

A Framework for n -dimensional Visibility Computations

L. Aveneau, S. Charneau, L. Fuchs and F. Mora

Abstract This chapter introduces global visibility computation using Grassmann Algebra. Visibility computation is a fundamental task in computer graphics, as in many other scientific domains. While it is well understood in two dimensions, this does not remain true in higher dimensional spaces.

Grassmann Algebra allows to think about visibility at a high level of abstraction, and to design a framework for solving visibility problems in any n -dimensional space, for $n \geq 2$. Contrary to Stolfi's framework which allows only the representation of real lines, it's algebraic nature dealt naturally without any particular cases.

This chapter shows how the space of lines can be defined as a projective space over the bivector vector space. Then line classification, a key point for the visibility computation, is achieved using the exterior product. Actually, line classification turns out to be equivalent to point vs. hyperplane classification relatively to a non-degenerate bilinear form. This ensures well-defined property and computationally robustness.

Using the previous result, the lines stabbing a n -dimensional convex face are characterized. This set of lines appears to be the intersection of the decomposable bivectors set (*i.e.* bivectors that represent a line) and a convex polytope. Moreover, this convex polytope is proved to be minimal. This property allows useful algorithmic improvements.

To illustrate the use of our framework in practice, we present the computation of soft shadows for 3-dimensional illuminated scene.

Lilian Aveneau

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

Sylvain Charneau

XLIM/SIC, CNRS, University of Poitiers, France e-mail: sylvain.charneau@univ-poitiers.fr

Laurent Fuchs

XLIM/SIC, CNRS, University of Poitiers, France e-mail: laurent.fuchs@univ-poitiers.fr

Frédéric Mora

XLIM/SIC, CNRS, University of Limoges, France e-mail: frederic.mora@unilim.fr

1 Problem statement

1.1 About Visibility

Visibility is a fundamental problem in computer graphics. All rendering algorithms aim at simulating the light transport in a virtual environment, which strongly depends on the mutual visibility of each element in the scene. This is clearly illustrated by the following well-known rendering equation:

$$L(x, \omega) = E(x, \omega) + \int_y \rho(x, \omega, x \rightarrow y) L(x, x \rightarrow y) \frac{\cos \theta_x \cos \theta_y}{|x - y|^2} V(x, y) dy$$

The radiance L leaving a point x in the direction ω is the sum of the emitted radiance E at x , plus the reflected light as the sum of the incoming radiance from all the points y in the scene, according to the surface property ρ and the incident angles. In this equation, $V(x, y)$ is the visibility function, whose value is 1 if x and y are visible, 0 otherwise.

As a consequence, the accuracy of the visibility solution has a direct impact on the quality of the result. This explains why visibility is a central question. And it goes beyond the scope of computer graphics: Other domains, such as electromagnetism or acoustics for instance, derive algorithms to simulate wave propagation.

There are many visibility problems. The simplest one is between two points. A classical solution uses a visibility ray, which works in any dimension where such a ray approach is available [9]. But visibility queries can be more complicated. For example: "What parts of the scene can be seen from this point?" or "What parts of the scene can be seen from this region?". In the latter case, the visibility problem becomes very complex. Contrary to the point-to-point visibility query, it is not sufficient to answer "It is visible" or "It is invisible". The challenge is to compute the whole visibility set, *i.e.* a global visibility information between two continuous sets of points. This implies to study all the discontinuities in the visibility that may occur because of the occluders lying between the sets of points. Visibility discontinuities, sometimes called visibility events, happen at the occluder boundaries. They are the frontiers where the visibility changes.

For simplifying global visibility problems, a common approach consists first to sample the continuous sets of points, and then to perform successive point-to-point visibility queries. However, this sampling step introduces noise, altering the quality of the result. Increasing the sample number helps to minimize the problem, but may badly affect the computational time. In addition, notice that a sampling strategy may be unusable. Considering the following problem: "Prove that two continuous sets of points are not mutually visible", an infinite number of samples would be required!

This illustrates the need for algorithms able to solve *exactly* any global visibility problem: On the one hand it ensures high quality results in applications, on the other hand it is the only way to solve some visibility problems.

1.2 The dimension problem

Global visibility problems take place in line-space. For example the visibility of two continuous sets of points correspond to the lines intersecting the two sets without intersecting their occluders. Thus, visibility discontinuities correspond to lines incident to occluder boundaries. As a consequence, the complexity of a visibility problem is strongly related to the dimension of its underlying line-space.

In a 2-dimensional space, the line-space is also 2-dimensional. Global visibility in 2D has been studied for convex objects through the visibility complex [16] and used in different applications such as radiosity computation [14]. Using another line parametrization, Bittner *et al.* [4] focus on the visibility from a region in the plane.

In a 3-dimensional space, the line-space is not 3-dimensional too, but of dimension 5. As a consequence, visibility problems are much more difficult to apprehend, and the generalization of 2-dimensional visibility algorithms is not possible. So, dedicated algorithms were proposed. F. Durand has developed the 3D visibility complex [7], a data structure that encodes the global visibility by tracking all the discontinuities generated by the vertices, edges and faces of a polygonal environment. This data structure illustrates the complexity of the 3-dimensional visibility, but is not practicable due to robustness issues. The visibility skeleton [6] is a derivative of the 3D visibility complex. It is a multi-purpose visibility tool, but it does not encode all the visibility data.

A line in 3D has 4 degrees of freedom, but a 4D parametrization is not possible without singularities (for instance, we can consider the bounding sphere of the scene ; then any line intersects the sphere in 2 different points, and since the sphere is a surface of degree 2, then a line can be defined using 4 parameters ; however, singularities remain at the poles: the azimuthal coordinate can take any value, leading to different coordinates describing a same line). This can make algorithms sensitive to numerical stability. For avoiding this problem, other approaches use the Plücker line parametrization. The Plücker space is a five dimensional projective space embedding all the 3D lines in a 4-dimensional manifold. It is useful to group lines according to the objects they intersect. Pellegrini [15] uses this formalism to find upper bounds on geometric problems involving 3-dimensional lines. In the Plücker space, lines stabbing a sequence of convex polygons can be represented as a convex polytope. This property is used by Teller [18] for computing visibility through a sequence of *portals*, or convex transparent polygons. Nirenstein [13] and Bittner [3] take into account occlusion to compute from-region visibility, further improved by Haumont [8] and Mora [11, 10].

The first practicable global visibility algorithms in 3-dimensional space are quite recent. This area of research is still investigated. If it is quite difficult to apprehend visibility in a 3-dimensional space, it is worse in a 4-dimensional space, *e.g.* in dynamic environments. At present, we are not aware of any practicable algorithms dedicated to 4D space.

1.3 Toward a global visibility framework

This brief overview highlights several difficulties. At first, visibility algorithms are dependent on the geometrical space dimension. The gap of complexity, for example from 2-dimensional to 3-dimensional space, prevents a general approach. In addition, since global visibility is expressed in a line space, the parametrization choice greatly affects the algorithm design, properties and robustness.

Geometry algebra gives the opportunity to think about visibility at a higher level of abstraction. It allows to analyse problems, and to design their solutions regardless of the dimension of space, using a single approach.

In this chapter, we propose a global visibility framework based on a n -dimensional line space, defined using Grassmann Algebra [5]. While this is a classical definition of lines in mathematics, it remains uncommon in computer graphics. Thanks to this formalism, we prove a major theorem on the representation of a set of lines by a convex polytope. Next, we propose a generalization of Mora's work [10], into a n -dimensional visibility framework. Finally, as an application, we explain how it can be used to compute very high quality soft shadows.

2 Line spaces

For computing visibility between objects, let us denote \mathcal{G}_n the n -dimension geometrical space of the geometric objects. It is embedded into the projective space \mathbb{P}^n . As \mathbb{P}^n is built from \mathbb{R}^{n+1} , linear subspaces of \mathbb{P}^n can be represented by elements of $\wedge(\mathbb{R}^{n+1})$.

2.1 n -dimensional lines

A line is, whatever the dimension of the space it belongs to, a 1-dimensional subspace. It expresses a dependency between two distinct points. So, we can formulate the following definition:

Definition 1. A n -dimensional line, passing through two projective points A and B of \mathcal{G}_n with respective 1-vector coordinates a and b , is represented by the exterior product $a \wedge b$.

Example 1 As an example, let us considering in 2 dimensions the line going through the points of homogeneous coordinates $(1, 0, 1)$ and $(2, 1, 1)$. Using the exterior product, it follows that the expression of this line in $\wedge(\mathbb{R}^3)$ is $\mathbf{e}_0 \wedge \mathbf{e}_1 - \mathbf{e}_1 \wedge \mathbf{e}_2 - \mathbf{e}_2 \wedge \mathbf{e}_1$, where the 1-vectors $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ form the basis of $\wedge^1(\mathbb{R}^3)$.

Example 2 In computer graphics, the 3-dimensional lines are most known using Plücker coordinates. In fact, they can be retrieved using definition 1. Using again a

homogeneous notation, a point P is denoted by 4 coordinates using the vector: $p = (x, y, z, w)^T$. The line going through A and B with respective coordinates $(x_a, y_a, z_a, 1)$ and $(x_b, y_b, z_b, 1)$ is known as [15]:

$$\begin{pmatrix} x_b - x_a \\ y_b - y_a \\ z_b - z_a \\ y_a z_b - y_b z_a \\ z_a x_b - z_b x_a \\ x_a y_b - x_b y_a \end{pmatrix}$$

With our definition, the same line through A and B is defined using the Grassmann exterior product, as $\Pi_{AB} = a \wedge b$ – see Exercise 1 –. Then, it is quite easy to show that Plücker coordinates are coordinates in $\Lambda^2(\mathbb{R}^4)$.

2.2 From line to line space

The elements of $\Lambda^k(\mathbb{R}^{n+1})$, for $k \leq n$, are homogeneous: If $\mathcal{K} \in \Lambda^k(\mathbb{R}^{n+1})$ represents a subspace of \mathbb{R}^{n+1} , then $\mathcal{K}' = \lambda \mathcal{K}$, for $\lambda \in \mathbb{R}^*$, represents the same subspace. Hence, $\mathbb{P}^{\binom{n+1}{k}-1} = \mathbb{P}(\Lambda^k(\mathbb{R}^{n+1}))$ is the space of the 1-subspaces of $\Lambda^k(\mathbb{R}^{n+1})$. Each point of $\mathbb{P}^{\binom{n+1}{k}-1}$ represents a unique linear manifold of \mathfrak{G}_n . This leads to the following definition of the line space \mathfrak{L}_n of \mathfrak{G}_n :

Definition 2. The space of lines of \mathfrak{G}_n , denoted by \mathfrak{L}_n , is the projective space $\mathbb{P}(\Lambda^2(\mathbb{R}^{n+1}))$.

From this definition, the line space is a projective space of dimension $\binom{n+1}{2} - 1$. With \mathfrak{G}_2 , the line space is of dimension $\binom{3}{2} - 1 = 2$, while with \mathfrak{G}_3 , it corresponds to the classical Plücker space of dimension 5. This is directly related to the dimension problem, as presented in Sect. 1.2.

Considering again the example 1, the line passing through the points with homogeneous coordinates $(1, 0, 1)$ and $(2, 1, 1)$ has coordinates $(1, -1, -1)$ using the basis $(\mathbf{e}_0 \wedge \mathbf{e}_1, \mathbf{e}_1 \wedge \mathbf{e}_2, \mathbf{e}_2 \wedge \mathbf{e}_0)$ of $\Lambda^2(\mathbb{R}^3)$.

2.3 About the Grassmannian

From previous works on visibility computations [19, 13], it is well known that 3-dimensional Plücker lines do not fill all the space line \mathfrak{L}_3 . The mapping of 3-dimensional lines to the Plücker space is not surjective. Indeed, a line must pass through at least two distinct points, and then is represented as a 2-blade. They are all located on the Grassmannian $G^{\mathbb{R}}(2, 4)$ – or Grassmann manifold –, defined as the set of all 2-subspaces of \mathbb{R}^4 . It is also the set of all decomposable bivectors, or

2-blades. In dimension n , the set of *actual* lines are located on the Grassmannian $G^{\mathbb{R}}(2, n+1)$.

For deciding if a given point of \mathfrak{L}_n is on the Grassmannian, and so is a actual line, it is sufficient to verify it is a decomposable vector. The following theorem gives an easy way to solve this:

Theorem 1. *Let $\wedge(\mathbb{R}^{n+1})$ be the Grassmann algebra. A non zero bivector M from $\wedge(\mathbb{R}^{n+1})$ is a 2-blade – a decomposable vector – if and only if $M \wedge M = 0$.*

Proof. If M is a 2-blade, then there exist two linearly independent vectors m_1 and m_2 from \mathbb{R}^{n+1} so that $M = m_1 \wedge m_2 \neq 0$. Then, $M \wedge M = m_1 \wedge m_2 \wedge m_1 \wedge m_2$, and using the antisymmetry property of the exterior product, $M \wedge M = 0$.

Now, assuming M is not decomposable, by definition it can be written as a finite sum of linearly independent 2-blades: $\sum_{i=1}^p M_i$, $\binom{n+1}{2} \geq p \geq 2$. It follows that:

$$M \wedge M = \left(\sum_{i=1}^p M_i \right) \wedge \left(\sum_{j=1}^p M_j \right) \quad (1)$$

$$= 2 \sum_{i < j} M_i \wedge M_j \quad (2)$$

Since all these terms are 4-vectors linearly independent, then $M \wedge M \neq 0$. \square

This theorem helps for computing the intersection between a set of lines in \mathfrak{L}_n and the Grassmannian, for instance to decide if it contains at least one actual line, or can be dropped in future computations.

2.4 Line orientation

Previous works on visibility computation use line orientation as a key element. In Grassmann algebra, it is expressed using the exterior product, which expresses the dependency between vector subspaces.

Property 1. Let M and M' be two projective linear subspaces of \mathfrak{G}_n . Their intersection is non empty if and only if $M \wedge M' = 0$.

This property can be applied to a bivector L and a $(n-1)$ -vector F of $\wedge(\mathbb{R}^{n+1})$, allowing to check if a line is incident to the boundary of an occluder, *i.e.* a $(n-2)$ -variety or flat. For instance, as proposed in Exercise 3, in three dimensions it is possible to check that the Plücker relation corresponds to the test $L \wedge H$, where H is also a line since $n - 1 = 2$. A particular case of this first property is the following:

Property 2. Let M and M' be respectively a k -vector and a $(n - k + 1)$ -vector, representing two projective linear subspaces of \mathfrak{G}_n . Their relative orientation is denoted by the sign of $\lambda \in \mathbb{R}$ using the exterior product $M \wedge M' = \lambda I$, where I is the pseudoscalar in $\wedge(\mathbb{R}^{n+1})$.

For instance, in 2 dimensions, it allows to check if a directed line turn clockwise or counterclockwise with respect to a point. This property is essential for computing the line stabbing a convex $(n-1)$ face, as presented in Sect. 3.1.

In order to work with lines in \mathcal{L}_n , we need a similar product using only bivectors. Moreover, we want a robust solution, working with actual lines, but also with any bivector. In fact, it is possible to express the property 2 into the line space, using only bivectors and the inner product. This is based on the duality between $\wedge^2(\mathbb{R}^{n+1})$ and $\wedge^{n-1}(\mathbb{R}^{n+1})$.

Theorem 2. *Let δ be an isomorphism from $\wedge^2(\mathbb{R}^{n+1})$ to $\wedge^{n-1}(\mathbb{R}^{n+1})$, such that $\delta(l) = l \rfloor I_{n+1}$, where \rfloor denotes the left contraction and I_{n+1} is the pseudoscalar over $\wedge(\mathbb{R}^{n+1})$. Then, the inner product between any bivectors L_1 and L_2 is equivalent to check the orientation of the line L_1 with a $(n-2)$ -flat :*

$$L_1 \cdot L_2 \equiv L_1 \wedge \delta(L_2)$$

up to an identification of pseudoscalars and scalars in $\wedge(\mathbb{R}^{n+1})$.

From this result, it follows two immediate and important properties:

1. Since the inner product is non degenerate, there is no singularity. It asserts that the line orientation test is always defined, whatever the line and the flat are.
2. The duality does not depend on the Grassmannian: The previous property is also valid for non decomposable bivectors and $(n-1)$ -vectors, or equivalently, when they do not respectively represent a line and a $(n-2)$ -flat.

The first property plays a fundamental role in our visibility framework. Firstly, it explains that the visibility computations work for all dimensions and all configurations, without any singularity, ensuring the generality of this approach. Secondly, it is computationally simple, as is it reduced to evaluations of a particular scalar product of vectors of dimension $\binom{n+1}{2}$, this ensures its robustness.

The n -dimensional proof of this theorem is left to the reader, but we illustrate it in 2 dimensions. Let $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ be an orthogonal basis of \mathbb{R}^3 , and $\mathbf{e}_0 \wedge \mathbf{e}_1 \wedge \mathbf{e}_2$ be the pseudoscalar for $\wedge(\mathbb{R}^3)$. Let $(\mathbf{e}_1 \wedge \mathbf{e}_2, \mathbf{e}_2 \wedge \mathbf{e}_0, \mathbf{e}_0 \wedge \mathbf{e}_1)$ be a basis of $\wedge^2(\mathbb{R}^3)$. We know that any 2-dimensional line can be represented using 3 coordinates in such a basis. The left contraction is used to define the isomorphism $\delta : \wedge^2(\mathbb{R}^3) \mapsto \wedge^1(\mathbb{R}^3)$ as:

$$\begin{aligned} \delta(\mathbf{e}_1 \wedge \mathbf{e}_2) &= (\mathbf{e}_1 \wedge \mathbf{e}_2) \rfloor \mathbf{e}_0 \wedge \mathbf{e}_1 \wedge \mathbf{e}_2 = \mathbf{e}_0 \\ \delta(\mathbf{e}_2 \wedge \mathbf{e}_0) &= (\mathbf{e}_2 \wedge \mathbf{e}_0) \rfloor \mathbf{e}_0 \wedge \mathbf{e}_1 \wedge \mathbf{e}_2 = \mathbf{e}_1 \\ \delta(\mathbf{e}_0 \wedge \mathbf{e}_1) &= (\mathbf{e}_0 \wedge \mathbf{e}_1) \rfloor \mathbf{e}_0 \wedge \mathbf{e}_1 \wedge \mathbf{e}_2 = \mathbf{e}_2 \end{aligned}$$

So, any 2-dimensional line can be mapped to a 1-vector using δ , and conversely with δ^{-1} . Without loss of generality, let $A : (\alpha_0, \alpha_1, \alpha_2)$ and $B : (\beta_0, \beta_1, \beta_2)$ be two lines, *i.e.* bivectors. Using the anticommutativity of the exterior product, it follows:

$$\begin{aligned} A \wedge \delta(B) &= (\alpha_0 \mathbf{e}_1 \wedge \mathbf{e}_2 + \alpha_1 \mathbf{e}_2 \wedge \mathbf{e}_0 + \alpha_2 \mathbf{e}_0 \wedge \mathbf{e}_1) \wedge (\beta_0 \mathbf{e}_0 + \beta_1 \mathbf{e}_1 + \beta_2 \mathbf{e}_2) \\ &= (\alpha_0 \beta_0 + \alpha_1 \beta_1 + \alpha_2 \beta_2) \mathbf{e}_0 \wedge \mathbf{e}_1 \wedge \mathbf{e}_2 \\ &\equiv A \cdot B \end{aligned}$$

Using any isomorphism between \mathbb{R}^3 and $\wedge^3(\mathbb{R}^3)$, the obtained expression can be recognized as a classical inner product in $\wedge^2(\mathbb{R}^3)$.

2.5 Dual line representation

Theorem 2 has a nice and useful interpretation in the line-space \mathfrak{L}_n .

As a classical result of vector algebra, in a vector space E of dimension m , it is well known that the set of vectors orthogonal to any other vector x (the set of v such that $v \cdot x = 0$, where \cdot is the inner product), describes a vector subspace of dimension $m - 1$ in E , *i.e.* a hyperplane. Transposed to our problem, this classical result means that each $(n - 1)$ -vector (*i.e.* $\delta(L_2)$ in the theorem statement) can be dually associated to a hyperplane in $\wedge^2(\mathbb{R}^{n+1})$ (the set of bivectors L orthogonal to L_2 in $\wedge^2(\mathbb{R}^{n+1})$, *i.e.* such that $L \cdot L_2 = 0$), which corresponds to a projective hyperplane in the line-space \mathfrak{L}_n . In dimension 2, it can be remarked that some similarities exist between this model of line-space \mathfrak{L}_2 with the well known dual plane where lines map to points and conversely points map to lines. This reveals that classifying a line L against a $(n - 2)$ -flat F , by computing the sign of the product $L \wedge F$, for the bivector L and the $(n - 1)$ -vector F , can always be seen as determining in which half-space is the point L of \mathfrak{L}_n , according to the oriented hyperplane $\delta^{-1}(F)$ associated to F in \mathfrak{L}_n . The product is zero if and only if L is a point on the hyperplane $\delta^{-1}(F)$.

This interpretation will be particularly helpful in Sect. 3.2, to give a geometrical significance to the global visibility computation and representation, which only makes sense in the line-space.

3 Visibility in \mathfrak{L}_n

3.1 Lines stabbing a convex $(n - 1)$ -face

The following theorem unambiguously characterizes the set of lines stabbing a convex face in any dimension n .

Theorem 3. *Let \mathbf{F} be a convex $(n - 1)$ -face in \mathfrak{G}_n , supported by the hyperplane \mathcal{H}_F (*i.e.* a $(n - 1)$ -flat in \mathfrak{G}_n) and bounded by the $(n - 2)$ -flats f_i for $i \in [1, \dots, r]$. The flats f_i have two orientations such that for any line L , $L \cap \mathcal{H}_F \neq L$, L stabs \mathbf{F} if and only if one of the following two properties is verified:*

$$\forall i \in [1 \dots r], L \wedge f_i \geq 0 \quad (3)$$

$$\forall i \in [1 \dots r], L \wedge f_i \leq 0 \quad (4)$$

Let $\mathcal{S}_{\mathbf{F}}$ be the set of lines stabbing the face \mathbf{F} .

The proof of the theorem is based on the following remarks. Firstly, \mathbf{F} is a convex polytope, delimited by the flats f_i and restricted to the hyperplane \mathcal{H}_F in \mathfrak{G}_n . Secondly, when a line L does not lay on \mathcal{H}_F , it has one and only one intersection point with \mathcal{H}_F , at infinity if L is parallel to \mathcal{H}_F . Thirdly, L stabs the face \mathbf{F} if and only if its intersection point with \mathcal{H}_F is in \mathbf{F} .

Proof. The line L corresponds to a 2-vector and the hyperplane \mathcal{H}_F to a n -vector in $\wedge(\mathbb{R}^{n+1})$. Since $n+2 > n+1$, then $L \wedge \mathcal{H}_F = 0$. So, there is always an intersection between L and \mathcal{H}_F , either of dimension 1 (a point) or 2 (L itself). This allows to propose the following lemma:

Lemma 1. *Every line L in \mathfrak{G}_n intersects \mathcal{H}_F in a unique point, except if L lays in \mathcal{H}_F .*

Let P_\cap be the intersection of L with \mathcal{H}_F . Assuming L does not lay on \mathcal{H}_F , P_\cap is a non zero 1-vector. Obviously, L stabs \mathbf{F} if and only if P_\cap is inside \mathbf{F} . Let P_γ be any other point on L out of \mathcal{H}_F , such that $P_\gamma \wedge \mathcal{H}_F > 0$, i.e. P_γ is in the positive half-space of \mathcal{H}_F . We can write $L = P_\cap \wedge P_\gamma$. It follows:

$$\forall i \in [1 \dots r], \quad P_\cap \wedge P_\gamma \wedge f_i = P_\cap \wedge (P_\gamma \wedge f_i)$$

By hypothesis, P_γ is not incident to f_i , and the n -vector $P_\gamma \wedge f_i \neq 0$ represents a hyperplane in \mathfrak{G}_n . The sign of the pseudoscalar $P_\cap \wedge (P_\gamma \wedge f_i)$ indicates in which half-space of the hyperplane $P_\gamma \wedge f_i$ the point P_\cap is.

Let \mathcal{P} be the polytope generated by \mathbf{F} and the vertex P_γ . Since \mathbf{F} is convex, so is \mathcal{P} : It is the intersection between \mathcal{H}_F^+ and the positive half-spaces associated to the hyperplanes $P_\gamma \wedge f_i$, $i \in [1 \dots r]$, for a particular but consistent orientation of them. By hypothesis, $P_\gamma \wedge \mathcal{H}_F$ is a non zero pseudoscalar. The orientation of each hyperplane $P_\gamma \wedge f_i$ can be determined from any point P_F into \mathbf{F} , such that $P_F \wedge (P_\gamma \wedge f_i)$ are only positively oriented pseudoscalars. From $P_F \wedge (P_\gamma \wedge f_i) = -P_\gamma \wedge P_F \wedge f_i$, and since $P_F \wedge f_i$ generates the flat \mathcal{H}_F , then $P_F \wedge f_i$ has an opposite orientation than the one of \mathcal{H}_F , since $\forall i \in [1 \dots r], P_F \wedge f_i = \lambda F, \lambda < 0$. This shows that the orientations of the flats f_i can be determined uniquely from those of \mathcal{H}_F and the position of P_γ relatively to \mathcal{H}_F .

The orientation of the flats f_i allows to state that a point P of \mathfrak{G}_n is in the polytope \mathcal{P} if and only if $P \wedge \mathcal{H}_F \geq 0$ and $P \wedge (P_\gamma \wedge f_i) > 0$, $\forall i \in [1 \dots r]$. In particular, this concerns every point in \mathbf{F} , including P_\cap . Then, P_\cap is a point of \mathcal{P} if and only if it:

$$P_\cap \wedge (P_\gamma \wedge f_i) \geq 0 \quad \forall i \in [1 \dots r]$$

In other words, L stabs the face \mathbf{F} if and only if:

$$L \wedge f_i \geq 0 \quad \forall i \in [1 \dots r]$$

Considering the opposite line $L' = P_\gamma \wedge P_\cap$, obviously it stabs \mathbf{F} if and only:

$$L' \wedge f_i \leq 0 \quad \forall i \in [1 \dots r]$$

This result does not depend on P_η with respect to the sign of the half-space associated to the hyperplane \mathcal{H}_F , it remains valid for all lines L not in \mathcal{H}_F .

Changing the orientation of some but not all of the flats f_i makes the previous result false. On the contrary, changing the orientation of all of them – or equivalently the orientation of \mathcal{H}_F or the position of P_η relatively to \mathcal{H}_F – only interchanges all signs in the in-equations, and so the result remains true. This means there are only two valid orientations for the flats f_i . \square

Theorem 3 has a fundamental consequence: It gives an algebraic method for determining whether or not a line stabs a given face in any n -dimensional space. This method has no singularity, since our algebraic framework also handles flats at infinity. However, the lines laying on the hyperplane \mathcal{H}_F are excluded, since they cannot properly “stab” \mathbf{F} . But this distinction does not impact the visibility computation, both from a theoretical and an algorithmic point of view.

Moreover, this theorem has a useful interpretation in \mathcal{L}_n . It first reveals what the visibility computation through faces relies on, geometrically, and then indicates what kind of data-structures and algorithms can be used to compute the visibility in practice. The following section aims to explain this interpretation.

3.2 Convex cells and visibility events in the line-space

3.2.1 Interpretation in \mathcal{L}_n and consequences

Theorem 3 has a suitable geometrical meaning in \mathcal{L}_n . By duality, every flat f_i bounding the face \mathbf{F} can be associated to a unique hyperplane f_i^* in \mathcal{L}_n , $\forall i \in [1 \dots r]$. Then, by choosing a positive orientation for the flats, Theorem 3 implies that $\mathcal{S}_{\mathbf{F}}$, the set of lines stabbing \mathbf{F} in \mathcal{L}_n , is the intersection between $G^{\mathbb{R}}(2, n+1)$ and the convex polytope defined as the intersection of the positive half-spaces delimited by the hyperplanes f_i^* . This is a useful result, as convex polytopes have the following well known properties in computational geometry:

- They have multiple representations: A hyperplane set, a vertex set, and a face lattice.
- It can easily be determined if a point is either inside or outside a polytope.
- It can be easily determined if two polytopes intersect each other.
- Boolean operations are expressed as geometrical computations, such as split, intersections, *etc.* ...

Nevertheless, a single face \mathbf{F} is not sufficient to define a polytope: The hyperplanes $f_i^* \forall i \in [1 \dots r]$ delimit a region in \mathcal{L}_n partially bounded by infinity. This is stated by the dimension of \mathcal{L}_n (see Sect. 2.3): While at least $2n - 1$ hyperplanes are required to define a simplex in \mathcal{L}_n , an $(n - 1)$ -face can only have n facets in general (*i.e.* independent) positions, for instance those of a $(n - 1)$ -simplex in \mathcal{G}_n . As a consequence of this closure by the infinity in \mathcal{L}_n , it becomes impossible to determine a convex-hull representation of the polytope $\mathcal{S}_{\mathbf{F}}$.

While this interpretation leads to some interesting properties, it also illustrates the fundamental role of the Grassmannian $G^{\mathbb{R}}(2, n+1)$ and its embedding line-space \mathcal{L}_n . Indeed, the polytope representing the lines stabbing some faces also contains points outside the Grassmannian. Then, representing the lines stabbing faces in \mathfrak{G}_n by a convex polytope in \mathcal{L}_n , is only possible by considering the whole line-space \mathcal{L}_n , but not the Grassmannian $G^{\mathbb{R}}(2, n+1)$ alone. This mainly explains why the Stolfi framework [17], which only represents points located on the Grassmannian, is not suitable for computing the global visibility. On the contrary, by enabling computations on non decomposable multivectors, geometric algebras make the global visibility computation sum up to boolean operations on convex polytopes in \mathcal{L}_n .

3.2.2 Global visibility in \mathfrak{G}_n as convex cells in \mathcal{L}_n

Extending this representation to two or more faces is straightforward. In the example depicted in Fig. 1, \mathbf{A} , \mathbf{B} and \mathbf{O} are three edges in \mathfrak{G}_2 , with bounding vertices $i \in [1 \dots 6]$. In \mathfrak{G}_2 , these vertices are associated to hyperplanes that subdivide the line-space into cells, grouping together the lines stabbing the same edges. Fig. 1 shows two such cells: $\mathcal{P}_{\mathbf{AB}}$ representing lines stabbing \mathbf{A} and \mathbf{B} but missing \mathbf{O} ; $\mathcal{P}_{\mathbf{AOB}}$ representing lines stabbing \mathbf{A} , \mathbf{O} and \mathbf{B} . It must be noticed that $\mathcal{P}_{\mathbf{AB}}$ completely describes the global visibility between \mathbf{A} and \mathbf{B} , by taking into account the occlusion by \mathbf{O} . This example shows that visibility in \mathfrak{G}_2 can be described in \mathcal{L}_2 by a set of convex polytopes, obtained using Theorem 3.

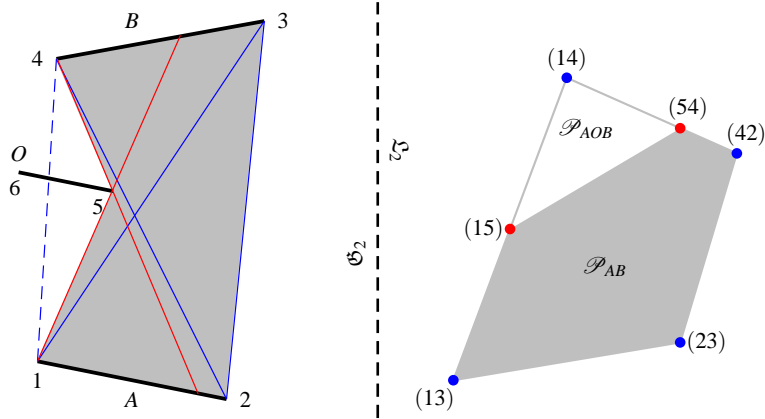


Fig. 1 Visibility computation and representation. Left (a): In \mathfrak{G}_2 , the edges \mathbf{A} and \mathbf{B} are partially hidden by \mathbf{O} . Right (b): In \mathcal{L}_2 , the lines stabbing the three edges is the convex cell $\mathcal{P}_{\mathbf{AOB}}$, while the lines stabbing \mathbf{A} and \mathbf{B} but not \mathbf{O} is represented by the convex cell $\mathcal{P}_{\mathbf{AB}}$. These two cells or polytopes are obtained from the intersection of the positive half-spaces associated to the six vertices bounding the edges.

Since Theorem 3 does not depend on the geometric space dimension, it can be applied to compute the visibility in \mathfrak{G}_n : Visibility through some faces in \mathfrak{G}_n can always be represented by a subdivision of \mathfrak{L}_n in cells which group together the lines stabbing the same faces. The boundary of the cells are then the hyperplanes associated to the $(n-2)$ -flats which bound the objects in \mathfrak{G}_n .

However, we show in Sect. 4.2 that some special configurations prevent grouping two or more faces in only one convex cell or polytope in \mathfrak{L}_n . The distinction and description of these degenerate cases come as a part of the proof of the minimal polytope solution. They give a precise understanding of how lines in the line-space subdivision are grouped together.

3.2.3 Visibility events in \mathfrak{L}_n

According to F. Durand [6], a visual event is defined as the locus where visibility changes in \mathfrak{G}_n . This notion is central in many approaches concerning visibility computation, since both the visibility modification and topology knowledges are sufficient to fully describe the visibility. In practice, a visual event appears as a line tangent to a finite number of geometrical objects. The degree of freedom gives supplementary information, leading to the k -visual event notion.

As depicted in the Fig. 1, vertex 5 is a locus with important visibility variations. The red lines 15 and 54 are two examples of visual events that separate the visibility for all the lines passing through 5. In \mathfrak{L}_2 , they become two 1-vectors that form a part of the cells \mathcal{P}_{AOB} and \mathcal{P}_{AB} , as they ly on a common 2-vector, the dual of 5. Obviously, this is a general rule: The visual events are located on the cell boundaries. It comes from the visual event definition, and Theorem 3.

In n dimensions, only the *real* visual events are of interest, so the whole cell boundary is not interesting. In \mathfrak{G}_n , the visual events are located on the Grassmannian too. Then, a k -visual event is a k -submanifold located at the intersection between $G^{\mathbb{R}}(2, n+1)$ and a cell in \mathfrak{L}_n . This shows that the visibility is fully described using a partition in \mathfrak{L}_n .

4 The minimal polytope

4.1 Minimal polytope interest

All the previous approaches fail to give the minimal set of lines stabbing two convex faces in \mathfrak{G}_n , for $n > 2$. The Grassmann algebra allows to define and to compute the minimal polytope enclosing this set of lines. This is a key for our visibility framework, as it ensures computation efficiency.

Let us enumerate some properties and goals of a minimal polytope representation, from both the theoretical and practical points of view:

1. It procures a vertex representation of the polytope containing the lines stabbing two faces. This is useful for applications needing to split polytopes, to detect collisions between them, or to classify them according to some hyperplanes in the line-space.
2. By splitting the minimal polytope with hyperplanes, such a vertex representation can be extended for representing lines stabbing more than two faces.
3. The minimal polytope is a general solution, in any dimension, to the open problem stated in three dimensions [13]. It is also the most appropriate to avoid the splittings leading to polytopes that do not represent any line in \mathfrak{G}_n , *i.e.* that do not intersect the Grassmannian in \mathfrak{L}_n .
4. From the polytope vertices, all the faces in the polytope boundary (edges, hyperplanes ...) and their incidences can be computed.
5. It unveils the case where a single polytope can not be used to represent the visibility through two polygons. These *degenerate* cases appear in previous 3 dimensional works [13], and are generalized in \mathfrak{G}_n in this paper.

4.2 The minimal polytope for two convex faces

Let \mathbf{A} and \mathbf{B} be two convex $(n - 1)$ -faces in \mathfrak{G}_n , and a_1, \dots, a_q and b_1, \dots, b_r their respective vertices.

Definition 3. The minimal polytope, denoted \mathcal{M}_A^B , represents the set of lines \mathcal{S}_A^B stabbing \mathbf{A} and \mathbf{B} in \mathfrak{L}_n . It is the convex polytope with the following properties:

1. $\mathcal{S}_A^B \subseteq \mathcal{M}_A^B$
2. $\mathcal{M}_A^B \cap G^{\mathbb{R}}(2, n + 1) \subseteq \mathcal{S}_A^B$
3. If \mathcal{P}_A^B is a convex set in \mathfrak{L}_n such that $\mathcal{S}_A^B \subseteq \mathcal{P}_A^B$, then $\mathcal{M}_A^B \subseteq \mathcal{P}_A^B$.

The properties 1 and 2 mean the polytope \mathcal{M}_A^B is a representation of \mathcal{S}_A^B in \mathfrak{L}_n , *i.e.* a line L stabs \mathbf{A} and \mathbf{B} if and only if its representation in \mathfrak{L}_n is contained in \mathcal{M}_A^B . The third property indicates that \mathcal{M}_A^B is the minimal polytope: There does not exist another convex polytope representing \mathcal{S}_A^B and contained in \mathcal{M}_A^B .

The following theorem gives a computational characterization of the minimal polytope for two faces in some canonical configurations, and indicates the non existence of any polytope for the other configurations.

Theorem 4. Let \mathcal{H}_A and \mathcal{H}_B be respectively the supporting planes of the faces \mathbf{A} and \mathbf{B} . If \mathcal{H}_A and \mathcal{H}_B do not respectively intersect the faces \mathbf{B} or \mathbf{A} , or only on their boundary, then the minimal polytope \mathcal{M}_A^B is the convex hull of the lines $L_{ij} = a_i \wedge b_j$, $(i, j) \in [1 \dots q] \times [1 \dots r]$ from the vertices of \mathbf{A} to the ones of \mathbf{B} . Otherwise, the set of lines stabbing \mathbf{A} and \mathbf{B} cannot be represented by any convex polytope in \mathfrak{L}_n .

4.3 Proof of the minimal polytope solution

To prove Theorem 4, we consider the two $(n-1)$ -faces \mathbf{A} and \mathbf{B} , with respective vertices a_1, \dots, a_q and b_1, \dots, b_r . We suppose that these faces are supported by the hyperplanes \mathcal{H}_A and \mathcal{H}_B , and bounded by the $(n-2)$ -flats $f_1^a \dots f_s^a$ and f_1^b, \dots, f_t^b respectively.

The proof is decomposed into three steps:

1. If the polytope \mathcal{M}_A^B exists, then it is minimal.
2. If the hyperplanes \mathcal{H}_A and \mathcal{H}_B do not intersect the faces \mathbf{B} and \mathbf{A} respectively, or only their boundary, then:
 - a. $\mathcal{S}_A^B \subset \mathcal{M}_A^B$
 - b. $\mathcal{M}_A^B \cap G^{\mathbb{R}}(2, n+1) \subset \mathcal{S}_A^B$
3. If the polytope \mathcal{M}_A^B is not defined, then the lines \mathcal{S}_A^B cannot be represented by only one convex polytope.

4.3.1 If the polytope \mathcal{M}_A^B exists, then it is minimal

Let \mathcal{M}_A^B be the polytope defined as the convex hull of the vertices $a_i \wedge b_j$, according to the theorem 4. Assuming this polytope represents lines \mathcal{S}_A^B stabbing \mathbf{A} and \mathbf{B} . Then, its vertices, *i.e.* the points in \mathfrak{L}_n associated to the lines $a_i \wedge b_j, \forall i \in [1 \dots q], j \in [1 \dots r]$, are in \mathcal{S}_A^B .

Let \mathcal{P} be a convex polytope strictly contained in \mathcal{M}_A^B . Obviously, any convex polytope containing all the vertices of \mathcal{M}_A^B , also contains their convex hull \mathcal{M}_A^B . Then, it follows that \mathcal{P} does not contain at least one of the vertices of \mathcal{M}_A^B . Since those vertices are in \mathcal{S}_A^B , we deduce that \mathcal{P} does not represent all the lines stabbing \mathbf{A} and \mathbf{B} , proving that \mathcal{M}_A^B is minimal.

4.3.2 Proof of $\mathcal{S}_A^B \subseteq \mathcal{M}_A^B$

Let us assume \mathcal{H}_A and \mathcal{H}_B do not intersect the inside of \mathbf{A} or \mathbf{B} respectively. The set \mathcal{S}_A^B in \mathfrak{G}_n contains lines defined by any couple of points on \mathbf{A} and \mathbf{B} , such that the point of \mathbf{A} is not on \mathcal{H}_B , and conversely the point of \mathbf{B} is not on \mathcal{H}_A .

Let $a \in \mathbf{A}$ and $b \in \mathbf{B}$ be two such points. Since \mathbf{A} and \mathbf{B} are convex, then the homogeneous representation of a and b in \mathfrak{G}_{n+1} can be represented by combinations of the vertices of \mathbf{A} and \mathbf{B} respectively. For instance¹:

$$a = \sum_{i=1}^q \alpha_i a_i, \quad \alpha_i \geq 0 \forall i \in [1 \dots q], \quad \text{and} \quad b = \sum_{j=1}^r \beta_j b_j, \quad \beta_j \geq 0 \forall j \in [1 \dots r]$$

¹ Using homogeneous coordinates, the sum of the coefficients does not need to be normalized to unity, as it is usually done in computational geometry.

So, the line $D = a \wedge b$ is:

$$\begin{aligned}
D &= a \wedge b \\
&= \left(\sum_{i=1}^q \alpha_i a_i \right) \wedge \left(\sum_{j=1}^r \beta_j b_j \right) \\
&= \sum_{i=1}^q \sum_{j=1}^r \alpha_i \beta_j a_i \wedge b_j \\
&= \sum_{i \in [1 \dots q], j \in [1 \dots r]} \gamma_{ij} a_i \wedge b_j
\end{aligned}$$

By hypothesis, since $\alpha_i \geq 0$ and $\beta_j \geq 0$, then $\gamma_{ij} = \alpha_i \beta_j \geq 0$. This shows that any line D in $\mathcal{S}_{\mathbf{A}}^{\mathbf{B}}$ is a convex combination of the points $a_i \wedge b_j$ in \mathfrak{L}_n , $\forall i \in [1 \dots q]$ and $j \in [1 \dots r]$. These points are precisely the vertices of $\mathcal{M}_{\mathbf{A}}^{\mathbf{B}}$. This proves that $\mathcal{S}_{\mathbf{A}}^{\mathbf{B}}$ is contained in $\mathcal{M}_{\mathbf{A}}^{\mathbf{B}}$.

4.3.3 Proof of $\mathcal{M}_{\mathbf{A}}^{\mathbf{B}} \cap G^{\mathbb{R}}(2, n+1) \subseteq \mathcal{S}_{\mathbf{A}}^{\mathbf{B}}$

By hypothesis, since $\mathcal{H}_{\mathbf{A}}$ (resp. $\mathcal{H}_{\mathbf{B}}$) does not split the inside of \mathbf{B} (resp. \mathbf{A}), all the vertices $b_j, \forall j \in [1 \dots r]$ (resp. $a_i, \forall i \in [1 \dots q]$) are in a same half-space delimited by $\mathcal{H}_{\mathbf{A}}$ (resp. $\mathcal{H}_{\mathbf{B}}$).

From this remark and Theorem 3, it can deduced that there is a unique orientation of the flats $f_i^a \forall i \in [1 \dots s]$ and $f_j^b \forall j \in [1 \dots t]$ verifying the following inequalities:

$$\begin{aligned}
a_i \wedge b_j \wedge f_k^a &> 0, \quad \forall (i, j, k) \in [1 \dots q] \times [1 \dots r] \times [1 \dots s] \\
a_i \wedge b_j \wedge f_l^b &> 0, \quad \forall (i, j, l) \in [1 \dots q] \times [1 \dots r] \times [1 \dots t]
\end{aligned}$$

Let $D = \sum_{i \in [1 \dots q], j \in [1 \dots r]} \gamma_{ij} a_i \wedge b_j$, $\gamma_{ij} \geq 0$ for all (i, j) in $[1 \dots q] \times [1 \dots r]$, be any point inside $\mathcal{M}_{\mathbf{A}}^{\mathbf{B}}$. It follows that:

$$\begin{aligned}
\forall k \in [1 \dots s], \quad D \wedge f_k^a &= \sum_{i \in [1 \dots q], j \in [1 \dots r]} \gamma_{ij} a_i \wedge b_j \wedge f_k^a \\
\forall l \in [1 \dots t], \quad D \wedge f_l^b &= \sum_{i \in [1 \dots q], j \in [1 \dots r]} \gamma_{ij} a_i \wedge b_j \wedge f_l^b
\end{aligned}$$

Since all γ_{ij} , $a_i \wedge b_j \wedge f_k^a$ and $a_i \wedge b_j \wedge f_l^b$ are positive scalars or pseudoscalars, then:

$$\begin{aligned}
D \wedge f_k^a &\geq 0, \quad \forall k \in [1 \dots s] \\
D \wedge f_l^b &\geq 0, \quad \forall l \in [1 \dots t]
\end{aligned}$$

Let us assume D is in the Grassmannian $G^{\mathbb{R}}(2, n+1)$ and $\mathcal{H}_{\mathbf{A}}$ and $\mathcal{H}_{\mathbf{B}}$. Then, from Theorem 3, D is in $\mathcal{S}_{\mathbf{A}}^{\mathbf{B}}$. From the hypothesis, D lies on one of the hyperplanes $\mathcal{H}_{\mathbf{A}}$ and $\mathcal{H}_{\mathbf{B}}$ if and only if it is incident to the $(n-2)$ -flat f_i , defined as the intersection of the two hyperplanes $\mathcal{H}_{\mathbf{A}}$ and $\mathcal{H}_{\mathbf{B}}$ in \mathfrak{G}_n . This $(n-2)$ -flat corresponds in \mathfrak{L}_n to a hyperplane f_i^* that bounds the polytope $\mathcal{M}_{\mathbf{A}}^{\mathbf{B}}$. Thus, the lines incident to f_i can be easily excluded from the polytope $\mathcal{M}_{\mathbf{A}}^{\mathbf{B}}$, by considering it *open* on the boundary corresponding to the hyperplane f_i^* .

Since the previous results are proved for any point in \mathcal{M}_A^B and on the Grassmannian $G^{\mathbb{R}}(2, n+1)$, we deduce: $\mathcal{M}_A^B \cap G^{\mathbb{R}}(2, n+1) \subseteq \mathcal{S}_A^B$.

4.3.4 When the hyperplane \mathcal{H}_B or \mathcal{H}_A intersects the inside of A or B

Assuming the hyperplane \mathcal{H}_B intersects the inside of **A**, there are at least two vertices a_{i_1} and a_{i_2} of **A** which are in the two opposite half-spaces delimited by \mathcal{H}_B (see Fig. 2).

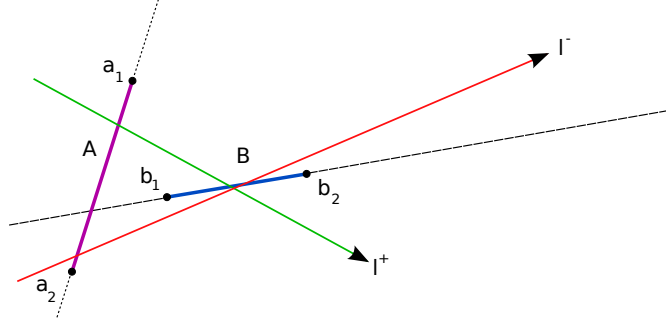


Fig. 2 Degenerate case in \mathfrak{G}_2 , where it is not possible to determine an orientation of the boundary of both the faces **A** and **B**, in order to characterize consistently all the lines stabbing the two polygons: the lines l^+ and l^- need opposite orientations.

Let b be a point of **B**, and let $D_1 = a_{i_1} \wedge b$ and $D_2 = a_{i_2} \wedge b$ be two lines in \mathcal{S}_A^B . Assuming the flats $f_i^a \forall i \in [1 \dots s]$ are correctly oriented, if $D_1 \wedge f_i^a$ are only positive pseudoscalars, for all $i \in [1 \dots s]$, then the $D_2 \wedge f_i^a$ are also positive, and conversely. This comes from the pseudoscalar's sign which only depends on which half-space delimited by \mathcal{H}_A the point b lies (see the proof of Theorem 3).

On the contrary, supposing the flats f_j^b for $j \in [1 \dots t]$ correctly oriented, since the vertices a_{i_1} and a_{i_2} do not lie on the same half-space, according to \mathcal{H}_B , and if $D_1 \wedge f_j^b$ is a positive pseudoscalar, then $D_2 \wedge f_j^b$ will be a negative pseudoscalar, and conversely.

Reversing the orientation of one of the two lines, for instance D_2 , is not a solution: The pseudoscalars $D_1 \wedge f_i^a$ and $-D_2 \wedge f_i^a$ still have an opposite sign.

This proves that it is not possible to point the flats f_i^a and $f_j^b, \forall i \in [1 \dots s]$ and $j \in [1 \dots t]$, such that the classification against those flats of all the lines stabbing both **A** and **B** only results in positive pseudoscalars. In other words, it is not possible to group together the lines \mathcal{S}_A^B in only one convex polytope. \square

4.3.5 Dealing with degenerate cases

In this paper, we talk about a degenerate case for two faces, when at least one of the two faces has an intersection with the hyperplane that extend the second face. From this definition, two different kinds of degenerate cases can be specified: Firstly when the intersection is limited to a boundary part of a face; Secondly, when the intersection also concern the inside of a face.

From Theorem 4, we know that there exists a minimal polytope representing the lines stabbing the two faces for the first kind of degeneracy, whereas there is not for the second one. However, this latter case can always be transformed in the former one, splitting the two faces along their intersection with the hyperplane supporting the opposite one. This split allows to divide the initial degenerate configuration in two or four configurations of the first type, depending on whether one or both faces are split.

5 An application example: Soft shadows computation

5.1 The n dimensional visibility framework implementation

As presented in previous sections, the set of lines intersecting two convex $n - 1$ faces A and B in \mathcal{G}_n can be represented as a $\binom{n+1}{2}$ -dimensional convex polytope P_{AB} in $\mathbb{P}^{\binom{n+1}{2}}$. Denoting by O_i , $1 \leq i \leq m$ the m occluding $(n - 1)$ -faces, the visibility between A and B is:

$$P_{AB} - \bigcup_{i=1}^m P_{O_i} = P_{AB} - \bigcup_{i=1}^m P_{AO_i} = P_{AB} - \bigcup_{i=1}^m P_{O_iB}$$

This can be computed using Computational Solid Geometry operations: Each polytope P_{AO_i} (or P_{O_iB}) has to be subtracted from P_{AB} . All n dimensional CSG operations can be implemented using Binary Space Partitioning trees [12]. The core of this method requires to split a n -dimensional convex polytope against a $(n - 1)$ -dimensional hyperplane. Two different approaches can be used:

1. An enumeration algorithm such as [1] can solve the linear system induced both by the splitting hyperplane and the bounding polytope hyperplanes (the so-called H-representation). However, such an approach is prone to numerical errors, especially in higher dimensions as noticed by Bittner [3] whose method relies on a similar algorithm.
2. Bajaj *et. al* [2] propose a more robust method relying on the relative position of a point and a hyperplane. Nirenstein [13] or Mora [11] use this algorithm. We also choose this technique because robustness is crucial in image synthesis. In particular, even a small error always leads to a blatant visual artifact.

As a result, this allows to implement the n -dimensional visibility framework whatever $n \geq 2$ is, contrary to previous works which are only correct in 2D or 3D space. In addition, our framework takes advantage of the minimal polytope theorem to optimize CSG computations, whereas previous works construct non-minimal polytopes, increasing the vertex number and thus the complexity of the CSG operations.

The visibility framework can be considered as a black box and easily plugged into any applications that need to perform visibility queries.

5.2 *Soft Shadow computations*

In computer graphics, soft shadows are very important to render realistic pictures, because they unveil the relative positions of the objects in the scene. But it is a difficult problem, since it requires to compute the visibility of an area light source from any point in the scene, which is very time consuming. In this section, we explain how the visibility framework can be used to solve exactly the visibility of an area light source and to speed-up the computation.

5.2.1 Application overview

We consider a 3D environment made of convex polygons and precompute their visibility with an area light source L . Denoting T a polygon in the scene, this leads to compute for each pair (L, T) a 6D BSP tree whose inner nodes are 5D projective hyperplanes corresponding to the duals of occluders' edges, and whose leaves are polytopes representing a visible or invisible set of lines. Such a tree is an exact and coherent representation of the visibility of L from any point on T . As a consequence it is used during the rendering step to query the visibility of L for each point on T visible from the camera. A simple algorithm to perform such a query is presented in [10]. It provides an exact polygonal subdivision of the visible parts of L from a given point. This result is then used to compute the direct illumination received by the point.

We compare our approach to the solution commonly used in production rendering software: A stratified sampling of the area light source. In this case, the visibility of L from a given point is evaluated by shooting shadow rays towards each sample on L . The quality of the result increases with the number of samples as well as the computation time.

5.2.2 Results

All tests are run on an Intel Core 2 Duo at 2.4 GHz with 3 Gbyte of memory. For comparison purposes, all pictures are rendered at 800x600 on one thread without anti-aliasing. The comparison method uses 256 samples per area light source, since

	Polygons	BSP-tree	Ptime	Rtime	Ctime
Eagle	5 520	19 154	9min 34s	2.9s	1min 24s
Panther	12 993	47 684	54min 12s	4.1s	1min 45s

Fig. 3 Result details for the two test scenes, *Eagle* and *Panther*, with one area light source. The first column gives the number of polygons in a scene. The second column presents the total number of inner nodes for all precomputed BSP trees. The third column indicates the time spent for precomputing all BSP trees. The fourth column gives the time spent in soft shadows computation using our framework, whereas the last columns gives this time using the comparison method.

this number is usually considered sufficient for producing quality results. The ray tracer is an implementation of [20], taking advantage of SSE instructions to trace four rays at a time.

Fig. 4 presents the pictures. The first scene, *Eagle*, is a model with a moderate shadow complexity, while the second scene, *Panther*, is a more complex case. Despite the high number of samples used by the comparison method, noise remains in soft shadows as illustrated by the close-ups. Using our visibility framework, the soft shadows quality is optimal, whatever the zooming is, since the visibility queries are exact. It is worth underlying that we were not able to precompute correctly the visibility on the *Panther* scene using a non-minimal polytope like in [13] or [10]: Because of numerical instabilities, errors occur in the visibility data, leading to visual artifacts in soft shadows. Using the minimal polytope, we avoid to perform useless CSG operations, improving robustness.

Table 3 presents the computation details. The size of the BSP-trees illustrates their compactness and ability to efficiently encode the visibility data. The precomputation times are significant since CSG operations in high dimensions are time consuming. However the method remains practicable and it does not depend on the point of view. As a consequence it can be computed once then stored into files to be reused later. Finally, the time spent in soft shadows computation during the rendering step clearly shows the efficiency of the visibility framework. Indeed, the visibility queries used on the BSP trees depend on their average depth and compactness. Thus, the benefit from the precomputation step is really important.

In this application, our visibility framework manages to reconcile accuracy and efficiency, often considered as two opposite qualities in computer graphics.

References

1. David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Appl. Math.*, 65:21–46, March 1996.
2. C.L. Bajaj and V. Pascucci. Splitting a complex of convex polytopes in any dimension. In *Proceedings of the Twelfth Annual Symposium On Computational Geometry (ISG '96)*, pages 88–97, New York, May 1996. ACM Press.
3. Jirí Bittner. *Hierarchical Techniques for Visibility Computation*. PhD thesis, Department of Computer Science and Engineering, Czech Technical University in Prague, 2002.

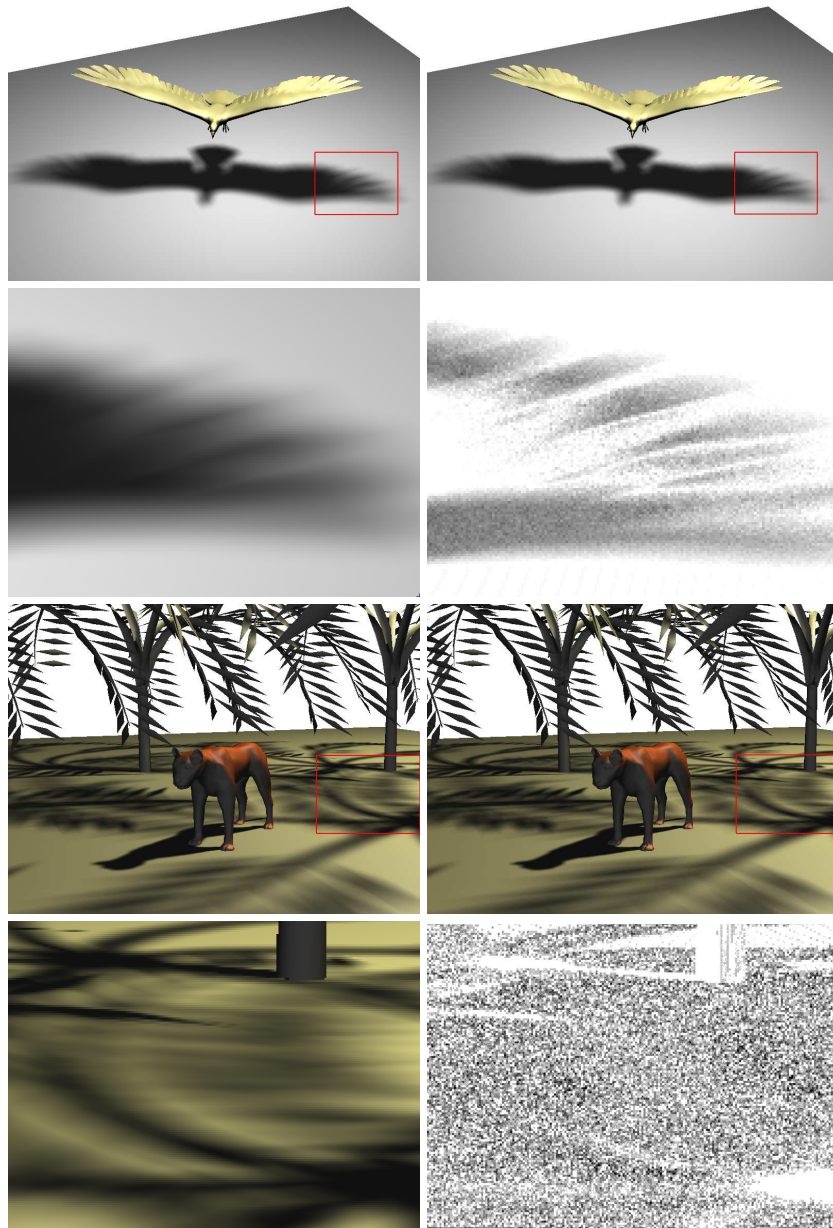


Fig. 4 The left column presents the pictures with soft shadows computed using our exact visibility framework. The right column presents the differences' images of the same pictures computed using a classical sampling strategy. The close-ups underline the significant differences in soft shadows.

4. Jiri Bittner, Jan Prikryl, and Pavel Slavík. Exact regional visibility using line space partitioning. *Computers & Graphics*, 27(4):569–580, 2003.
5. 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.
6. 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.
7. Frédo Durand, George Drettakis, and Claude Puech. The 3d visibility complex. *ACM Trans. Graph.*, 21:176–206, April 2002.
8. Denis Haumont, Otso Mäkinen, and Shaun Nirenstein. A Low Dimensional Framework for Exact Polygon-to-Polygon Occlusion Queries. In *Rendering Techniques*, pages 211–222, 2005.
9. Tomas Möller and Ben Trumbore. Fast, Minimum Storage Ray-Triangle Intersection. *Journal of Graphics, GPU, and Game Tools*, 2(1):21–28, 1997.
10. Frédéric Mora and Lilian Aveneau. Fast and exact direct illumination. In *Proceedings of the Computer Graphics International 2005*, pages 191–197, 2005.
11. Frédéric Mora, Lilian Aveneau, and Michel Mériaux. Coherent and Exact Polygon-to-Polygon Visibility. In WSCG'05, pages 87–94, 2005.
12. Bruce F. Naylor, John Amanatides, and William C. Thibault. Merging BSP trees yields polyhedral set operations. In *SIGGRAPH*, pages 115–124, 1990.
13. Shaun Nirenstein, Edwin H. Blake, and James E. Gain. Exact From-Region Visibility Culling. In *Rendering Techniques*, pages 191–202, 2002.
14. Rachel Orti, Frédo Durand, Stéphane Rivière, and Claude Puech. Using the Visibility Complex for Radiosity Computation. In *WACG*, pages 177–190, 1996.
15. 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.
16. Michel Pocchiola and Gert Vegter. The Visibility Complex. *Int. J. Comput. Geometry Appl.*, 6(3):279–308, 1996.
17. J. Stolfi. *Oriented Projective Geometry: A Framework for Geometric Computations*. Academic Press, 1991.
18. 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.
19. 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.
20. Ingo Wald, Philipp Slusallek, Carsten Benthin, and Markus Wagner. Interactive rendering with coherent ray tracing. *Comput. Graph. Forum*, 20(3):153–164, 2001.

6 Exercises

1. Prove that Plücker's coordinates correspond to the coordinates of a bivector in $\wedge^2(\mathbb{R}^4)$.
2. From the parametric equation of a line (*i.e.* $P * t + Q * (1 - t)$ where P and Q are n -dimensional points) find one of its representing bivector.
3. The Plücker relation between two lines expressed with their six Plücker coordinates (Π_0, \dots, Π_5) and $(\Delta_0, \dots, \Delta_5)$ is: $\Pi_0\Delta_3 + \Pi_1\Delta_4 + \Pi_2\Delta_5 + \Pi_3\Delta_0 + \Pi_4\Delta_1 + \Pi_5\Delta_2$. Show that it is equivalent to the inner product in \mathfrak{L}_3 .

4. Let $A, B,$ and C three euclidean points in \mathfrak{G}_3 , with respective coordinates $(1, 0, 0), (2, 1, 1)$ and $(1, 0, 2)$. Let F be the triangle (A, B, C) . Let $P, Q,$ and R three euclidean points in \mathfrak{G}_3 , with respective coordinates $(0, 1, 1), (2, 0, 2)$ and $(4, 0, 4)$. Are the lines (PQ) and (RP) stab the face F ? Same question for the lines (PR) and (QP) , but without any new computations.
5. Find a bivector that is not decomposable. Show that this bivector cannot represent a line into the space of the geometric objects. (hint: consider dimension 4).
6. Prove Theorem 2. Notice that the duality cannot be expressed easily directly in $\wedge(\mathbb{R}^{n+1})$. The left contraction allows to express it, so the difficulty only resides in dimension n .
7. Consider two faces A and B , show that any line that is outside the minimal polytope cannot cross A and B .