upper and lower curve

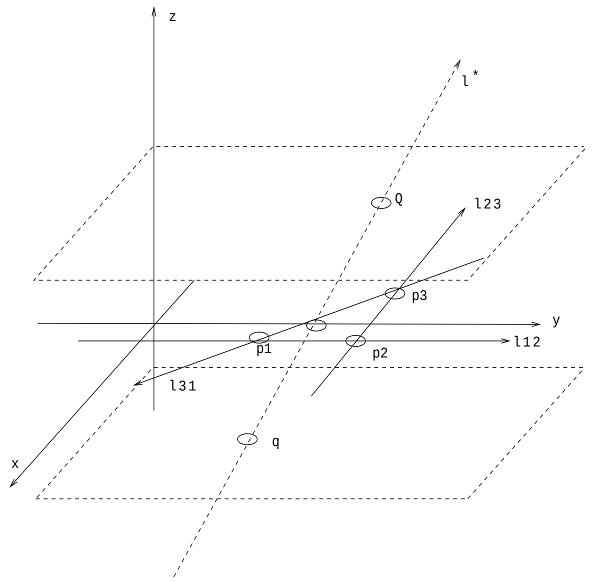
z
l*
Q
l23
p3
y
l12
p1
p2
l31
x
q

Figure 2: Stabbing line and triangle

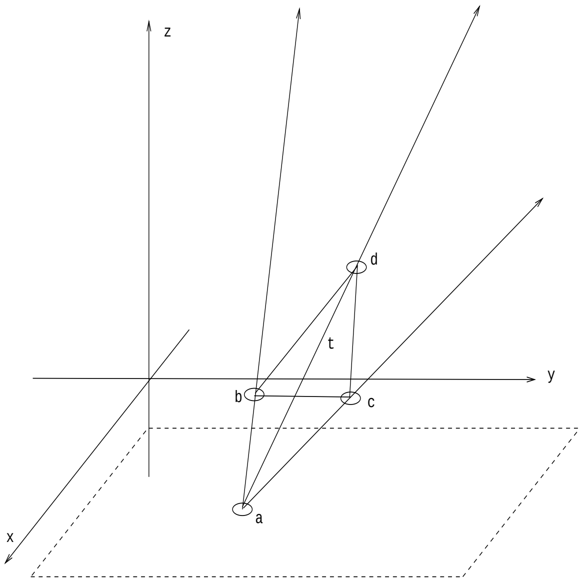

Figure 3: Cone with apex a based on t

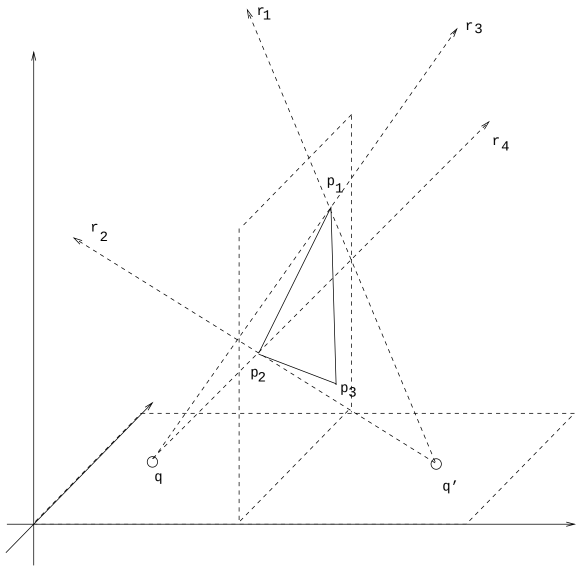
r1
r3
r4
p1
r2
p2
p3
q
q'

Figure 4: Inequalities for cones based at q and q'

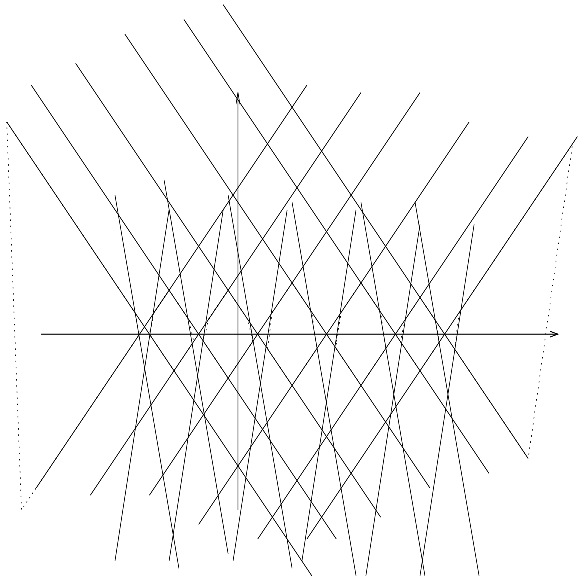

Figure 5: Dual set of stabbing lines with n components