32 RAY SHOOTING AND LINES IN SPACE

Marco Pellegrini

INTRODUCTION

The geometry of lines in 3-space has been a part of the body of classical algebraic geometry since the pioneering work of Plücker. Interest in this branch of geometry has been revived in recent years by several converging trends in computer science. The discipline of computer graphics has pursued the task of rendering realistic images by simulating the flow of light within a scene according to the laws of elementary optical physics. In these models light moves along straight lines in 3-space and a computational challenge is to find efficiently the intersections of a very large number of rays with the objects comprising the scene. In robotics the chief problem is that of moving 3-dimensional objects without collisions. Effects due to the edges of objects have been studied as a special case of the more general problem of representing and manipulating lines in 3-space. Computational geometry (whose core is better termed "design and analysis of geometric algorithms") has moved recently from the realm of planar problems to tackling directly problems that are specifically 3-dimensional. The new and sometimes unexpected computational phenomena generated by lines (and segments) in 3-space have emerged as a main focus of research.

In this chapter we will survey the present state of the art on lines and ray shooting in 3-space from the point of view of computational geometry. The emphasis is on provable nontrivial bounds for the time and storage used by algorithms for solving natural problems on lines, rays, and polyhedra in 3-space. We start by mentioning different possible choices of coordinates for lines (Section 32.1). This is an essential initial step because different coordinates highlight different properties of the lines in their interaction with other geometric objects. Here a special role is played by *Plücker coordinates* [Plu65] (Section 32.1), which represent the starting point for many of the most recent results. Then we consider how lines interact with each other (Section 32.2). We are given a finite set of lines L that act as obstacles and we will define other (infinite) sets of lines induced by L that capture some of the important properties of visibility and motion problems. We show bounds on the storage required for a complete description of such sets. Then we move a step forward by considering the same sets of lines when the obstacles are polyhedral sets, more commonly encountered in applications. We arrive in Section 32.3 at the ray-shooting problem and its variants (on-line, off-line, arbitrary direction, fixed direction, and shooting with objects other than rays). Again, the obstacles are usually polyhedral objects, but in one case we are able to report a ray-shooting result on curved objects (spheres).

Section 32.4 is devoted to the problem of collision-free movements (arbitrary or translation only) of lines among obstacles. This problem arises, for example, when lines are used to model radiation or light beams (e.g., lasers). In Section 32.5 we define a few notions of distance among lines, and as a consequence we have several

0-8493-8524-5/97/$0.00+$.50

natural proximity problems for lines in 3-space. Finding the closest pair in a set of lines is the most basic of such problems.

In Section 32.6 we survey what is known about the "dominance" relation among lines. This relation is central for many visibility problems in graphics. It is used, for example, in the painter's algorithm for hidden surface removal (Chapter 42). Another direction of research has explored the relation between lines in 3-space and their orthogonal projections. One main topic is that of realizability, that is, given a set of planar lines together with a relation, does there exist a corresponding set of lines in 3-space whose dominance is consistent with the given relation?

32.1 COORDINATES OF LINES

GLOSSARY

Homogeneous coordinates: A point (x, y, z) in Cartesian coordinates has homogeneous coordinates (x_0, x_1, x_2, x_3), where $x = x_1/x_0$, $y = x_2/x_0$, and $z = x_3/x_0$.

Oriented lines: A line may have two distinct orientations. A line and an orientation form an oriented line.

Unoriented line: A line for which an orientation is not distinguished.

(I) Canonical coordinates by pairs of planes. The intersection of two planes with equations $y = az+b$ and $x = cz+d$ is a nonhorizontal line in 3-space, uniquely defined by the four parameters (a, b, c, d). Thus these parameters can be taken as coordinates of such lines. In fact, the space of nonhorizontal lines is homeomorphic to $\mathbb{R}^4$. Results on ray shooting among boxes and some lower bounds on stabbing are obtained using these coordinates.

(II) Canonical coordinates by pairs of points. Given two parallel horizontal planes, $z = 1$ and $z = 0$, the intersection points of a nonhorizontal line l with the two planes uniquely define that line. If $(x_0, y_0, 0)$ and $(x_1, y_1, 1)$ are two such points for l, then the quadruple (x_0, y_0, x_1, y_1) can be used as coordinates of l. Results on sets of horizontal polygons are obtained using these coordinates.

Although four is the minimum number of coordinates needed to represent an *unoriented* line, such parametrizations have proved useful only in special cases. Many interesting results have been derived using instead a five-dimensional parametrization for *oriented* lines, called ***Plücker coordinates***.

(III) Plücker coordinates of lines. An oriented line in 3-space can be given by the homogeneous coordinates of two of its points. Let l be a line in 3-space and let $a = (a_0, a_1, a_2, a_3)$ and $b = (b_0, b_1, b_2, b_3)$ be two distinct points in homogeneous coordinates on l. We can represent the line l, oriented from a to b, by the matrix

$$l = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \end{pmatrix}, \quad \text{with } a_0, b_0 > 0.$$

By taking the determinants of the six 2×2 submatrices of the above 2×4 matrix we obtain the ***homogeneous Plücker coordinates*** of the line:

$$p(l) = (\xi_{01}, \xi_{02}, \xi_{03}, \xi_{12}, \xi_{31}, \xi_{23}), \quad \text{with} \quad \xi_{ij} = det \begin{pmatrix} a_i & a_j \\ b_i & b_j \end{pmatrix}.$$

The six numbers ξ_{ij} are interpreted as homogeneous coordinates of a point in 5-space. For a given line l the six numbers are unique modulo a positive multiplicative factor, and they do not depend on the particular distinct points a and b that we have chosen on l. We call $p(l)$ the ***Plücker point*** of l in projective 5-dimensional space $\mathbb{P}^5$.

We also define the ***Plücker hyperplane*** of the line l to be the hyperplane in $\mathbb{P}^5$ with vector of coefficients $v(l) = (\xi_{23}, \xi_{31}, \xi_{12}, \xi_{03}, \xi_{02}, \xi_{01})$. So the Plücker hyperplane is:

$$h(l) = \{p \in \mathbb{P}^5 \mid v(l) \cdot p = 0\}.$$

For each Plücker hyperplane we have a positive and a negative halfspace given by $h^+(l) = \{p \in \mathbb{P}^5 \mid v(l) \cdot p \geq 0\}$ and $h^-(l) = \{p \in \mathbb{P}^5 \mid v(l) \cdot p \leq 0\}$. Not every tuple of 6 real numbers corresponds to a line in 3-space since the Plücker coordinates must satisfy the condition

$$\xi_{01}\xi_{23} + \xi_{02}\xi_{31} + \xi_{03}\xi_{12} = 0. \tag{32.1.1}$$

The set of points in $\mathbb{P}^5$ satisfying Equation 32.1.1 forms the so-called ***Plücker hypersurface*** Π; it is also called the ***Klein quadric*** or the ***Grassmannian*** (manifold). The converse is also true: every tuple of six real numbers satisfying Equation 32.1.1 is the Plücker point of some line in 3-space. Given two lines l and l', they intersect or are parallel (i.e., they intersect at infinity) when the four defining points are coplanar. In this case the determinant of the 4×4 matrix formed by the 16 homogeneous coordinates of the four points is zero. In terms of Plücker coordinates we have the following basic lemmas.

LEMMA 32.1.1

Lines l and l' intersect or are parallel (meet at infinity) if and only if $p(l) \in h(l')$.

Note that Equation 32.1.1 states in terms of Plücker coordinates the fact that any line always meets itself.

LEMMA 32.1.2

Let l be an oriented line and t a triangle in Cartesian 3-space with vertices (p_0, p_1, p_2). Let l_i be the oriented line through (p_i, p_{i+1}) (indices mod 3). Then l intersects t if and only if either $p(l) \in h^+(l_0) \cap h^+(l_1) \cap h^+(l_2)$ or $p(l) \in h^-(l_0) \cap h^-(l_1) \cap h^-(l_2)$.

These two lemmas allow us to map combinatorial and algorithmic problems involving lines (and polyhedral sets) in 3-space into problems involving sets of hyperplanes and points in projective 5-space (Plücker space). The main advantage is that we can use the rich collection of results on the combinatorics of high dimensional arrangements of hyperplanes (see Chapter 21). The main drawback is that we are using five (nonhomogeneous) parameters, instead of four which is the minimum number necessary. This choice has a potential for increasing the time bounds of line algorithms. We are rescued by the following theorem:

THEOREM 32.1.3 [APS93]

Given a set H of n hyperplanes in 5-dimensional space, the complexity of the cells of the arrangement $\mathcal{A}(H)$ intersected by the Plücker hypersurface Π (also called the ***zone*** *of Π in $\mathcal{A}(H)$) is $O(n^4 \log n)$.*

Although the entire arrangement $\mathcal{A}(H)$ can be of complexity $\Theta(n^5)$, if we are working only with Plücker points we can limit our constructions to the zone of Π, the complexity of which is one order of magnitude smaller. Theorem 32.1.3 is especially useful for deriving ray-shooting results.

The list of coordinatizations discussed in this section is by no means exhaustive. Other parametrizations are used, for example, in [Ame92], [AAS97], and [AS96].

A TYPICAL EXAMPLE

A typical example of the use of Plücker coordinates in three-dimensional problems is the result for fast ray shooting among polyhedra (see Table 32.3.1). We triangulate the faces of the polyhedra and extend each edge to a full line. Each such line is mapped to a Plücker hyperplane. Lemma 32.1.2 guarantees that each cell in the resulting arrangement of Plücker hyperplanes contains Plücker points that pass through the same set of triangles. Thus to answer a ray-shooting query, we first locate the query Plücker point in the arrangement, and then search the list of triangles associated with the retrieved cell. This final step is accomplished using a binary search strategy when the polyhedra are disjoint. Theorem 32.1.3 guarantees that we need to build a point location structure only for the zone of the Plücker hypersurface, thus saving an order of magnitude over general point location methods for arrangements (see Sections 21.3 and 30.7).

32.2 SETS OF LINES IN 3-SPACE

With Plücker coordinates (III) to represent oriented lines, we can use the topology induced by the standard topology of 5-dimensional projective space $\mathbb{P}^5$ on Π as a natural topology on sets of oriented lines. Using the four-dimensional coordinatizations (I) or (II), we can impose the standard topology of $\mathbb{R}^4$ on the set of nonhorizontal unoriented lines. Thus we can define the concepts of "neighbourhood," "continuous path," "open set," "closed set," "boundary," "path-connected component," and so on, for the set $\mathcal{L}$ of lines in 3-space. The distinction between oriented lines and unoriented lines is mainly technical and the complexity bounds hold in either case.

GROUPS OF LINES INDUCED BY A FINITE SET OF LINES

GLOSSARY

Semialgebraic set: The set of all points that satisfy a Boolean combination of a finite number of algebraic constraints (equalities and inequalities) in the Cartesian coordinates of $\mathbb{R}^d$. See Chapter 29.

Path-connected component: A maximal set of lines that can be connected by a path of lines, a continuous function from the interval $[0,1]$ to the space of lines.

Positively-oriented lines: Oriented lines l'_1 and l'_2 on the xy-plane are positively-oriented if the triple scalar product of vectors parallel to l'_1, l'_2, and the positive z-axis is positive.

Consistently-oriented lines: An oriented line l in 3-space is oriented consistently with a 3-dimensional set L of oriented lines if the projection l' of l onto a plane is positively-oriented with the projection of every line in L.

A finite set L of n lines in 3-space can be viewed as an obstacle to the free movement of other lines in 3-space. Many applications lead to defining groups of lines with some special properties with respect to the fixed lines L. The resources used by algorithms for these applications are often bounded by the "complexity" of such groups.

The boundary of a semialgebraic set in $\mathbb{R}^4$ is partitioned into a finite number of faces of dimension 0, 1, 2, and 3, each of which is also a semialgebraic set. The number of faces on the boundary of a semialgebraic set is the ***complexity*** of that set. The groups of lines that we consider are represented in $\mathbb{R}^4$ by semialgebraic sets, with the coefficients of the corresponding algebraic constraints a function of the given finite set of lines L.

The set Miss(L) consists of lines that do not meet any line in L. The sets Vert(L) and Free(L) consists of lines that may be translated to infinity without collision with lines in L. The basic complexities displayed in Table 32.2.1 are derived from [CEGS96, Pel94b, Aga94].

TABLE 32.2.1 Complexity of groups of lines defined by lines.

SET OF LINES	DEFINITION	COMPLEXITY
Miss(L)	do not meet any line in L	$\Theta(n^4)$
1 component of Miss(L)	1 path-connected component	$\Theta(n^2)$
Vert(L)	can be translated vertically to ∞	$\Theta(n^3)$
Free(L)	can be translated to ∞ in some direction	$\Omega(n^3)$,$O(n^3 \log n)$
VertCO(L)	above L and oriented consistently with L	$\Theta(n^2)$

MEMBERSHIP TESTS

When we are given L we can build a data structure during a preprocessing phase so that when we are given a new (query) line l we can decide very efficiently whether l is in one of the sets defined in the previous section. Such an algorithm implements a membership test for a group of lines. Table 32.2.2 shows the main results.

GROUPS OF LINES INDUCED BY POLYHEDRA

GLOSSARY

ϵ: A positive real number, which we may choose arbitrarily close to zero for each

TABLE 32.2.2 Membership tests for groups of lines defined by lines.

SET OF LINES	QUERY TIME	PREPROC/STORAGE	SOURCE
$\mathrm{Miss}(L)$	$O(\log n)$	$O(n^{4+\epsilon})$	[Pel93b, AM93]
1 component of $\mathrm{Miss}(L)$	$O(\log n)$	$O(n^{2+\epsilon})$	[Pel91]
$\mathrm{Vert}(L)$, $\mathrm{VertCO}(L)$	$O(\log n)$	$O(n^{2+\epsilon})$	[CEGS96]
$\mathrm{Free}(L)$	$O(\log n)$	$O(n^{3+\epsilon})$	[Pel94b]

algorithm or data structure. A caveat is that the multiplicative constant implicit in the big-O notation depends on ϵ and its value increases when ϵ tends to zero.

$\alpha(\cdot)$: The inverse of Ackerman's function. $\alpha(n)$ grows very slowly and is at most 4 for any practical value of n. See Section 40.4.

$\beta(\cdot)$: $\beta(n) = 2^{c\sqrt{\log n}}$ for a constant c. $\beta(\cdot)$ is a function that is smaller than any polynomial but larger than any polylogarithmic factor. Formally we have that for every $a, b > 0$, $\log^a n \leq \beta(n) \leq n^b$ for any $n \geq n_0(a, b)$.

Polyhedral set P: A region of 3-space bounded by a collection of interior-disjoint vertices, segments, and planar polygons. We denote with n the total number of vertices, edges, and faces.

Star-shaped polyhedron: A polyhedron P for which there exists a point $o \in P$ such that for every point $p \in P$, the open segment op is contained in P.

Terrain: When the star-shaped polyhedron is unbounded and o is at infinity we obtain a terrain, a monotone surface (cf. Section 23.1).

A collection of polyhedra in 3-space may act as obstacles limiting the collision-free movements of lines. Thus, following the blueprint of the previous section, we consider the complexity of some interesting groups of lines induced by polyhedra, displayed in Table 32.2.3 (see [HS94, Pel94b, Aga94]).

TABLE 32.2.3 Complexity of groups of lines defined by polyhedra.

SET OF LINES	DEFINITION	COMPLEXITY
$\mathrm{Miss}(P)$	do not meet polyhedron P	$\Theta(n^4)$
$\mathrm{Vert}(P)$	can be translated vertically to ∞	$\Omega(n^3)$, $O(n^3\beta(n))$
$\mathrm{Free}(P)$	can be translated to ∞ in some direction	$\Omega(n^3)$
$\mathrm{Miss}(Q), \mathrm{Free}(Q)$	Q star-shaped polyhedron or a terrain	$\Omega(n^2\alpha(n))$, $O(n^3 \log n)$

Similarly, we can define groups of *3D segments* defined by polyhedra in 3D. The set of relatively open segments that miss P is also a semi-algebraic set, known as the *3D Visibility skeleton* (see [DDP97, Dur99]). Its combinatorial complexity is $\Theta(n^4)$.

OPEN PROBLEMS

1. Find an almost cubic upper bound on the complexity of the group of lines Free(P) for a polyhedron P.
2. Close the gap between the quadratic lower and the cubic upper bound for the group Free(T) induced by a terrain T (Table 32.2.3).

SETS OF STABBING LINES

GLOSSARY

Stabber: A line l that intersects every member of a collection $\mathcal{P} = \{P_1, ..., P_k\}$ of polyhedral sets. The sum of the sizes of the polyhedral sets in $\mathcal{P}$ is n. The set of lines stabbing $\mathcal{P}$ is denoted $S(\mathcal{P})$.

Box: A parallelopiped each of whose faces is orthogonal to one of the three Cartesian axes.

c-oriented: Polyhedra whose face normals come from a set of c fixed directions.

Table 32.2.4 lists the worst-case complexity of the set $S(\mathcal{P})$ and the time to find a witness stabbing line.

TABLE 32.2.4 Complexity of the set of stabbing lines and detection time.

OBJECTS	COMPLEXITY OF $S(\mathcal{P})$	FIND TIME	SOURCES
Convex polyhedra	$\Omega(n^3)$, $O(n^3 \log n)$	$O(n^3\beta(n))$	[PS92, Pel93a, Aga94]
Boxes	$O(n^2)$	$O(n)$	[Ame92, Meg91]
c-oriented polyhedra	$O(n^2)$	$O(n^2)$	[Pel91]
Horiz polygons	$\Theta(n^2)$	$O(n)$	[Pel91]

Note that in some cases (boxes, parallel polygons) a stabbing line can be found in linear time, even though the best known bound on the complexity of the stabbing set is quadratic. These results are obtained using linear programming techniques (Chapter 38).

We can determine whether a given line l is a stabber for a preprocessed set $\mathcal{P}$ of convex polyhedra in time $O(\log n)$, using data structures of size $O(n^{2+\epsilon})$ that can be constructed in time $O(n^{2+\epsilon})$. [PS92].

For an *oriented* stabber l and a set $\mathcal{O}$ of k *disjoint* convex bodies in R^d, the order of the intersection of the objects along l is called a *geometric permutation*. A recent advance of Zhou and Suri [ZS01] shows that for balls of unit radius and k large enough there are at most 4 geometric permutations. For k rectangular boxes there are at most 2^{d-1} geometric permutations, which is tight (see also [Oro01]).

OPEN PROBLEMS

1. Can linear programming techniques yield a linear-time algorithm for c-oriented polyhedra?
2. The lower bound for $S(\mathcal{P})$ for a set of pairwise *disjoint* convex polyhedra is only $\Omega(n^2)$ [PS92]. Close the gap between this and the cubic upper bound.

32.3 RAY SHOOTING

Ray shooting is an important operation in computer graphics and a primitive operation useful in several geometric computations (e.g., hidden surface removal, and detecting and computing intersections of polyhedra). The problem is defined as follows. Given a large collection $\mathcal{P}$ of simple polyhedral objects, we want to know, for a given point p and direction $\vec{d}$, the first object in $\mathcal{P}$ intersected by the ray defined by the pair $p, \vec{d}$. A single polyhedron with many faces can be represented without loss of generality by the collection of its faces, each treated as a separate polygon.

ON-LINE RAY SHOOTING IN AN ARBITRARY DIRECTION

Here we consider the on-line model in which the set $\mathcal{P}$ is given in advance and a data structure is produced and stored. Afterwards we are given the query rays one-by-one and the answer to one query must be produced before the next query is asked.

Table 32.3.1 summarizes the known complexity bounds on this problem. For a given class of objects we report the query time, the storage, and the preprocessing time of the method with the best bound. In this table and in the following ones we have omitted the big-O symbols. Again, n denotes the sum of the sizes of all the polyhedra in $\mathcal{P}$. The main references on ray shooting (Table 32.3.1) are in [Pel93b, dBH$^+$94] (boxes), [AM93, AM94, Pel93b, dBH$^+$94, AS93b] (polyhedra), [Pel96] (horizontal polygons), [AAS97, MS97] (spheres), and [DK85, AS96a] (convex polyhedra).

GLOSSARY

Fat horizontal polygons: Convex polygons contained in planes parallel to the xy-plane, with a lower bound on the size of their minimum interior angle.

Curtains: Polygons in 3-space bounded by one segment and by two vertical rays from the endpoints of the segment.

Axis-oriented curtains: Curtains hanging from a segment parallel to the x- or y-axis.

When we drop the fatness assumption for horizontal polygons we obtain bounds that depend on K, the complexity of the set of lines missing the *edges* of the polygons. K is in the range $[n^2, ..., n^4]$ and reaches the upper end of the range only when the polygons are very long and thin.

Most of the data structures for ray shooting mentioned in Table 32.3.1 can be made dynamic (under insertion and deletion of objects in the scene) by using general

TABLE 32.3.1 On-line ray shooting in an arbitrary direction.

OBJECTS	QUERY	STORAGE	PREPROCESSING
Boxes, terrains, curtains	$\log n$	$n^{2+\epsilon}$	$n^{2+\epsilon}$
Boxes	$n^{1+\epsilon}/m^{1/2}$	$n \le m \le n^2$	$m^{1+\epsilon}$
Polyhedra	$\log n$	$n^{4+\epsilon}$	$n^{4+\epsilon}$
Polyhedra	$n^{1+\epsilon}/m^{1/4}$	$n \le m \le n^4$	$m^{1+\epsilon}$
Fat horiz polygons	$\log n$	$n^{2+\epsilon}$	$n^{2+\epsilon}$
Horiz polygons	$\log^3 n$	$n^{3+\epsilon} + K$	$n^{3+\epsilon} + K \log n$
Spheres	$\log^4 n$	$n^{3+\epsilon}$	$n^{3+\epsilon}$
1 convex polyhedron	$\log n$	n	$n \log n$
s convex polyhedra	$\log^2 n$	$n^{2+\epsilon} s^2$	$n^{2+\epsilon} s^2$

dynamization techniques (see [Meh84]) and other more recent results [AEM92].

ON-LINE RAY SHOOTING IN A FIXED DIRECTION

We can usually improve on the general case if the direction of the rays is fixed a priori, while the source of the ray can lie anywhere in $\mathbb{R}^3$. See Table 32.3.2; here k is the number of vertices, edges, faces, and cells of the arrangement of the (possibly intersecting) polyhedra. References for ray shooting in a fixed direction (Table 32.3.2) are [dB93, dBGH94].

TABLE 32.3.2 On-line ray shooting in a fixed direction.

OBJECTS	QUERY TIME	STORAGE	PREPROCESSING
Boxes	$\log n$	$n^{1+\epsilon}$	$n^{1+\epsilon}$
Boxes	$\log n (\log \log n)^2$	$n \log n$	$n \log^2 n$
Axis-oriented curtains	$\log n$	$n \log n$	$n \log n$
Polyhedra	$\log^2 n$	$n^{2+\epsilon} + k$	$n^{2+\epsilon} + k \log n$
Polyhedra	$n^{1+\epsilon}/m^{1/3}$	$n \le m \le n^3$	$m^{1+\epsilon}$

OFF-LINE RAY SHOOTING IN AN ARBITRARY DIRECTION

In the previous section we considered the on-line situation when the answer to the query must be generated before the next question is asked. In many situations we do not need such strict requirements. For example, we might know all the queries from the start and are interested in minimizing the total time needed to answer all of the queries (the ***off-line*** situation). In this case there are simpler algorithms that improve on the storage bounds of on-line algorithms:

THEOREM 32.3.1

Given a polyhedral set $\mathcal{P}$ with n vertices, edges, and faces, and given m rays off-line*, we can answer the m ray-shooting queries in time $O(m^{0.8} n^{0.8+\epsilon} + m \log^2 n + \log n \log m)$ using $O(n+m)$ storage.*

One of the most interesting applications of this result is the current asymptotically fastest algorithm for detecting whether two nonconvex polyhedra in 3-space intersect, and to compute their intersection. See Table 32.3.3; here k is the size of the intersection.

TABLE 32.3.3 Detection and computation of intersection among polyhedra.

OBJECTS	DETECTION	COMPUTATION	SOURCES
Polyhedra	$n^{1.6+\epsilon}$	$n^{1.6+\epsilon} + k\log^2 n$	[Pel93b]
Terrains	$n^{4/3+\epsilon}$	$n^{4/3+\epsilon} + k^{1/3}n^{1+\epsilon} + k\log^2 n$	[CEGS94, Pel94b]

Lower bounds on off-line ray-shooting and intersection problems in 3D are difficult to prove. It has been shown in [Eri95] that many such problems are at least as hard as Hopcroft's incidence problem (in the appropriate ambient space).

RAY-SHOOTING IN SIMPLICIAL COMPLEXES

If we have a subdivision of the free space $R^3 \setminus \mathcal{P}$ into a simplicial complex we can answer ray shooting queries by locating the tetrahedron containing the source of the ray and tracing the ray in the complex at cost $O(1)$ for each visited face of the complex. There are scenes $\mathcal{P}$ for which any simplicial complex has some line meeting $\Omega(n)$ faces of the complex. The average time for tracing a ray in a simplicial complex is proportional to the sum of the areas of all faces in the complex. It is possible to find a complex of total surface within a constant multiplicative factor of the minimum, with $O(n^3 \log n)$ simplices in time $O(n^3 \log n)$ for general $\mathcal{P}$. For $\mathcal{P}$ a point-set or a single polyhedron $O(n^2 \log n)$ time suffices (see [AAS95, AF99, CD99]). These results are obtained via a generalization of Eppstein's method for two-dimensional Minimum Weighted Steiner Triangulation (2d-MWST) of a point set [Epp94]. In the 3D context the weight is the surface of the 2d faces of the complex. In a nutshell, starting from the set $\mathcal{P}$ of polyhedral obstacles in R^3, an oct-tree-based decomposition of R^3 is produced which is "balanced" and "smooth". It is then proved, via a charging argument, that the sum of the surfaces of all the boxes in the decomposition is within a constant factor of the surface of any Minimum Surface Steiner Simplicial Complex compatible with $\mathcal{P}$. From the oct-tree the final complex is derived within just a constant factor increase in the total surface.

EXTENSIONS AND ALTERNATIVE METHODS

Some ray-shooting results of Agarwal and Matoušek are obtained from the observation that a ray is traced by a family of segments $\rho(t)$, where one endpoint is the ray source and the second endpoint lies on the ray at distance t from the source. Using *parametric search* techniques (Chapter 37), Agarwal and Matoušek compute the first value of t for which $\rho(t)$ intersects $\mathcal{P}$, and thus answer the ray-shooting query.

An interesting extension of the concept of shooting rays against obstacles is obtained by shooting triangles and more generally simplices. We consider a family of simplices $s(t)$, indexed by real parameter $t \in \mathbb{R}^+$, where t is the volume of the simplex $s(t)$, such that the simplices form a chain of inclusions: $t_1 \leq t_2 \Rightarrow s(t_1) \subset s(t_2)$, Intuitively we grow a simplex until it first meets one of the obstacles. Surprisingly, when the obstacles are general polyhedra, shooting simplices is not harder than shooting rays.

THEOREM 32.3.2 [Pel94a]

Given a set of polyhedra $\mathcal{P}$ with n edges we can preprocess it in time $O(m^{1+\epsilon})$ into a data structure of size m, such that the following queries can be answered in time $O(n^{1+\epsilon}/m^{1/4})$: Given a simplex s, does s avoid $\mathcal{P}$? Given a family of simplices $s(t)$ as above, which is the first value of t for which $s(t)$ intersects $\mathcal{P}$?

Computing the interaction between beams and polyhedral objects is a central problem in radio-therapy and radio-surgery (see e.g. [SAL93] [For99] [CHX00]).

Other popular methods for solving ray-shooting problems are based binary space partitions, kD-trees, solid modeling schemes, etc. These methods, although important in practice, are usually not fully analyzable a priori using algorithmic analysis. In [ABCC02] Aronov et al. propose techniques that give *a posteriori* estimates of the cost or ray-shooting.

OPEN PROBLEMS

1. Find time and storage bounds for ray-shooting general polyhedra that are sensitive to the actual complexity of a group of lines (as opposed to the worst case bound on such a complexity).
2. For a collection of s convex polyhedra there is a wide gap in storage and preprocessing between the special case $s = 1$ and the case for general s. It would be interesting to obtain a bound that depends smoothly on s.
3. No lower bound on time or storage required for ray shooting is known.

32.4 MOVING LINES AMONG OBSTACLES

ARBITRARY MOTIONS

So far we have treated lines as static objects. In this section we consider moving lines. A laser beam in manufacturing or a radiation beam in radiation therapy can be modeled as lines in 3-space moving among obstacles. The main computational problem is to decide whether a source line l_1 can be moved continuously until it coincides with a target line l_2 so that it avoids a set of obstacles $\mathcal{P}$. We consider the following situation where the set of obstacles $\mathcal{P}$ is given in advance and preprocessed to obtain a data structure. When the query lines l_1 and l_2 are given the answer is produced before a new query is accepted. We have the results shown in Table 32.4.1, where K is the complexity of the set of lines missing the edges of the polygons (cf. Section 32.2).

TABLE 32.4.1 On-line collision-free movement of lines among obstacles.

OBJECTS	QUERY TIME	STORAGE	PREPROC	SOURCES
Polyhedra	$\log n$	$n^{4+\epsilon}$	$n^{4+\epsilon}$	[Pel93b]
Horiz polygons	$\log^3 n$	$n^{3+\epsilon} + K$	$n^{3+\epsilon} + K \log n$	[Pel96]

OPEN PROBLEMS

It is not known how to trade off storage and query time, or whether better bounds can be obtained in an off-line situation.

TRANSLATIONS

We now restrict the type of motion and consider only translations of lines. The first result is negative: there are sets of lines which cannot be split by any collision-free translation. There exists a set L of 9 lines such that, for all directions v and all subsets $L_1 \subset L$, L_1 cannot be translated continuously in direction v without collisions with $L \setminus L_1$. [SS93]. Positive results are displayed in Table 32.4.2.

GLOSSARY

Towering property: Two sets of lines L_1 and L_2 are said to satisfy the towering property if we can translate simultaneously all lines in L_1 in the vertical direction without any collision with any lines in L_2.

Separation property: Two sets of lines satisfy the separation property if they satisfy the towering property in some direction (not necessarily vertical).

TABLE 32.4.2 Separating lines by translations.

PROPERTY	TIME TO CHECK PROPERTY	SOURCES
Towering	$O(n^{4/3+\epsilon})$	[CEGS96]
Separation	$O(n^{3/2+\epsilon})$	[Pel94b]

32.5 CLOSEST PAIR OF LINES

GLOSSARY

Distance between lines: The Euclidean distance between two lines l_1 and l_2

in 3-space is the length of the shortest segment with one endpoint on l_1 and the other on l_2.

Vertical distance between lines: The length of the vertical segment with one endpoint on l_1 and one endpoint of l_2 (provided a unique such segment exists).

Vertical distance between segments: The length of the vertical segment with one endpoint in s_1 and one in s_2. If a unique such vertical segment does not exist the vertical distance is undefined.

TABLE 32.5.1 Closest and farthest pair of lines and segments.

PROBLEM	OBJECTS	TIME	SOURCES
Smallest distance	lines	$O(n^{8/5+\epsilon})$	[CEGS93]
Smallest vertical distance	lines, segments	$O(n^{8/5+\epsilon})$	[Pel94a]
Largest vertical distance	lines, segments	$O(n^{4/3+\epsilon})$	[Pel94a]

Any centrally symmetric convex polyhedron C in 3D defines a metric L_C. If C has constant combinatorial complexity, then the complexity of the Voronoi diagram of n lines in 3-space is $O(n^2\alpha(n)\log n)$ [CKS+98]. For Euclidean distance the best bound is $O(n^{3+\epsilon})$.

OPEN PROBLEM

1. Finding an algorithm with subquadratic time complexity for the smallest distance among segments (and more generally, among polyhedra) is a notable open question.
2. Close the gap between the complexity of Voronoi diagrams of lines induced by polyhedral metrics and the Euclidean metric.

32.6 DOMINANCE RELATION AND WEAVINGS

GLOSSARY

Dominance relation: Given a finite set L of nonvertical disjoint lines in $\mathbb{R}^3$, define a dominance relation $\prec$ among lines in L as follows: $l_1 \prec l_2$ if l_2 lies above l_1, i.e., if, on the vertical line intersecting l_1 and l_2, the intersection with l_1 has smaller z-coordinate than the intersection with l_2.

Weaving: A weaving is a pair $(L', \prec')$ where L' is a set of lines on the plane and $\prec'$ is an anti-symmetric nonreflexive binary relation $\prec' \subset L' \times L'$ among the lines in L'.

Realizable: A weaving is realizable if there exists a set of lines L in 3-space such that L' is the projection of L and $\prec'$ is the image of the dominance relation $\prec$ for L.

Elementary cycle: A cycle in the dominance relation such that the projections of the lines in such a cycle bound a cell of the arrangement of projected lines.

Perfect: A weaving $(L', \prec')$ is perfect if each line l alternates below and above the other lines in the order they cross l (see Figure 32.6.1a).

Bipartite weaving: Two families of segments in 3-space such that, when projecting on the xy-plane, each segment does not meet segments from its own family and meets all the segments from the other family. (A bipartite weaving of size 4×4 is shown in Figure 32.6.1b.)

Perfect bipartite weaving: Every segment alternates above and below the segments of the other family (see Figure 32.6.1b).

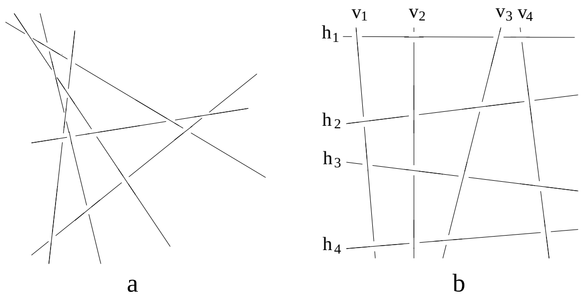

FIGURE 32.6.1
(a) *A perfect weaving;*
(b) *a perfect bipartite weaving.*

The dominance relation is possibly cyclic, that is, there may be three lines such that $l_1 \prec l_2 \prec l_3 \prec l_1$. Some results in [CEG+92, PPW93, dBOS94, Sol98] related to dominance are the following:

1. *How fast can we generate a consistent linear extension if the relation $\prec$ is acyclic?* $O(n^{4/3+\epsilon})$ time is sufficient for the case of lines. This result has been extended to the case of segments and polyhedra. If an ordering is given as input, it is possible to *verify* that it is a linear extension of $\prec$ in time $O(n^{4/3+\epsilon})$.

2. *How many elementary cycles in the dominance relation can n lines define?* In the case of bipartite weavings, the dominance relation can have $O(n^{3/2})$ elementary cycles and there is a family of bipartite weavings attaining the lower bound $\Omega(n^{4/3})$. For general weavings there is a construction attaining $\Omega(n^{3/2})$.

3. *If we cut the segments to eliminate cycles, how many "cuts" are necessary to eliminate all cycles?* From the previous result we have that sometimes $\Omega(n^{4/3})$ cuts are necessary since a single cut can eliminate only one elementary cycle. In order to eliminates all cycles (including the nonelementary ones) in a bipartite weaving, $O(n^{9/5})$ cuts are always sufficient.

4. *How fast can we find those cuts?* There is an algorithm to find cuts in bipartite weavings in time $O(n^{9/5} \log n)$. In a general weaving, calling μ is the minimum number of cuts, there is an algorithm to cut all cycles in time $O(n^{4/3+\epsilon}\mu^{1/3})$ that produces $O(n^{1+\epsilon}\mu^{1/3})$ cuts.

5. The fraction of realizable weavings over all possible weavings of n lines tends to 0 exponentially as n tends to ∞.
6. A perfect weaving of $n \geq 4$ lines is not realizable.
7. Perfect bipartite weavings are realizable only if one of the families has fewer than four segments.

32.7 SOURCES AND RELATED MATERIAL

FURTHER READING

Books and Surveys.

[Som51, HP52, Jes03]: Extensive book-length treatments of the geometry of lines in space.

[Sto89, Sto91]: Algorithmic aspects of computing in projective spaces.

[BR79, Shi78]: Uses of the geometry of lines in robotics. For uses in graphics see [FVFH90].

[dB93]: A detailed description of many ray-shooting results.

[Spe92, Dur99, Hav01]: Pointers to the vast related literature on pragmatic aspects of ray shooting.

RELATED CHAPTERS

Chapter 21: Arrangements
Chapter 31: Point location
Chapter 32: Range searching
Chapter 33: Geometric intersection
Chapter 37: Parametric search
Chapter 40: Algorithmic motion planning
Chapter 42: Computer graphics

REFERENCES

[AAS95] P. K. Agarwal, B. Aronov and S. Suri. Stabbing triangulations by lines in 3D. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 267–276, 1995.

[AAS97] P.K. Agarwal, B. Aronov, and M. Sharir. Computing envelopes in four dimensions with applications. *SIAM J. Comput.*, 26:1714–1732, 1997.

[AEM92] P.K. Agarwal, D. Eppstein, and J. Matoušek. Dynamic half-space range searching, geometric optimization and minimum spanning trees. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 80–89, 1992.

[Aga94] P.K. Agarwal. On stabbing lines for convex polyhedra in 3D. *Comp. Geom. Theory Appl.*, 4:177–189, 1994.

[AM93] P.K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22:794–806, 1993.

[AM94] P.K. Agarwal and J. Matoušek. Range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.

[Ame92] N. Amenta. Finding a line transversal of axial objects in three dimensions. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 66–71, 1992.

[AF99] B. Aronov and S. Fortune. Approximating minimum-weight triangulations in three dimensions. *Discrete Comput. Geom.*, 21:527–549, 1999.

[APS93] B. Aronov, M. Pellegrini, and M. Sharir. On the zone of an algebraic surface in a hyperplane arrangement. *Discrete Comput. Geom.*, 9:177–188, 1993.

[ABCC02] B. Aronov, H. Brnnimann, A.Y. Chang, and Y.-J. Chiang. Cost prediction for ray shooting. In *Proc. of Eighteenth ACM Symp. on Geom. Comput.* pages 293-302, 2002.

[AS96a] P.K. Agarwal and M. Sharir. Ray shooting amidst convex polyhedra and polyhedral terrains in three dimensions. *SIAM J. Comput.*, 25:100–116, 1996.

[AS93b] P.K. Agarwal and M. Sharir. Applications of a new space partitioning technique. *Discrete Comput. Geom.*, 9:11–38, 1993.

[AS96] P.K. Agarwal and M. Sharir. Efficient randomized algorithms for some geometric optimization problems. *Discrete Comput. Geom.*, 16:317–337, 1996.

[BR79] O. Bottima and B. Roth. *Theoretical Kinematics*. North-Holland, Amsterdam, 1979.

[CEG+92] B. Chazelle, H. Edelsbrunner, L.J. Guibas, R. Pollack, R. Seidel, M. Sharir, and J. Snoeyink. Counting and cutting cycles of lines and rods in space. *Comput. Geom. Theory Appl.*, 1:305–323, 1992.

[CEGS96] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Lines in space: combinatorics and algorithms. *Algorithmica*, 15:428–447, 1996.

[CEGS93] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Diameter, width, closest line pair and parametric search. *Discrete Comput. Geom.*, 10:183–196, 1993.

[CEGS94] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Algorithms for bichromatic line segment problems and polyhedral terrains. *Algorithmica*, 11:116–132, 1994.

[CHX00] Danny Z. Chen, Xiaobo Hu and Jinhui Xu Optimal Beam Penetrations in Two and Three Dimensions. Proceedings ISAAC 2000 *Lecture Notes in Computer Science* vol. 1969, 491-502, 2000.

[CD99] S.W. Cheng and T.K. Dey. Approximate minimum weight Steiner triangulation in three dimensions. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 205–214, 1999.

[CKS+98] L.P. Chew, K. Kedem, M. Sharir, B. Tagansky, and E. Welzl. "Voronoi diagrams of lines in 3-space under polyhedral convex distance functions. *J. of Algorithms* 29:238–255, 1998.

[dB93] M. de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal*, volume 703 of *Lecture Notes in Comput. Sci.* Springer-Verlag, New York, 1993.

[dBGH94] M. de Berg, L. Guibas, and D. Halperin. Vertical decompositions for triangles in 3-space. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 1–10, 1994.

[dBH+94] M. de Berg, D. Halperin, M. Overmars, J. Snoeyink, and M. van Kreveld. Efficient ray-shooting and hidden surface removal. *Algorithmica*, 12:31–53, 1994.

[dBOS94] M. de Berg, M. Overmars, and O. Schwarzkopf. Computing and verifying depth orders. *SIAM J. Comput.*, 23:437–446, 1994.

[DK85] D.P. Dobkin and D.G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *J. Algorithms*, 6:381–392, 1985.

[DDP97] F. Durand, G. Drettakis, and C. Puech. The visibility skeleton: a powerful and efficient multi-purpose global visibility tool. *Computer Graphics* 31:89–100, 1997.

[Dur99] F. Durand. *3D Visibility: Analytical Study and Applications.* PhD thesis, Universite Joseph Fourier, Grenoble, 1999.

[Epp94] D. Eppstein. Approximating the minimum weight Steiner triangulation. *Disc. Comp. Geom.* 11:163–191, 1994.

[EE99] D. Eppstein and J. Erickson. Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions. *Discrete Comput. Geom.* 22(4):569–592, 1999.

[Eri95] J. Erickson. The relative complexities of some geometric problems. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 85–90, 1995.

[For99] S. Fortune. Topological beam tracing. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 59–68, 1999.

[FVFH90] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, 1990.

[Hav01] V. Havran. *Heuristic Ray Shooting Algorithms.* PhD thesis, Czech Technical University, Praha , Czech Republic, 2001.

[HP52] W.V.D. Hodge and D. Pedoe. *Methods of Algebraic Geometry.* Cambridge University Press, 1952.

[HS94] D. Halperin and M. Sharir. New bounds for lower envelopes in three dimensions, with applications to visibility in terrains. *Discrete Comput. Geom.*, 12:313–326, 1994.

[Jes03] C.M. Jessop. *A Treatise on the Line Complex.* Cambridge University Press, 1903.

[Meg91] N. Megiddo. Personal communication. 1991.

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

[MS97] S. Mohaban and M. Sharir. Ray-shooting amidst spheres in three-dimensions and related problems. *SIAM J. Comput.* 26:654–674, 1997.

[Oro01] J. O'Rourke, Computational Geometry Column 41. *Int. J. Comp. Geom. Appl.*, Vol. 11, No. 2 (2001) 239-242.

[Pel91] M. Pellegrini. *Combinatorial and Algorithmic Analysis of Stabbing and Visibility Problems in 3-Dimensional Space.* PhD thesis, Courant Institute, New York Univ., 1991. Robotics Lab Tech. Rep. 241.

[Pel93a] M. Pellegrini. Lower bounds on stabbing lines in 3-space. *Comput. Geom. Theory Appl.*, 3:53–58, 1993.

[Pel93b] M. Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9:471–494, 1993.

[Pel94a] M. Pellegrini. On collision-free placements of simplices and the closest pair of lines in 3-space. *SIAM J. Comput.*, 23:133–153, 1994.

[Pel94b] M. Pellegrini. On lines missing polyhedral sets in 3-space. *Discrete Comput. Geom.*, 12:203–221, 1994.

[Pel96] M. Pellegrini. On point location and motion planning in arrangements of simplices. *SIAM J. Comput.*, 25:1061–1081, 1996.

[Plu65] J. Plücker. On a new geometry of space. *Philos. Trans. Royal Soc. London*, 155:725–791, 1865.

[PPW93] J. Pach, R. Pollack, and E. Welzl. Weaving patterns of lines and line segments in space. *Algorithmica*, 9:561–571, 1993.

[PS92] M. Pellegrini and P. Shor. Finding stabbing lines in 3-space. *Discrete Comput. Geom.*, 8:191–208, 1992.

[SAL93] A. Schweikard, J.R. Adler, and J.C. Latombe. *Motion Planning in Stereotaxic Radiosurgery. IEEE Transactions on Robotics and Automation*, 9(6):764-774, 1993.

[Shi78] B.E. Shimano. *The Kinematic Design and Force Control of Computer Controlled Manipulators.* PhD thesis, Dept. of Mechanical Engineering, Stanford Univ., 1978.

[Sol98] A. Solan. Cutting Cycles of Rods in Space. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 135–142, 1998.

[Som51] D.M.H. Sommerville. *Analytical Geometry of Three Dimensions.* Cambridge University Press, 1951.

[Spe92] R. Speer. An updated cross-indexed guide to the ray-tracing literature. *Comput. Graphics*, 26:41–72, 1992.

[SS93] J. Snoeyink and J. Stolfi. Objects that cannot be taken apart with two hands. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 246–256, 1993.

[Sto89] J. Stolfi. Primitives for computational geometry. Tech. Rep. 36, Digital Systems Research Center, Palo Alto, 1989.

[Sto91] J. Stolfi. *Oriented Projective Geometry: a Framework for Geometric Computations.* Academic Press, San Diego, 1991.

[ZS01] Y. Zhou and S. Suri. *Shape Sensitive Geometric Permutations.* In *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, pages 234–243, 2001.