

Lines Classification in the Conformal Space

$\mathbb{R}^{n+1,1}$

Lilian Aveneau and Ángel F. Tenorio

Abstract Lines classification is the central tool for visibility calculation in dimension $n \geq 2$. It has been previously expressed in Grassmann Algebra, allowing to work with any couple of 2-vectors, which may represent two real lines or not. This article discusses about the nature of lines in the conformal model, searching if such a classification is still valid in $\mathbb{R}^{n+1,1}$. First, it shows that the projective classification can be expressed in terms of a *meet* operator. Then, given two real lines, the classification still works in the conformal model, but also allowing us to propound some techniques to identify lines and circles among general 3-vectors.

1 Introduction

1.1 Motivation

Visibility calculation is a major problem in many applications, for instance in computer graphics, robotics, electromagnetic propagation simulation, *et caetera*. This problem was unsolved for many years, until first advances in two dimensions appear mainly due to the particular line nature in such a space, where lines can be associated to hyperplanes [11].

For a generalization in \mathbb{R}^n , the last 20 years have seen many advances thanks to research in dimension three [2, 5, 9, 10, 12, 13]. This was mainly based on the Plücker coordinates, being used instead of a well-formed definition of line space. Later, using the Grassmann algebra and the projective model, a generalization has

L. Aveneau
University of Poitiers, XLIM/SIC, CNRS, France, e-mail: lilian.aveneau@univ-poitiers.fr

A. F. Tenorio
Pablo de Olavide University, Seville, Spain e-mail: aftenorio@upo.es

been proposed [1, 3]. This definition works in a well-defined line space, where lines are 2-vectors.

The key element of visibility calculation is the line classification, consisting in characterizing the relative orientation of two lines. The latter is based on a dot product which has been proved to be equivalent to computing the orientation of the space spanned by the two lines. One of the most important results concerns the validity of this dot product; *i.e.* the product is still correct even if applied to 2-vectors not encoding real lines at all – which are necessary to group lines in convex sets. –

The question this article aims to answer is to determine the line classification in the conformal model, where lines are defined as particular 3-vectors. This is a necessary step in order to use a hierarchy of bounding sphere or bounding circles around occluders, for instance to accelerate the visibility calculation in practice.

In particular, the line classification can be shown in the projective model as, in fact, an intersection by using the appropriate *join* and *meet* operators. This is equivalent to study the intersection of two 3-vectors in the conformal model, as well as giving a characterization of lines and circles among general 3-vectors.

1.2 Outlines

This article is organized as follows: Section 2 recalls some well-known results for projective lines, starting from the projective model and the line classification working for any pair of 2-vectors. Then, it shows that such a classification is equivalent to intersecting lines. Section 3 details our notation for the conformal model, and presents the conformal line space, corresponding to the 3-vector space – which contains lines, among other things. – Section 4 propounds some results to discriminate lines and circles from other 3-vectors. Section 5 provides some preliminary results with regard to classification of 3-vectors, starting from lines, circles, 3-blades and, at last, 3-vectors. Finally, the last section gives some conclusions and perspectives.

It should be noticed that Geometric Algebra are considered well known by the readers; otherwise, readers can consult Hestenes' books [6, 7].

2 Projective Lines

2.1 Projective Model

This section briefly recalls the basics about the projective model. As many authors use their own notation, we also briefly present the one used in this paper, mostly based on that by Dorst *et al.* [4].

The projective model is based on a Euclidean space of dimension n , denoted by \mathbb{E}^n , with an orthonormal basis (e_1, e_2, \dots, e_n) completed by adding an additional orthogonal vector e_0 with positive signature – implying $e_0^2 = +1$. –

Any Euclidean point – in bold – is embedded in the projective model using the following transformation:

$$\begin{aligned}\mathbb{E}^n &\longrightarrow \mathbb{R}^{n+1} \\ \mathbf{p} &\longrightarrow p_h = \mathbf{p} + e_0\end{aligned}$$

Finally, the pseudoscalar in \mathbb{R}^{n+1} is denoted by $I_{n+1} = e_0 e_1 e_2 \dots e_n$.

2.2 Projective Lines

A line l is a subspace of dimension 1. A projective line can be defined as the linear subspace spanned by two distinct Euclidean points belonging to l :

$$e_0 \wedge (\mathbf{b} - \mathbf{a}) \quad (1)$$

or equivalently by a point and the line direction \mathbf{u} :

$$l = a_h \wedge \mathbf{u} = \mathbf{a} \wedge \mathbf{u} + e_0 \wedge \mathbf{u} \quad (2)$$

Sceptical readers can easily verify that these equivalent expressions correctly encode a line. Hence, a point \mathbf{x} belongs to a line l if and only if:

$$x_h \wedge l = 0$$

In the standard approach, any point \mathbf{x} on the line l belongs to the linear subspace spanned by a and b :

$$\mathbf{x} = \mathbf{a} + \lambda \mathbf{u} = (1 - \lambda) \mathbf{a} + \lambda \mathbf{b}$$

for a given $\lambda \in \mathbb{R}$. Then, anyone can verify point \mathbf{x} belongs to the line:

$$\begin{aligned}x_h \wedge l &= (\mathbf{a} + \lambda \mathbf{u} + e_0) \wedge (\mathbf{a} \wedge \mathbf{u} + e_0 \wedge \mathbf{u}) \\ &= \mathbf{a} \wedge e_0 \wedge \mathbf{u} + e_0 \wedge \mathbf{a} \wedge \mathbf{u} \\ &= 0\end{aligned}$$

Conversely, assuming $x_h \wedge l = 0$ it follows:

$$\begin{aligned}x_h \wedge l = 0 &\Rightarrow x_h = \mu a_h + \lambda \mathbf{u} \text{ with } \lambda, \mu \in \mathbb{R} \\ &\Rightarrow \mathbf{x} + e_0 = \mu(\mathbf{a} + e_0) + \lambda \mathbf{u} \\ &\Rightarrow \mu = 1 \text{ and } \mathbf{x} = \mathbf{a} + \lambda \mathbf{u}.\end{aligned}$$

2.3 Projective Lines Classification

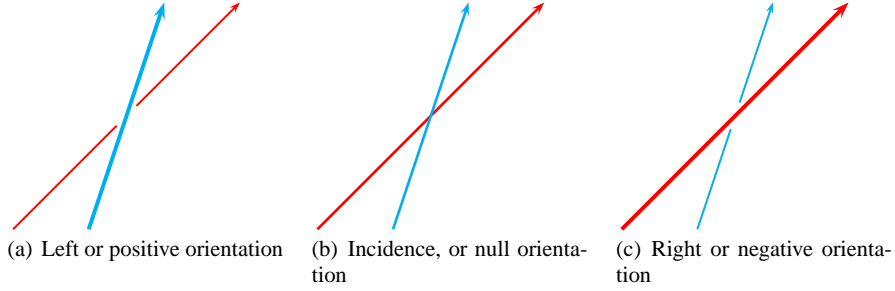


Fig. 1 Lines classification, illustrated by the three different possible orientations of two real lines. Notice that the two lines are oriented in the same way: They intersect – figure (b) – or twist together, turning left (a) or right (c).

Line classification gives the relative orientation of two lines with respect to each other, as depicted in Fig. 1. In Fig. 1(a), the lines twist to the left; in Fig. 1(b), they intersect each other; in Fig. 1(c), each line turns on right around the other.

These three orientations are directly related to the orientation of the subspace spanned by the two lines and consequently directly related to the intersection of lines. By using the *meet* operator, denoted by \vee , we propose to compute such an orientation as follows:

$$\begin{aligned} l_1 \vee l_2 &= (l_1^* \wedge l_2^*)^* \\ l_1 \vee l_2 &= ((l_1 \rfloor \mathbf{J}^{-1}) \wedge (l_2 \rfloor \mathbf{J}^{-1})) \rfloor \mathbf{J}^{-1} \\ &= \alpha \end{aligned} \quad (3)$$

where \mathbf{J} is the *join*, and $\alpha \in \mathbb{R}$ is null. When the two lines intersect, then $\alpha = 0$ since the spanned subspace is empty. When α is strictly negative or positive, the lines do not intersect each other but are twisting. As depicted in Fig. 1, the two lines are then oriented and pass either left or right each other [1].

Let us note that the dualization involved in Equation (3) does not use the pseudoscalar I_{n+1} as *join*, but a normalized version of $l_1 \wedge l_2$. Obviously, if this *join* is null, the lines are intersecting. To avoid such a test and supplementary calculations, Aveneau *et al.* [1] use a dualization by the pseudoscalar and a particular application from the $(n-1)$ -vector space to the 2-vector one, leading to a correct inner product.

Here, we just compute the un-normalised *join* value:

$$\begin{aligned} l_1 \wedge l_2 &= (\mathbf{a} \wedge \mathbf{u} + e_0 \wedge \mathbf{u}) \wedge (\mathbf{b} \wedge \mathbf{v} + e_0 \wedge \mathbf{v}) \\ &= (\mathbf{a} \wedge \mathbf{b} + e_0 \wedge (\mathbf{b} - \mathbf{a})) \wedge \mathbf{v} \wedge \mathbf{u} \end{aligned}$$

It appears to indicate that in a 3-dimensional Euclidean space, the *join* corresponds to the pseudoscalar I_4 , to emphasize the fact that the intersection leads to $\alpha = 0$,

for both intersecting lines – *i.e.* $\mathbf{b} - \mathbf{a} = 0$ – and parallel ones – *i.e.* $\mathbf{u} \wedge \mathbf{v} = 0$. – In dimension $n \geq 4$, the *join* does not correspond to the pseudoscalar I_{n+1} . Indeed, it can be simplified as:

$$\begin{aligned} l_1 \wedge l_2 &= (\mathbf{a} \wedge \mathbf{b} + e_0 \wedge (\mathbf{b} - \mathbf{a})) \wedge \mathbf{v} \wedge \mathbf{u} \\ &\sim I_4^n = \sum_{0 \leq i < j < k < l \leq n} e_i e_j e_k e_l \end{aligned}$$

As conclusion, the *join* operator I_4^n leads to a correct line classification in dimension $n \geq 3$.

3 Conformal Lines

3.1 Conformal Model

This section briefly recalls the basics about the conformal model. As many authors use their own notation, we also briefly present that used in this paper – mostly based on Dorst *et al.* [4]. –

The conformal model is based on a Euclidean space of dimension n , denoted by \mathbb{E}^n with an orthonormal basis (e_1, e_2, \dots, e_n) which is completed by adding two orthonormal vectors e_+ and e_- with positive and negative signatures, respectively. The particular vectors representing the origin and the infinity of \mathbb{E}^n are denoted respectively by $n_o = \frac{1}{2}(e_+ + e_-)$ and $n_\infty = e_- - e_+$. Any Euclidean point – in bold – can be embedded into the conformal model using the following transformation:

$$\begin{aligned} \mathbb{E}^n &\longrightarrow \mathbb{R}^{n+1,1} \\ \mathbf{p} &\longrightarrow p = F(\mathbf{p}) = n_o + \mathbf{p} + \frac{1}{2}\mathbf{p}^2 n_\infty \end{aligned} \quad (4)$$

To avoid mistakes in distance calculations, we will respect the following normalisation condition:

$$n_\infty \cdot p = -1 \quad (5)$$

The Euclidean and conformal pseudoscalars are respectively denoted by $\mathbf{I}_n = e_1 \wedge e_2 \wedge \dots \wedge e_n$ and $I_{n+1,1} = n_o \wedge \mathbf{I}_n \wedge n_\infty$.

3.2 Conformal Line Space

A conformal line is a particular 3-vector [4, 14]; the line passing through the conformal point p with direction \mathbf{u} is expressed as:

$$p \wedge \mathbf{u} \wedge n_\infty \quad (6)$$

Readers can recognize here the projective line, *wedged* with the infinite conformal point.

Therefore, all projective lines are embedded into the conformal 3-vector space. Nevertheless, not all 3-vectors encode real lines. This is also true with 2-vectors in the projective model. Then, we retrieve the same exceptions, corresponding to non decomposable 2-vectors. With the conformal model, we find new exceptions, corresponding to:

- Circles, defined as the outer product of three conformal points, with real lines as particular cases.
- General 3-vectors, not encoding circles or lines, independently on they are 3-blades or not.

4 Lines and Circles Recognition

This section explores the recognition of a 3-blade, encoding an object but not its dual.

4.1 Recognizing a Line

Let us take a line passing through p with direction \mathbf{u} :

$$p \wedge \mathbf{u} \wedge n_\infty = n_o \wedge \mathbf{u} \wedge n_\infty + \mathbf{p} \wedge \mathbf{u} \wedge n_\infty$$

Then, a first and obvious necessary condition to be a line, starting from a 3-vector M , consists to check if it contains an infinite part:

$$n_\infty \wedge M = 0$$

But this condition is not sufficient, since we only retrieve the projective 2-vector part, with its exceptions – for non-blade elements. –

Therefore, while a line is a flat, we can verify that a 3-vector M encodes a line using the well-known conditions about flats in conformal model [4]. Hence, the following lemma allows to determine whether a 3-vector encodes a line or not:

Lemma 1 (3-vectors encoding lines). *A 3-vector M encodes a line in the conformal model if and only if the following three conditions are satisfied:*

1. M is a 3-blade – i.e. fully factorisable. –
2. M is factorisable by n_∞ – i.e. $M \wedge n_\infty = 0$. –
3. The direction of M is non null – i.e. $n_\infty \lrcorner M \neq 0$. –

Conditions 2 and 3 imply that M is a flat, while the first one verifies it is a blade.

4.2 Recognizing a Circle

In the same spirit, since a circle is a round [4], we can check if a 3-vector encodes a circle, giving us the following lemma:

Lemma 2 (3-vectors encoding circles). *A 3-vector M encodes a circle in the conformal model if and only if the following four conditions are satisfied:*

1. M is a 3-blade – i.e. fully factorisable. –
2. M is not factorisable by n_∞ – i.e. $M \wedge n_\infty \neq 0$. –
3. The direction of M is not null – i.e. $n_\infty \lrcorner M \neq 0$. –
4. The square of M is not null – i.e. $M^2 \neq 0$. –

The last three conditions involve that the 3-vector is a round, and so a circle. The first one is equivalent to be a blade.

5 Orientation of 3-vectors

This section extends the previous works on line classification in the projective model to the conformal space. The purpose is to show that the intersection between two lines or – by extension – two 3-vectors, computes the orientation of them. As in the projective case, the choice of the *join* operator is crucial again.

5.1 Orientation of two lines

This section aims to determine the orientation of two lines encoded as 3-blades in the conformal model. In fact, this is trivial since such blades contain all projective 2-blades. Hence, the projective classification is still usable. But, a naive approach is not very powerful, since it consists in extracting the direction and the location of each line, and then in applying the projective solution.

Consequently, we need a more efficient and direct approach. For example, intersecting the two lines [8] and giving the following well-known result:

$$l_1 \vee l_2 = \begin{cases} 0 & \text{if the lines intersect} \\ \alpha n_\infty & \text{if the lines do not intersect} \end{cases}$$

where the *meet* operator is calculated using a *join* similar to that used in the projective model:

$$\mathbf{J} = \frac{l_1 \wedge (n_o \lrcorner l_2)}{(l_1 \wedge (n_o \lrcorner l_2))^2}$$

Indeed, while n_∞ appears in each line, we need to remove one occurrence. The left contraction by n_∞ cannot be used, since n_∞ is a null vector. Using the relation $n_o \lrcorner n_\infty = n_o \cdot n_\infty = -1$, the contraction by n_o allows to cancel n_∞ :

$$\begin{aligned} n_o \rfloor l_2 &= n_o \rfloor (n_o + \mathbf{q}) \wedge \mathbf{v} \wedge n_\infty \\ &= \mathbf{v} \wedge (n_o + \mathbf{q}) \end{aligned}$$

This leads to the following un-normalised *join* value:

$$\begin{aligned} l_1 \wedge (n_o \rfloor l_2) &= (n_o + \mathbf{p}) \wedge \mathbf{u} \wedge n_\infty \wedge \mathbf{v} \wedge (n_o + \mathbf{q}) \\ &= n_\infty \wedge \mathbf{u} \wedge \mathbf{v} \wedge (n_o \wedge (\mathbf{q} - \mathbf{p}) + \mathbf{p} \wedge \mathbf{q}) \\ &\sim n_\infty (\mathbf{I}_4^n + n_o \mathbf{I}_3^n) \end{aligned}$$

where the bold k -vectors \mathbf{I}_k^n contain only Euclidean basis vectors, as in the projective model – without the vector e_0 : –

$$\mathbf{I}_k^n = \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq n} e_{i_1} e_{i_2} \dots e_{i_k}$$

This intersection is not a conformal point, but a scalar multiplied by the point at infinity, which always belongs to the intersection of two lines. This scalar is proportional to the minimal distance between the two lines, and oriented in the same way as the projective model. Hence, we retrieve the projective model orientation of lines, which still can be used in the conformal model. This result is not so surprising, while projective lines are directly embedded into the conformal line space – or 3-vector space. –

Note that, according to [4], for retrieving the intersection point of two intersecting lines, it suffices to use another *join*: When the two lines are not equal, a solution consists in using their common plane [14].

5.2 Orientation of two Circles or a Circle and a Line

This section aims to determine the orientation between two circles and, by extension, a line and a circle. Any circle can be defined as the wedge between 3 of its points, or as the intersection between a hypersphere and a 2-dimensional flat [4]. Although intuitively the intersection between two circles should be a point pair for $n = 3$, using the conformal pseudoscalar $I_{4,1}$ as *join*, the intersection with another circle or line produces a 1-vector. Indeed, there exist two intersection points only when the intersection between one circle and the supporting hyperplane of the second belongs to the second circle too. For this situation, the dual must be computed with the other supporting plane as *join*.

Nevertheless, in the general 3-dimensional case [14] the intersection is the following:

$$\begin{aligned} C_1 \vee C_2 &= X \text{ where } X^2 = 0 \text{ if circles have one intersection} \\ C_1 \vee C_2 &= X \text{ where } X^2 \neq 0 \text{ if circles have no intersection} \\ C_1 \vee C_2 &= 0 \text{ if circles have two intersections} \end{aligned}$$

The non-intersection case resembles to that appearing for two lines: Let us verify if X^2 denotes an orientation between the circles. Indeed, we can be interested in determining if the circles are interwoven or not, or alternatively if a line passes through a disc or outside. In fact, this is the case and can be used to compute a fast ray-tracing intersection test including discs – by avoiding the calculation of the ray-plane intersection, – or even to accelerate triangle or polygon intersection with a bounding circle.

Now, in an arbitrary dimension n , let us calculate the *join* of two circles:

$$\begin{aligned} C_1 \wedge C_2 &= (a \wedge b \wedge c) \wedge (d \wedge e \wedge f) \\ &= (n_o + \mathbf{a} + a^2 n_\infty / 2) \wedge (n_o + \mathbf{b} + b^2 n_\infty / 2) \wedge (n_o + \mathbf{c} + c^2 n_\infty / 2) \\ &\quad \wedge (n_o + \mathbf{d} + d^2 n_\infty / 2) \wedge (n_o + \mathbf{e} + e^2 n_\infty / 2) \wedge (n_o + \mathbf{f} + f^2 n_\infty / 2) \\ &\sim \mathbf{I}_6^n + n_o \mathbf{I}_5^n + n_o \mathbf{I}_4^n n_\infty + \mathbf{I}_5^n n_\infty \end{aligned}$$

This, of course, should be used carefully in dimension $n \leq 4$. For instance, for $n = 3$, the *join* of two general circles is the conformal pseudoscalar $n_o \mathbf{I}_3^n n_\infty = I_{3+1,1}$, *i.e.* the conformal space itself. In higher dimension, this does not remain true.

To calculate the orientation of a line and a circle, the *join* is the following:

$$\begin{aligned} C \wedge l &= (a \wedge b \wedge c) \wedge (n_o + \mathbf{p}) \wedge \mathbf{u} \wedge n_\infty \\ &= (n_o + \mathbf{a} + a^2 n_\infty / 2) \wedge (n_o + \mathbf{b} + b^2 n_\infty / 2) \wedge (n_o + \mathbf{c} + c^2 n_\infty / 2) \\ &\quad \wedge (n_o + \mathbf{p}) \wedge \mathbf{u} \wedge n_\infty \\ &\sim (\mathbf{I}_5^n + n_o \mathbf{I}_4^n) n_\infty \end{aligned}$$

We retrieve the infinite point n_∞ as factor, and a 5-vector without n_∞ . This resembles to the last part of the *join* obtained for intersecting two circles, and that obtained for two lines with a lower dimension.

5.3 3-blades Orientation

This section extends the orientation for general 3-blades, not necessarily encoding lines or circles. In fact, the various types of 3-blades are already well-known. In [4], they were categorized as follows:

- Directions or dual directions – *i.e.* $\mathbf{E} n_\infty$ or $-\mathbf{E}^* n_\infty$. –
- Lines or 1-flats – *i.e.* $T_p[n_o \wedge (\mathbf{u} n_\infty)]$ where \mathbf{u} is a Euclidean vector. –
- Real or imaginary circles or rounds – *i.e.* $T_p[(n_o + \rho^2 n_\infty / 2) \mathbf{E}]$. –
- Dual flats – *i.e.* $T_p[\widehat{\mathbf{E}^*}]$. –
- Tangents and dual tangents – *i.e.* $T_p[n_o \mathbf{E}]$ or $T_p[n_o \mathbf{E}^* (-1)^n]$. –

where T_p is a translation, \mathbf{E} a Euclidean k -vector for $k = 2$ or $k = 3$, \mathbf{E}^* is the dual of \mathbf{E} in the Euclidean space, and the wide hat denotes a grade involution.

For each type of 3-blade, it is always possible to calculate the maximum *join* as we do it for circles, and then to deduce the correct *meet* giving the orientation of the subspace they span. These can be calculated once and then stored to be applied later, after the recognition of 3-blades.

6 Conclusions

This article propounds a generalization of the line classification, or orientation, using the conformal model. The goal is to use this classification for visibility calculations or ray-tracing, for instance.

First, it shows that line classification in the projective model corresponds to an intersection. This result is easier to understand and to prove than the previous solution in [1]. Some criteria are presented to determine if a given 3-vector is a line, or a circle. Then, the intersection between two 3-vectors is studied by considering different cases, like line with line, or circle with either line or circle.

As main result, we can exhibit a process to determine the orientation of the subspace resulting in the intersection of 3-vectors.

As future work, we might use these results in a new visibility calculation framework, where the various occluders can be grouped together into some spheres of circles, leading to a bounding hierarchy, and then allowing to speed up the computation times. Moreover, we also might use these results to accelerate ray-tracing, to accelerate the intersection of a ray with a triangle using its bounding circle for instance.

References

1. Lilian Aveneau, Sylvain Charneau, Laurent Fuchs, and Frédéric Mora. A Framework for n-dimensional Visibility Computations. In Joan Lasenby Léo Dorst, editor, *Guide to Geometric Algebra in Practice*, pages 273–294. Springer, September 2011.
2. Jiri Bittner, Jan Prikryl, and Pavel Slavík. Exact regional visibility using line space partitioning. *Computers & Graphics*, 27(4):569–580, 2003.
3. Sylvain Charneau. *Étude et application des algèbres géométriques pour le calcul de la visibilité globale dans un espace projectif de dimension $n \geq 2$* . PhD thesis, XLIM/SIC, University of Poitiers, France, 2007.
4. Leo Dorst, Daniel Fontijne, and Stephen Mann. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.
5. Frédo Durand, George Drettakis, and Claude Puech. The visibility skeleton: a powerful and efficient multi-purpose global visibility tool. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97*, pages 89–100, 1997.
6. D. Hestenes. *New Foundations for Classical Mechanics*. Kluwer Academic Publishers, 1999.
7. David Hestenes and Garreth Sobczyk. *Clifford Algebra to Geometric Calculus*. Kluwer Academic Publishers, 1992.

8. Joan Lasenby, Anthony N. Lasenby, and Rich Wareham. A Covariant Approach to Geometry using Geometric Algebra. Technical Report CUED/F-INFENG/TR-483, Cambridge University Engineering Department, 2004.
9. Shaun Nirenstein, Edwin H. Blake, and James E. Gain. Exact From-Region Visibility Culling. In *Rendering Techniques*, pages 191–202, 2002.
10. Marco Pellegrini. Ray Shooting and Lines in Space. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of discrete and computational geometry - second edition*, pages 839–856. Chapman & Hall/CRC Press, Boca Raton, FL, USA, 2004.
11. Michel Pocchiola and Gert Vegter. The Visibility Complex. *Int. J. Comput. Geometry Appl.*, 6(3):279–308, 1996.
12. Seth J. Teller. Computing the antipenumbra of an area light source. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '92, pages 139–148, New York, NY, USA, 1992. ACM.
13. Seth J. Teller and Carlo H. Séquin. Visibility preprocessing for interactive walkthroughs. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '91, pages 61–70, New York, NY, USA, 1991. ACM.
14. Richard James Wareham. *Computer Graphics using Conformal Geometric Algebra*. PhD thesis, Robinson College, University of Cambridge, November 2006.