Ray Shooting and Lines in Space

Marco Pellegrini

INTRODUCTION

Analytic geometry of lines in 3-space is a part of the body of classical analytic 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 vary large number of rays with the objects comprising the scene. In Robotics the chief problem is that of moving 3-dimensional objects without collisions. The effects due to the edges of objects have been studied with reference to 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 in 3-space and ray-shooting from the point of view of Computational Geometry. The emphasis is on provable non-trivial 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 the coordinates of lines (Section). This is an essential initial step since different coordinates highlight different properties of the lines in their interaction with other geometric objects. Here a special role is played by the *Plücker coordinates* (see Section), which represent the starting point for many of the most recent results. Then we consider how lines interact with each other (Section 34.1). 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, which 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 which are more commonly found in applications. We arrive in Section 34.2 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 mostly polyhedral objects, but in one case we are able to report a ray-shooting result on curved objects (spheres).

Section 34.3 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 34.4 we define a few notions of distance among lines, and as a consequence we have several 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 34.5 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. 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?

34.1 Coordinates of Lines

GLOSSARY

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

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

unoriented line: a line for which orientation is not distinguished

(I) Canonical coordinates by pairs of planes. The intersection of two planes of equation $y = az + b$ and $x = cz + d$ is a non-horizontal 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 other words, the space of non-horizontal lines is homeomorphic to R^4. Results on ray-shooting on 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 non-horizontal line l with the two planes define uniquely 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 parameterization have proved useful only in special cases. Many interesting results have been derived using instead a five-dimensional parameterization 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:

$$l = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \end{pmatrix}, \quad \text{with } a_0, b_0 > 0. \tag{34.1}$$

By taking the determinants of the six two-by-two submatrices of the above two-by-four matrix we obtain the *homogeneous Plücker coordinates* of the line:

$$p(l) = (\xi_{01}, \xi_{02}, \xi_{03}, \xi_{12}, \xi_{31}, \xi_{23}), \quad \text{with} \quad \xi_{ij} = det \begin{pmatrix} a_i & a_j \\ b_i & b_j \end{pmatrix}. \qquad (34.2)$$

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 P^5.

We also define the *Plücker hyperplane* of the line l to be the hyperplane in 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 :

$$h(l) = \{p \in P^5 | v(l) \cdot p = 0\} \qquad (34.3)$$

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

$$\xi_{01}\xi_{23} + \xi_{02}\xi_{31} + \xi_{03}\xi_{12} = 0 \qquad (34.4)$$

The set of points in P^5 satisfying (34.4) form the so called *Plücker hypersurface* Π, also called Klein quadric or Grassmanian manifold. The converse is also true; that is, every tuple of six real numbers satisfying (34.4) 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 four-by-four 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 1 *Lines l and l' intersect or are parallel (i.e. meet at infinity) if and only if $p(l) \in h(l')$.*

Note that condition 34.4 states in terms of Plücker coordinates the fact that any line always meets itself.

Lemma 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 [Arrangements]). The main drawback is that we are using five (non-homogeneous) 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 1 *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 1 is especially useful for deriving ray-shooting results.

This list of coordinatizations is by no means exhaustive. Other parameterizations are used for example in [Ame92, AAS94, Sha94].

34.0.1 A typical example

A typical example of how Plücker coordinates are used in 3-D problems is the result for fast ray shooting among polyhedra (see Table 5). We triangulate the faces of the polyhedra and we extend each edge to a full line. Each such line is mapped into a Plücker hyperplane. Lemma 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, then we 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 1 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 Chapters [Arrangements] and [Point Location]).

34.1 Sets of lines in 3-space

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

34.1.1 Groups of lines induced by a finite set of lines

Glossary:

- **semi-algebraic 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 R^d.
- **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 set 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 consistently-oriented with a set of 3-dimensional oriented lines L if the projection l' of l 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 R^4 is partitioned in 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 are going to consider are represented in 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 $Miss(L)$ contains lines that do not meet any line in L. The sets $Vert(L)$ and $Free(L)$ contain lines that may be translated to infinity without collision with lines in L.

TABLE 34.1 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 consistently oriented	$\Theta(n^2)$

34.1.2 Membership tests

When we are given L we can build a data structure during a pre-processing 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 2. Membership tests for groups of lines defined by lines.

Set of Lines	**Query time**	**Preproc./Storage**
$Miss(L)$	$O(\log n)$	$O(n^{4+\epsilon})$
1 component of $Miss(L)$, $Vert(L)$, $VertCO(L)$	$O(\log n)$	$O(n^{2+\epsilon})$
$Free(L)$	$O(\log n)$	$O(n^{3+\epsilon})$

34.1.3 Groups of lines induced by polyhedra

Glossary:

- **epsilon:** we denote by ϵ a positive real number, which, for each algorithm or data structure, we can choose arbitrarily close to zero. The caveat is that the multiplicative constant implicit in the big-Oh notation depends on ϵ and its value increases when ϵ tends to zero.
- $\alpha(\cdot)$: The function $\alpha(n)$ is the inverse of Ackerman's function. $\alpha(n)$ grows very slowly and is at most 4 for any practical value of n.
- $\beta(\cdot)$: $\beta(n) = 2^{c\sqrt{\log n}}$ 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**: A polyhedral set P is a region of 3-space bounded by a collection of interior-disjoint vertices, segments and planar polygons. We denote with n the number of vertices, edges and faces.
- **starshaped polyhedron:** a polyhedron P is star-shaped if there exist 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 single polyhedron or 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 will consider some interesting groups of lines induced by polyhedra and we will determine their complexity.

Table 3. Complexity of groups of lines defined by Polyhedra.

Set of Lines	**Definition**	**Complexity**
$Miss(P)$	do not meet polyhedron P	$\Theta(n^4)$
$Vert(P)$	can be translated vertically to ∞	$\Omega(n^3)$, $O(n^3\beta(n))$
$Free(P)$	can be translated to ∞ in some direction	$\Omega(n^3)$
$Miss(Q), Free(Q)$	Q star-shaped polyhedron or a terrain	$\Omega(n^2\alpha(n))$, $O(n^3 \log n)$

34.1.4 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.

34.1.5 Sets of stabbing lines

Glossary:

- **stabber**: A line l is a *stabber* for $\mathcal{P} = \{P_1, ..., P_k\}$ of polyhedral sets if l intersects every set P_i for $i = 1, .., k$. The sum of the sizes of the polyhedral sets in $\mathcal{P}$ is n.
- **box**: Polyhedra whose faces are orthogonal to one of the three Cartesian axis.
- **c-oriented**: polyhedra whose face normals come from a set of c fixed directions.

In the following table we have listed the worst case complexity of the set $S(\mathcal{P})$ of the lines stabbing $\mathcal{P}$ and the time to find a witness stabbing line.

Table 4. Complexity of the set of stabbing lines and detection time.

Objects	**Complexity of $S(P)$**	**Find Time**
Convex Polyhedra	$\Omega(n^3)$, $O(n^3 \log n)$	$O(n^3\beta(n))$
Boxes	$O(n^2)$	$O(n)$
c-oriented polyhedra	$O(n^2)$	$O(n^2)$
Horiz. Polygons	$\Theta(n^2)$	$O(n)$

Note that in some case (boxes, parallel polygons) we can find a stabbing line in linear time, although the best known bound on the complexity of the stabbing set is quadratic. These results are obtained using linear programming techniques (see Chapter [Linear Programming]).

If we are given a line l we can determine whether l is a stabber for a pre-processed set $\mathcal{P}$ of convex polyhedra in time $O(\log n)$ using data structures of size $O(n^{2+\epsilon})$ that can be obtained in time $O(n^{2+\epsilon})$.

34.1.6 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.

34.2 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, detecting and computing intersections of polyhedra). The problem is defined as follows: given a large collection P of simple polyhedral objects, we want to know, for a given point p and direction $\vec{d}$, the first object in 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 considered as a separate polygon.

34.2.1 On-line ray shooting in arbitrary direction

Here we consider the on-line model in which the set 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.

The next table 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-Oh symbols. Again, n denotes the sum of sizes of all the polyhedra in P.

Glossary:

- **fat horizontal polygons**: convex polygons contained in planes parallel to the xy-plane with a lower bound on the size of the minimum interior angle.
- **curtains**: polygons bounded by one rod in 3-space and by two vertical rays from the end-points of the rod.
- **axis-oriented curtains**: curtains hanging from a segment parallel to the x- or y-axis.

Table 5. On-line ray-shooting in arbitrary direction.

Objects	**Query**	**Storage**	**Preproc.**
Boxes, Terrains, Curtains	$\log n$	$n^{2+\epsilon}$	$n^{2+\epsilon}$
Boxes	$n^{1+\epsilon}/m^{1/2}$	$n \leq m \leq n^2$	$m^{1+\epsilon}$
Polyhedra	$\log n$	$n^{4+\epsilon}$	$n^{4+\epsilon}$
Polyhedra	$n^{1+\epsilon}/m^{1/4}$	$n \leq m \leq n^4$	$m^{1+\epsilon}$
Fat Horiz. Polyg.	$\log n$	$n^{2+\epsilon}$	$n^{2+\epsilon}$
Horiz. Polyg.	$\log^3 n$	$n^{3+\epsilon} + K$	$n^{3+\epsilon} + K \log n$
Spheres	$\log^4 n$	$n^{3+\epsilon}$	$n^{3+\epsilon}$
1 Conv. Polyhedr.	$\log n$	n	$n \log n$
s Conv. Polyhedr.	$\log^2$ n	$n^{2+\epsilon}s^2$	$n^{2+\epsilon}s^2$

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 the table can be made dynamic (under insertion and deletion of objects in the scene) by using general dynamization techniques (see [Meh84]) and other more recent results [AEM92].

34.2.2 On-line ray-shooting in 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 R^3.

Table 6. On-line ray-shooting in 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	$n^{1+\epsilon}/m^{1/3}$	$n \leq m \leq n^3$	$m^{1+\epsilon}$

In the table above k is the number of vertices, edges, faces and cells of the arrangement of the (possibly intersecting) polyhedra.

34.2.3 Off-line ray shooting in arbitrary direction

In the previous section we considered the *on-line* situation when we can preprocess the polyhedral set but the rays are given afterwards and the answer to the query must be received before the next question is asked. In many situations we do not need such strict requirements. For example we might know all of the queries from the start and we are interested in minimizing the total time needed to answer all of the queries (*off-line* situation). In this case we have simpler algorithms and we can improve on the storage bounds over the on-line algorithms:

Theorem 2 *Given a polyhedral set P with n vertices, edges and faces, and m rays* off-line, *we can answer 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 non-convex polyhedra in 3-space intersect, and to compute the intersection.

Table 7. Detection and Computation of Intersection among Polyhedra.

Objects	Detection	Computation
Polyhedra	$n^{1.6+\epsilon}$	$n^{1.6+\epsilon}+klog^2n$
Terrains	$n^{4/3+\epsilon}$	$n^{4/3+\epsilon}+k^{1/3}n^{1+\epsilon}+K\log^2 n$

In this table, similarly to Table 6, k is the size of the intersection.

34.2.4 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 end-point is the source of the ray and the second end-point is on the ray at distance t form the source. Using powerful *parametric search techniques* (see Chapter [Parametric Search]) Agarwal and Matoušek compute the first value of t for which $\rho(t)$ intersects 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 $t \in R^+$ that form a chain of inclusions (i.e. that $t_1 \leq t_2 \Rightarrow s(t_1) \subset s(t_2)$), where t is the volume of the simplex $s(t)$. Intuitively we are growing a simplex until it meets one of the obstacles. Surprisingly, when the obstacles are general polyhedra shooting simplices is not harder than shooting rays!

Theorem 3 *Given a set of polyhedra 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 P? Given a family of simplices $s(t)$ as above, which is the first value of t for which $s(t)$ intersects P?*

Other popular methods for solving ray-shooting problems are based on triangulations of 3-dimensional space, binary space partitions, solid modelling schemes etc. The performances of these methods are usually not fully analyzable using algorithmic analysis.

34.2.5 Open Problems

1. Time/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 general case for any value of s. It would be interesting to obtain a smooth bound in s.
3. No lower bound on time/storage required for ray-shooting is known.

34.3 Moving lines among obstacles

34.3.1 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 modelled 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 in such way to avoid a set of obstacles P. We consider the following situation where the set of obstacles 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 following results:

Table 8. On-line Collision-free Movement of Lines among Obstacles.

Objects	**Query Time**	**Storage**	**PreProc.**
Polyhedra	$\log n$	$n^{4+\epsilon}$	$n^{4+\epsilon}$
Horiz. Polygons	$\log^3 n$	$n^{3+\epsilon} + K$	$n^{3+\epsilon} + K \log n$

As in Table 5, K is the complexity of the set of lines missing the edges of the polygons (see Section 34.1.3).

34.3.2 Open problems

At the moment it is not known how to trade-off storage and query time or whether in an off-line situation we can obtain better bounds.

34.3.3 Translations

We now restrict the type of motion and we consider only translations of lines. The first result is of a negative nature since it states that 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 $L \setminus L_1$.

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 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 9. Separating lines by translations.

Property	**Time to check property**
Towering	$O(n^{4/3+\epsilon})$
Separation	$O(n^{3/2+\epsilon})$

34.4 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 end-point on l_1 and the other on l_2.
- **vertical distance between lines**: the length of the vertical segments with one end-point on l_1 and one end-point of l_2.
- **vertical distance between segments**: the length of the vertical segment with one end-point in s_1 and one in s_2. If such vertical segment does not exist the vertical distance is undefined.

Table 10. Closest and Farthest Pair of Lines and Segments.

Problem	**Objects**	**Time**
Smallest distance	Lines	$O(n^{8/5+\epsilon})$
Smallest vertical distance	Lines, Segments	$O(n^{8/5+\epsilon})$
Largest vertical distance	Lines, Segments	$O(n^{4/3+\epsilon})$

34.4.1 Open Problems

Finding a non trivial algorithm for the smallest distance among segments and more generally polyhedra is a notable open question.

34.5 Dominance relation and weavings

Glossary:

- **dominance relation**: Given a finite set L of non-vertical disjoint lines in R^3, we define a *dominance* relation $\prec$ among lines in L. We write $l_1 \prec l_2$ if l_2 lies above l_1, or, formally, 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 non-reflexive binary relation $\prec' \subset L' \times L'$ among the lines in L'.
- **realizable:** A weaving is *realizable* if there exists a set of spatial lines L such that L' is the projection of L and $\prec'$ is homeomorphic to 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 alternates below and above the other lines in the order they cross l (see Figure 1.a).
- **bipartite weaving**: two families of segments in 3-space such that, projecting them on the xy-plane, every segment does not meets segment from its own family and it meets all the segments from the other family. A bipartite weaving of size 4-by-4 is shown in figure 1(a).
- **perfect bipartite weaving**: every segment alternates above and below the segments of the other family (see Figure 1(b)).

The dominance relation is possibly cyclic, that is, we can find three lines such that $l_1 \prec l_2 \prec l_3 \prec l_1$. Some results 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 $\Omega(n^{4/3})$.
(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 (also non elementary) in a bipartite weaving $O(n^{9/5})$ cuts are always sufficient
(4) (4) The fraction of realizable weavings over all possible weavings of n lines tends to 0 exponentially as n tends to ∞.
(5) A perfect weaving of $n \geq 4$ lines is not realizable.
(6) Perfect bipartite weaving exist only if one of the families has less than four segments.

34.6 Sources and related material

Books and Surveys. An extensive treatment of analytic geometry of lines in space can be found in books by Sommerville [Som51], Hodge and Pedoe [HP52] and Jessop [Jes69]. Algorithmic aspects of computing in projective spaces are discussed in [Sto89, Sto91].
Uses of the geometry of lines in robotics are reported in [BR79, Shi78]. For uses in Graphics see [FVFH90].
A detailed description of many ray-shooting results is in [dB93].
A guide to the vast related literature on pragmatic aspects of ray-shooting can be found in [Spe92].
Key papers. The original description of Plücker coordinates is in [Plu65]. Theorem 1 is in [APS93].
Groups of lines induced by lines (Table 1) are discussed in [CEGS89, Pel94b]. Membership tests (Table 2) are in [CEGS89, Pel93b, Pel94b]. The bounds on sets of lines induced by polyhedra in Table 3 are in [HS94] [Pel94b, Aga94]. A key paper in stabbing is [PS92]. Other results on stabbing are in [Pel93a, Aga94, Ame92, Meg91, Pel91].
The main references on ray-shooting (Table 5) are in [Pel93b, dBHO$^+$94] (boxes), [AM93, AM94, Pel93b, dBHO$^+$94, AS93b] (polyhedra), [Pel94c] (horizontal polygons), [AAS94, MS94] (spheres), and [DK85, AS93a] (convex polyhedra). References for ray shooting in fixed direction (Table 6) are [dB93, dBGH94].
Results on off-line ray shooting and computing intersection of polyhedra (Table 7) are in [CEGS94, Pel94b, Pel93b]. Theorem 3 is in [Pel94a]. Results on moving and translating lines in 3-space (Tables 8 and 9) are in [SS93], [Pel93b, Pel94c] and [CEGS89, Pel94b]. Neighbour problems for lines (Table 10) are discussed in [CEGS93, Pel94a].
Weavings are discussed in [CEG$^+$92] and [PPW93]. Results on linear extensions of the dominance relation are in [dBOS94].
Related Chapters
Chapter X: Arrangements
Chapter X: Visibility
Chapter X: Point Location
Chapter X: Range Searching
Chapter X: Geometric Intersection
Chapter X: Computer Graphics
Chapter X: Motion Planning
Chapter X: Parametric Search
Chapter X: Randomized algorithms

References

[AAS94] P.K. Agarwal, B. Aronov, and M. Sharir. Computing envelopes in four dimensions with applications. In *Proceedings of the 10th ACM Symposium on Computational Geometry*, pages 348–358, 1994.

[AEM92] P.K. Agarwal, D. Eppstein, and J. Matoušek. Dynamic half-space range searching, geometric optimization and minimum spanning trees. In *Proceeding of the 33rd Symposium on Foundations of Computer Science*, pages 80–89, 1992. To appear in Algorithmica.

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

[AM93] P.K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. on Computing*, 22(4):794–806, 1993.

[AM94] P.K. Agarwal and J. Matoušek. Range searching with semialgebraic sets. *Discrete & Computational Geometry*, 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.

[APS93] B. Aronov, M. Pellegrini, and M. Sharir. On the zone of an algebraic surface in a hyperplane arrangement. *Discrete & Computational Geometry*, 9(2):177–188, 1993.

[AS93a] P. K. Agarwal and M. Sharir. Ray shooting amidst convex polytopes in three dimensions. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 260–270, 1993.

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

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

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

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

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

[dB93] M. de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal*. Number 703 in Lecture Notes in Computer Science. Springer Verlag, 1993.

[dBGH94] M. de Berg, L. Guibas, and D. Halperin. Vertical decompositions for triangles in 3-space. In *Proceedings of the 10h ACM Symposium on Computational Geometry*, pages 1–10, 1994.

[dBHO+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. on Computing*, 23(2):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.

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

[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 & Computational Geometry*, 12:313–326, 1994.

[Jes69] C.M. Jessop. *A Treatise on the Line Complex.* Chelsea, 1969.

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

[Meh84] K. Mehlhorn. *Multidimensional Searching and Computational Geometry*, chapter 2. Springer Verlag, 1984.

[MS94] S. Mohaban and M. Sharir. Ray-shooting amidst spheres in three-dimensions and related problems. *SIAM J. Computing*, 1994. To appear.

[Pel91] M. Pellegrini. *Combinatorial and Algorithmic Analysis of Stabbing and Visibility Problems in 3-Dimensional Space.* PhD thesis, New York University–Courant Institute of Mathematical Sciences, February 1991. Courant Institute, Robotics Lab. Report Number 241.

[Pel93a] M. Pellegrini. Lower bounds on stabbing lines in 3-space. *Computational geometry theory and applications*, 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. on Computing*, 23(1):133–153, 1994.

[Pel94b] M. Pellegrini. On Lines Missing Polyhedral Sets in 3-Space. *Discrete & Computational Geometry*, 12:203–221, 1994.

[Pel94c] M. Pellegrini. On Point Location and Motion Planning in Arrangements of Simplices. In *Proceedings of the 26th ACM Symposium on Theory of Computing*, pages 95–104, 1994.

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

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

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

[Sha94] M. Sharir. Vertical decomposition of loer envelopes in four dimensions and its applications. Manuscript, September 1994.

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

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

[Spe92] R. Speer. An updated cross-indexed guide to the ray-tracing literature. *Computer Graphics*, 26(1):41–72, 1992.

[SS93] J. Snoeyink and J. Stolfi. Objects that cannot be taken apart with two hands. In *Proceedings of the 9th ACM Symposium on Computational Geometry*, pages 246–256, 1993.

[Sto89] J. Stolfi. Primitives for computational geometry. Technical Report 36, Digital Systems Research Center, 130 Lytton Avenue, Palo Alto, California, 1989.

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