Journal of Graphics Tools, 4(3):11-22, 1999

Determining the Lines Through Four Lines

Seth Teller*
Computer Graphics Group
MIT Lab for Computer Science

Michael Hohmeyer†
ICEM CFD Engineering
Berkeley, CA

Abstract

This paper describes how to compute the line or lines which intersect four given lines in three dimensions. This intersection computation arises in computer graphics (for visibility computations), computational geometry (for line traversals), and computer vision (for object recognition).

Given four distinct lines in three dimensions there exist zero, one, two, or various infinities of lines intersecting the given lines. We use the Plücker coordinatization of lines to cast this problem as a null-space computation in five dimensions, and show how the singular value decomposition (SVD) yields a simple, stable characterization of the incident lines, and an efficient algorithm to determine them.

Finally, we enumerate the types of input degeneracies that may arise, show how to detect each type in practice, and describe for each case the solution set of lines that arises.

CR Categories and Subject Descriptors: [Computer Graphics]: I.3.5 Computational Geometry and Object Modeling – *geometric algorithms, languages, and systems.*

Additional Key Words and Phrases: Singular value decomposition, Plücker coordinates, line incidence, line transversals.

*545 Technology Square, Cambridge MA 02139
†2855 Telegraph Avenue #501, Berkeley CA 94705

1 Introduction

The line is an important primitive element in the geometry of three-space. Computations involving lines arise in computer graphics, computational geometry, and computer vision. Consider the computer graphics problem of characterizing the regions of light and shadow in three dimensions produced by a polygonal light source illuminating a scene of polyhedral objects. The illumination function has discontinuous derivatives along surfaces arising from combinations of three edges of polyhedra [7, 11]. Characterizing the effects of these surfaces requires the determination of their intersections, which occur wherever a probe line – a light ray – intersects four edges from the polyhedral scene [15, 4]. Line intersections arise in computational geometry, for example in the determination of lines that simultaneously intersect a collection of geometric objects [1, 8, 2]. Finally, in computer vision, line intersections arise during construction of data structures for three-dimensional object recognition [10, 5].

Lines, however, do not behave as simply as points and planes. For example, three generic points determine a unique (homogeneous) plane by incidence, and three generic planes determine a unique (homogeneous) point; both computations involve solution of a system of equations linear in the input coordinates. In contrast, *four* generic lines determine *two* further lines by incidence (Figure 1). Moreover, as we shall show, computing line incidence is an inherently quadratic problem.

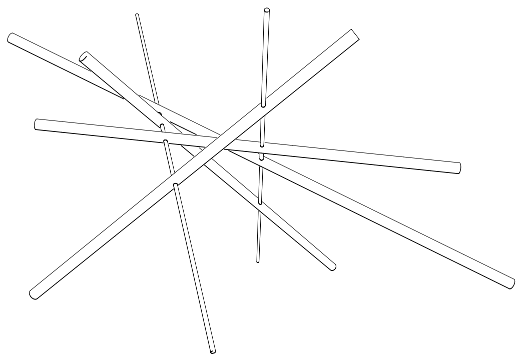

Figure 1: Four (generic) lines determine two further lines by incidence.

This paper presents an algorithm that, given four arbitrary lines, determines the line(s) or family of lines incident on the input lines. Four generic lines induce exactly two incident lines. In practice, however, input degeneracies may result in zero, one, two, or various infinities of incident lines. We characterize the degeneracies in each case, and describe the representation of the resulting line families. This paper is accompanied by C source code which performs the incidence computation for generic input lines, and detects and classifies degenerate input instances.

2 Plücker Coordinates

Plücker coordinates provide a convenient representation for directed lines in three space [13, 14, 12]. This section reviews relevant notation for, and some useful properties of, Plücker coordinates.

Each pair of distinct points $p = (p_x, p_y, p_z)$ and $q = (q_x, q_y, q_z)$ defines a directed line ℓ in $\mathbf{R}^3$. This line corresponds to a set of six coefficients called the Plücker coordinates Π_ℓ of the line:

$$\Pi_\ell = (\pi_{\ell 0}, \pi_{\ell 1}, \pi_{\ell 2}, \pi_{\ell 3}, \pi_{\ell 4}, \pi_{\ell 5}).$$

Each Plücker coordinate is the determinant of a 2×2 minor of the matrix

$$\begin{pmatrix} p_x & p_y & p_z & 1 \\ q_x & q_y & q_z & 1 \end{pmatrix}.$$

We adopt the following convention relating these minors and the π_i [9]:

$$\begin{array}{rcl}
\pi_{\ell 0} & = & p_x q_y - q_x p_y \\
\pi_{\ell 1} & = & p_x q_z - q_x p_z \\
\pi_{\ell 2} & = & p_x \quad - q_x \\
\pi_{\ell 3} & = & p_y q_z - q_y p_z \\
\pi_{\ell 4} & = & p_z \quad - q_z \\
\pi_{\ell 5} & = & q_y \quad - p_y
\end{array}$$

If a and b are two directed lines, and Π_a, Π_b their corresponding Plücker mappings, a relation *side*(a, b) can be defined as the permuted inner product

$$side(a, b) = \Pi_a \odot \Pi_b = (\pi_{a0}\pi_{b4} + \pi_{a1}\pi_{b5} + \pi_{a2}\pi_{b3} + \pi_{a3}\pi_{b2} + \pi_{a4}\pi_{b0} + \pi_{a5}\pi_{b1}), \tag{1}$$

which is zero whenever a and b intersect or are parallel, and non-zero otherwise.

There are two useful, geometrically dual, ways to view the Plücker coordinates of a line. The first is as a point: the six-tuple Π_l can be treated as a (homogeneous) point $\Pi_\ell = (\pi_{\ell 0} \ldots \pi_{\ell 5})$ in $\mathbf{P}^5$. Only points satisfying a quadratic relation (given below) correspond to real lines in $\mathbf{R}^3$. Alternatively, the Plücker coordinates (after permutation) describe the coefficients $(\pi_{\ell 4}, \pi_{\ell 5}, \pi_{\ell 3}, \pi_{\ell 2}, \pi_{\ell 0}, \pi_{\ell 1})$ of a five-dimensional hyperplane, each point of which is dual to a line incident on ℓ. The advantage of transforming lines to Plücker coordinates is that determining the relationship (incidence; relative orientation) between two lines in $\mathbf{R}^3$ reduces to computing the permuted inner product of a homogeneous point (the mapping of one line) with a hyperplane (the mapping of the other).

Plücker coordinates simplify computations on lines by mapping them to points and hyperplanes, which are familiar objects. However, although every directed line in $\mathbf{R}^3$ maps to a point in Plücker coordinates, not every six-tuple of Plücker coordinates corresponds to a *real line*. Only those points Π satisfying the quadratic relation

$$\Pi \odot \Pi = 0 \tag{2}$$

correspond to real lines in $\mathbf{R}^3$. The remaining points correspond to imaginary lines, i.e., those with complex coefficients (e.g., plane intercepts) in $\mathbf{R}^3$.

The Plücker coordinates of a real line are not independent. First they are determined only to within a scale factor [13]. That is, if points p, q and r lie on a line ℓ, $\Pi(p, q)$ and $\Pi(p, r)$ are identical "up to scale," that is, there is some constant $c \neq 0$ such that $c \cdot \Pi(p,q) = \Pi(p,r)$. Indeed, for any line ℓ the set of points

$$\Pi(\ell) := \{\, \Pi(a,b) \mid a,b \in \ell \,\}$$

is a line through the origin in $\mathbf{R}^6$. The set of all lines through the origin in $\mathbf{R}^6$ is called a "five-dimensional real projective space," and denoted $\mathbf{P}^5$. Thus Π defines a map from directed lines in three space to points in $\mathbf{P}^5$.

The Plücker coordinates, in order to represent a line with real coefficients in $\mathbf{R}^3$, must satisfy Equation 2. Corresponding to intuition, then, the six Plücker coordinates describe a four-parameter space: one could parameterize lines in $\mathbf{R}^3$ in terms of, for example, their intercepts on a pair of parallel reference planes. (However, with only one pair of planes, one would need six numbers – a homogeneous point on one plane, and an ordinary point and offset from the other – to capture all the lines in $\mathbf{R}^3$, including lines parallel to the planes. This is one of the difficult aspects of representing lines!)

The set of points in $\mathbf{P}^5$ satisfying Equation 2 is called the *Plücker quadric* [13]. One might visualize this set as a four-dimensional surface, embedded in $\mathbf{P}^5$, that is analogous to a quadric hyperboloid of one sheet in $\mathbf{R}^3$ (Figure 2).

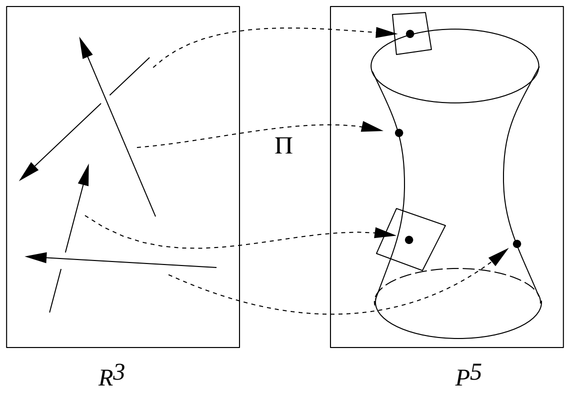

Figure 2: Real lines map to points on, or hyperplanes tangent to, the Plücker quadric.

Henceforth, we use $\mathbf{\Pi} : \ell \rightarrow \Pi_\ell$ to denote the map $\mathbf{\Pi}$ that takes a directed line ℓ to the Plücker point (hyperplane) Π_ℓ, and $\mathcal{L} : \Pi \rightarrow \ell_\Pi$ to denote the map that constructs, from a point Π on the Plücker quadric, the corresponding real directed line ℓ_Π in $\mathbf{R}^3$.

3 Determining the Incident Lines

This section describes the intersection computation, which has three parts. Essentially, the input problem is mapped into a different space; the new problem is solved in that space; and the solution is mapped back to the original space. After an overview, each stage is described in detail.

3.1 Overview

Suppose we are given four lines $l_k, 1 \le k \le 4$ in $\mathbf{R}^3$, and wish to compute the set of lines that are *incident* on, or intersect, the l_k. By the sidedness relation above, we wish to find all lines s such that $side(s, l_k) = 0$ for all k. Each line l_k, under the Plücker mapping, is mapped to a hyperplane Π_k in $\mathbf{P}^5$. Four such hyperplanes in general position intersect in a *line* $\mathbf{L}$ in $\mathbf{P}^5$. In Plücker coordinates, $\mathbf{L}$ contains the images under $\mathbf{\Pi}$ of all lines, real or imaginary, incident on the four l_k. To find the *real* incident lines in $\mathbf{R}^3$, we must intersect $\mathbf{L}$ with the Plücker quadric (Figure 3). As in three space, a line-quadric intersection may contain 0, 1, 2, or (since the quadric is ruled) an infinite number of points. (In the generic situation shown in the figure, there are two intersection points, each of which is the image of a line in three dimensions intersecting each of the l_k.)

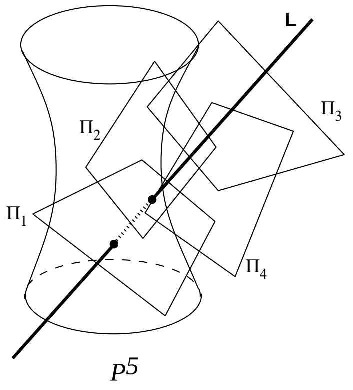

Figure 3: The four Π_k determine a line to be intersected with the Plücker quadric.

Thus the incidence computation has three parts. The first is the mapping of the four input lines to four hyperplanes Π_k in five dimensions, and the intersection of these hyperplanes to form a line (the *null space*) of the Π_k. The second is an intersection of this line with a quadric surface to produce a discrete result. The third part maps this discrete result, represented as Plücker coordinates, to three dimensions for display. We have implemented this computation in the C language using a FORTRAN singular value decomposition routine from Netlib [3]. Figure 1 depicts the algorithm applied to four generic lines. Note that the input lines (thick) are mutually skew, and that each of the two solution lines (thin) pierces the input lines in a distinct order. Both the input and output lines are represented as line segments for the purposes of display.

3.2 Null Space Determination

We formulate the first part of the problem as a singular value decomposition. Each of the input lines l_k corresponds to a six-coefficient hyperplane Π_k under the Plücker mapping. Thus, we must find the null space of the matrix

$$\mathbf{M} = \begin{pmatrix} \Pi_{04} & \Pi_{05} & \Pi_{03} & \Pi_{02} & \Pi_{00} & \Pi_{01} \\ \Pi_{14} & \Pi_{15} & \Pi_{13} & \Pi_{12} & \Pi_{10} & \Pi_{11} \\ \Pi_{24} & \Pi_{25} & \Pi_{23} & \Pi_{22} & \Pi_{20} & \Pi_{21} \\ \Pi_{34} & \Pi_{35} & \Pi_{33} & \Pi_{32} & \Pi_{30} & \Pi_{31} \end{pmatrix}.$$

where Π_{ij} denotes Plücker coordinate j of input line i.

By the singular value decomposition theorem (see, e.g., [6]), this 4×6 real matrix can be written as the product of three matrices, $\mathbf{U} \in \mathbf{R}^{4\times4}$, $\mathbf{\Sigma} \in \mathbf{R}^{4\times6}$, and $\mathbf{V} \in \mathbf{R}^{6\times6}$, with $\mathbf{U}$ and $\mathbf{V}$ orthogonal, and $\mathbf{\Sigma}$ zero except along its diagonal:

$$\mathbf{M} = \mathbf{U\Sigma V}^T = \begin{pmatrix} u_{00} & \cdots & u_{03} \\ \vdots & & \vdots \\ u_{30} & \cdots & u_{33} \end{pmatrix} \begin{pmatrix} \sigma_0 & & & & 0 & 0 \\ & \sigma_1 & 0 & & 0 & 0 \\ & & \sigma_2 & & 0 & 0 \\ 0 & & & \sigma_3 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_{00} & \cdots & \cdots & v_{05} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ v_{50} & \cdots & \cdots & v_{55} \end{pmatrix}.$$

The σ_i are the *singular values* of $\mathbf{M}$, and can be ordered by decreasing magnitude. The number of non-zero[1] σ_i equals the rank r of $\mathbf{M}$. Each zero or elided σ_i corresponds to a row of $\mathbf{V}$; collectively, these rows form the *null space* of $\mathbf{M}$, with dimension $5 - r$.

If $\sigma_3 \neq 0$, $\mathbf{M}$ has rank 4, and its null space has dimension 1 and is spanned by the vectors comprising the last two rows of $\mathbf{V}$, here denoted $\mathbf{F}$ and $\mathbf{G}$ respectively. (This is the generic case.) We parametrize the null space with the function $\mathbf{\Lambda} : \mathbf{P} \to \mathbf{P}^5$ defined as

$$\mathbf{\Lambda}(t) \equiv t\mathbf{F} + \mathbf{G}. \tag{3}$$

The null space property implies that

$$\mathbf{\Lambda}(t) \odot \Pi_k = 0, \qquad 0 \leq k \leq 3, \forall t.$$

The map $\mathbf{\Lambda}$ is injective; it is an isomorphism between $\mathbf{P}$ and the set of all lines (real and imaginary) incident on the l_k.

3.3 Intersection with the Plücker Quadric

Intuitively, $d-1$ hyperplanes in d dimensions intersect to form a line. Formally, the null space $\mathbf{\Lambda}(t)$ defined above is a one-parameter family (a line) of five-dimensional points, all of which have a zero permuted inner product with the Plücker coordinates of the four given lines. However, in general only two of these 5-D points satisfy the Plücker relation (Equation 2), and correspond to real 3-D lines. These are the lines $\mathbf{\Lambda}(t)$ for which:

$$\mathbf{\Lambda}(t) \odot \mathbf{\Lambda}(t) = 0.$$

Substituting from Equation 3 produces a quadratic equation in t:

$$\mathbf{F} \odot \mathbf{F}t^2 + 2\mathbf{F} \odot \mathbf{G}t + \mathbf{G} \odot \mathbf{G} = 0,$$

or

$$at^2 + 2bt + c = 0,$$

[1]Our finite-precision implementation classifies as zero any singular value less than a constant threshold.

where $a = \mathbf{F} \odot \mathbf{F}$, $b = \mathbf{F} \odot \mathbf{G}$, and $c = \mathbf{G} \odot \mathbf{G}$. This is an "even" quadratic with the discriminant $b^2 - ac$, rather than the more familiar $b^2 - 4ac$.

If $a^2 + b^2 + c^2 = 0$, all t are solutions, and the null space line $\mathbf{\Lambda}(t)$ in $\mathbf{P}^5$ lies entirely in the (ruled) Plücker quadric. In this case any linear combination of $\mathbf{F}$ and $\mathbf{G}$ corresponds to a real line incident on the l_k.

If $b^2 - ac < 0$, $\mathbf{\Lambda}$ is disjoint from the Plücker quadric, and there are no real lines incident on the l_k. If $b^2 - ac = 0$, $\mathbf{\Lambda}$ is tangent to the Plücker quadric, and there is a single real line incident on the l_k given by $\mathbf{\Lambda}(\frac{-b}{a})$. Finally, if $b^2 - ac > 0$ there are two lines, $\mathbf{\Lambda}(t_+)$ and $\mathbf{\Lambda}(t_-)$, corresponding respectively to the real roots

$$t_\pm = \frac{-b \pm \sqrt{b^2 - ac}}{a}. \tag{4}$$

3.4 Remapping the Solution Lines

The Plücker coordinates determined in the previous section form a complete representation of the line or lines incident on the input lines. However, this representation is not ideal for all purposes, for example computer graphic display. Thus, the final step of the algorithm remaps the intersection points (identified in the previous step) to lines in three dimensions represented as pairs of distinct points. This is accomplished through the following straightforward operation: the intersection of a line represented by its six Plücker coordinates, and a plane represented by its four coefficients A, B, C, D such that $Ax+By+Cz+D = 0$ ([14], page 205). Our implementation performs repeated line-plane intersections to "clip" the solution line or lines to an axial unit cube centered at the origin.

4 Degeneracies

The four planes in Plücker coordinates arising from dualization of the input lines may not be in general position, i.e., they may intersect in a subspace of $\mathbf{P}^5$ with dimension greater than one. In this case, one or more of the four σ_i will be zero. If n of the σ_i are zero, the set of real and imaginary lines incident on the l_k are spanned by the last $n + 2$ rows of $\mathbf{V}$. This set of points can be parametrized by $\mathbf{P}^{n+1}$. Any real lines in this set must correspond to points satisfying the Plücker relationship (Eq. 2), inducing a quadratic constraint on $\mathbf{P}^{n+1}$. This is a quadratic equation on the line (as we have seen above) for $n = 0$; a conic in the projective plane for $n = 1$; and a quadric surface in projective space for $n \geq 2$. The solution to the quadratic equation can be all of $\mathbf{P}^{n+1}$, empty, reducible or, when $n > 1$, irreducible. If it is reducible, each component can be parametrized by $\mathbf{P}^n$; otherwise it is irreducible, and the entire set of lines can be parametrized by $\mathbf{P}^n$. The following sections describe the various special cases that arise.

4.1 One-Dimensional Line Families

If $\sigma_3 = 0$ and $\sigma_2 \neq 0$ then $\mathbf{M}$ has rank three: only three rows of $\mathbf{M}$ are linearly independent, and its null space has dimension two (it is a 2-D subspace, an ordinary plane). In this case, the set of lines incident on the l_k are those lines whose Plücker coordinates are orthogonal to the first three rows of $\mathbf{V}$. Thus, by the SVD, the last three rows of $\mathbf{V}$ span the space of lines (real and imaginary) incident on the l_k. As above, we parametrize this space with the map $\mathbf{\Lambda}(u, v) : \mathbf{P}^2 \to \mathbf{P}^5$ given by

$$\mathbf{\Lambda}(u, v) = u\mathbf{E} + v\mathbf{F} + \mathbf{G}$$

where $\mathbf{E}$, $\mathbf{F}$ and $\mathbf{G}$ are the last three rows of $\mathbf{V}$. $\mathbf{\Lambda}(u, v)$ parametrizes the real and imaginary incident lines. As before, any real lines incident on the l_k must satisfy

$$\mathbf{\Lambda}(u, v) \odot \mathbf{\Lambda}(u, v) = 0.$$

This is a quadratic equation $q(u, v) = 0$ in the variables u and v, i.e., an ordinary conic. If this conic is a line pair, the set of lines incident on the l_k comprise two 1-parameter families of lines. Otherwise the conic can be parametrized by a single variable t. Thus if $u(t), v(t)$ satisfy $q(u(t), v(t)) = 0$, the incident lines are given by $\mathcal{L}(u(t)\mathbf{E} + v(t)\mathbf{F} + \mathbf{G})$.

4.2 Two-Dimensional Line Families

If $\sigma_3 = 0$, $\sigma_2 = 0$, and $\sigma_1 \neq 0$, the null space of $\mathbf{M}$ has dimension three (it is a 3-D subspace). In this case, the set of real and imaginary lines incident on the l_k can be parametrized by the map $\mathbf{\Lambda}(u, v, w) : \mathbf{P}^3 \to \mathbf{P}^5$ given by

$$\mathbf{\Lambda}(u, v, w) = u\mathbf{D} + v\mathbf{E} + w\mathbf{F} + \mathbf{G},$$

where $\mathbf{D}$, $\mathbf{E}$, $\mathbf{F}$ and $\mathbf{G}$ are the last four rows of $\mathbf{V}$. As before, the real lines satisfy a quadratic equation $q(u, v, w) = 0$ in $\mathbf{P}^3$, i.e., an ordinary quadric. If this quadric is expressible as a plane pair, the set of lines incident on the l_k comprises two 2-parameter families of lines. Otherwise the quadric (the zero surface of the preceding equation) can be parametrized by the projective plane. If $u(t), v(t), w(t)$ satisfy $q(u(t), v(t), w(t)) = 0$, the incident lines are given by $\mathcal{L}(u(t)\mathbf{D} + v(t)\mathbf{E} + w(t)\mathbf{F} + \mathbf{G})$.

4.3 Extreme Degeneracy

Even more extreme degeneracies may arise in practice. For example, the null space of $\mathbf{M}$ may have dimension four (if all of the input lines are non-zero, but identical to scale) or five (if all of the input lines are zero, as when their defining segments have zero length). These cases can be detected by examination of the singular values σ_i as above.

Implementation

Source code for our algorithm appears on JGT's accompanying web site. The code defines a type `plucker` (a vector of six floating-point coefficients), along with the C subroutines:

- `void p_construct(` A`,` B`,` L `)`, to compute the Plücker coordinates L of a line through $3-D$ points A and B;
- `int p_incident(` p`,` q`,` r`,` s`,` T_1`,` T_2 `)`, which computes the line or lines (T_1 and T_2) incident on four input lines p, q, r, and s, and returns an integer degeneracy classification; and
- `void p_deconstruct(` L`,` A`,` B `)` which determines two points A and B in three dimensions lying on a line L described by its Plücker coordinates.

The incidence computation uses the FORTRAN subroutine `dsvdc()` from Netlib [3] to perform stable singular value decomposition. Our implementation handles the case of four input lines in general position (all mutually skew, not lying on the same hyperboloid of one sheet). The algorithm detects and reports, but does not solve, degenerate input instances.

Conclusion

Using a duality transformation relating directed lines in three dimensions and hyperplanes in five dimensions, we have described an algorithm that computes the set of lines intersecting four given lines. The computation amounts to a null space determination of a line lying simultaneously in four hyperplanes in five dimensions, intersection of this line with a quadric surface, and finally remapping of the resulting five-dimensional point(s) to real lines in three dimensions. The null space determination can be cast as a stable singular value decomposition. The line-quadric intersection involves finding the roots of a quadratic equation in one variable. The remapping is straightforward. Finally, we describe the ways in which degenerate input can arise, and the incident line families that result, and give an implementation that solves the generic case and reports degenerate input instances.

Acknowledgments

We are grateful to the late René de Vogelaere for sharing with us his insights about regulae and classical geometry; to Jim Ruppert and Jim Demmel for their helpful advice; to Paul Haeberli, who created Figure 1 from data generated by the implemented algorithm; and to the reviewers for their thoughtful suggestions.

References

[1] D. Avis and M. Doskas. Algorithms for high dimensional stabbing problems. Technical Report SOCS 87.2, McGill University, 1987.

[2] Bernard Chazelle, Herbert Edelsbrunner, Leonidas J. Guibas, and Micha Sharir. Lines in space – combinatorics, algorithms, and applications. In *Proc. 21st ACM Symp. on Theory of Computation*, pages 382–393, 1989.

[3] J. Dongarra and E. Grosse. Distribution of mathematical software via electronic mail. *CACM*, 30(5):403–407, 1987.

[4] Frédo Durand, George Drettakis, and Claude Puech. The visibility skeleton: A powerful and efficient multi-purpose global visibility tool. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 89–100. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.

[5] Ziv Gigus, John Canny, and Raimund Seidel. Efficiently computing and representing aspect graphs of polyhedral objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):542–551, 1991.

[6] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, 1989.

[7] P. Heckbert. *Simulating Global Illumination Using Adaptive Meshing.* PhD thesis, CS Department, UC Berkeley, June 1991.

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

[9] Marco Pellegrini and Peter Shor. Finding stabbing lines in 3-dimensional space. In *Proc. 2nd ACM-SIAM Symposium on Discrete Algorithms*, pages 24–31, 1991.

[10] W.H. Plantinga, C.R. Dyer, and W.B Seales. Visibility, occlusion, and the aspect graph. Technical Report 736, University of Wisconsin at Madison, December 1987.

[11] David Salesin, Dani Lischinski, and Tony DeRose. Reconstructing illumination functions with selected discontinuities. In *Proc. 3rd Eurographics Workshop on Rendering*, pages 99–112, May 1992.

[12] Ken Shoemake. Plücker coordinate tutorial, `http://www.acm.org/tog/resources/rtnews/html/rtnv11n1.html#art3`. In Eric Haines, editor, *Ray Tracing News*, volume 11. ACM Press, July 1998.

[13] D.M.Y. Sommerville. *Analytical Geometry of Three Dimensions.* Cambridge University Press, 1959.

[14] Jorge Stolfi. Primitives for computational geometry. Technical Report 36, DEC SRC, 1989.

[15] Seth Teller. Computing the antipenumbra cast by an area light source. *Computer Graphics (Proc. Siggraph '92)*, 26(2):139–148, 1992.