Least Square for Grassmann-Cayley Agelbra in Homogeneous Coordinates

Vincent Lesueur, Vincent Nozick

Least Square for Grassmann-Cayley Agelbra in Homogeneous Coordinates

Vincent Lesueur and Vincent Nozick

Gaspard Monge Institute, UMR 8049
Université Paris-Est Marne-la-Vallée, France

Abstract. This paper presents some tools for least square computation in Grassmann-Cayley algebra, more specifically for elements expressed in homogeneous coordinates. We show that building objects with the outer product from k-vectors of same grade presents some properties that can be expressed in term of linear algebra and can be treated as a least square problem. This paper mainly focuses on line and plane fitting and intersections computation, largely used in computer vision. We show that these least square problems written in Grassmann-Cayley algebra have a direct reformulation in linear algebra, corresponding to their standard expression in projective geometry and hence can be solved using standard least square tools.

Keywords: Grassmann-Cayley algebra, least square, line fitting, plane fitting, intersection.

1 Introduction

Grassmann-Cayley algebra presents some powerful operators that have direct applications to computer vision and computer graphics. In computer vision side, problems are often treated in a 3-steps approach consisting in data measurement, least square estimation of the considered model and an non-linear refinement. There already exist many geometric models expressed in Grassmann-Cayley algebra, from elementary such as lines or planes to more complex. However least square estimation has not been fully investigated. This paper presents an extension of both outer and regressive products, designed to adapt the grade of an element being computed to the required grade and not to the expected one. The proposed method is based on a least square approximation of elements of same grade wedged together. This paper mainly focuses on line and plane fitting as well as lines and planes intersection.

1.1 Brief Overview of Grassmann-Cayley Algebra

This section gives a very brief introduction to Grassmann algebra. The central operation in Grassmann algebra is the outer product (or wedge product) of two vectors $\mathbf{a}$ and $\mathbf{b}$. This product, written $\mathbf{a} \wedge \mathbf{b}$, denotes the oriented surface defined

by the two vectors. This oriented property involves the anticommutativity of the wedge product:

$$\mathbf{a} \wedge \mathbf{b} = -\mathbf{b} \wedge \mathbf{a}$$

and thus, the property:

$$\mathbf{a} \wedge \mathbf{a} = 0$$

Moreover, the wedge product is distributive such that $\mathbf{a} \wedge (\mathbf{b}+\mathbf{c}) = (\mathbf{a} \wedge \mathbf{b}) + (\mathbf{a} \wedge \mathbf{c})$ and associative. The wedge operation between two vectors generates a bivector. Wedging three vectors makes a trivector, and so on. In a space of dimension n, the grade of a k-vector is the number $k \leq n$ of vectors wedged together to build the k-vector. Thus, for a p-vector $\mathbf{a}$ and a q-vector $\mathbf{b}$, we have:

$$\text{grade}(\mathbf{a} \wedge \mathbf{b}) = \text{grade}(\mathbf{a}) + \text{grade}(\mathbf{b}) = p + q$$

Each component of a n-dimensional vector is expressed by the unit vectors $\{\mathbf{e_1}, \mathbf{e_2}, \cdots, \mathbf{e_n}\}$. The components of a k-vector are expressed by unit k-vectors that are composed of $k \leq n$ unit vectors wedged together and abbreviated $\mathbf{e_i} \wedge \cdots \wedge \mathbf{e_j} = \mathbf{e_{i \cdots j}}$, e.g. each component of a bivectors of $\mathbb{R}^3$ is expressed with the set of unit bivectors $\{\mathbf{e_{12}}, \mathbf{e_{13}}, \mathbf{e_{23}}\}$.

Grassmann-Cayley algebra can also express a k-vector in a dual basis containing all the unit $(n-k)$-vectors of $\mathbb{R}^n$ that are not used to compose the k-vector (e.g. $\bar{\mathbf{e}}_\mathbf{1} = \mathbf{e_{234}}$ in 4 dimensions). Grassmann-Cayley algebra includes an operator called regressive product (or anti-wedge product) that operates on dual k-vectors in a manner symmetric to how the wedge product operates on k-vectors. This operator, written $\mathbf{a} \vee \mathbf{b}$, affects the grade such that:

$$\text{grade}(\mathbf{a} \vee \mathbf{b}) = \text{grade}(\mathbf{a}) + \text{grade}(\mathbf{b}) - n$$

For more details about Grassmann-Cayley algebra, the reader should refer to Doubliet et al. [1] and Barnabei et al. [2].

1.2 Grassmann-Cayley Algebra and Homogeneous Coordinates

Grassmann-Cayley algebra presents some interesting properties when applied in projective space, where vectors are expressed in homogeneous coordinates. Some of these properties have direct applications in computer vision and computer graphics, especially those related to the computation of lines and planes detailed in Carlsson [3] and summarized in Table 1. Some additional relationships between points, lines and planes are summarized in Table 2. All these properties as well as others relationships such as line parallelism, line orthogonality or identity between elements are presented in Förstner et al. [4].

Table 1. Wedge and anti-wedge geometric application on $\mathbb{P}^2$ and $\mathbb{P}^3$.

Grassmann-Cayley algebra in $\mathbb{P}^2$	
point $\wedge$ point $=$ line	(passing through the two points)
line $\vee$ line $=$ point	(intersection of the two lines)

Grassmann-Cayley algebra in $\mathbb{P}^3$	
point $\wedge$ point $=$ line	(passing through the two points)
point $\wedge$ point $\wedge$ point $=$ plane	(containing the three points)
point $\wedge$ line $=$ plane	(containing the line and the point)
plane $\vee$ line $=$ point	(intersection of the plane and the line)
plane $\vee$ plane $\vee$ plane $=$ point	(intersection of the three planes)
plane $\vee$ plane $=$ line	(intersection of the two planes)

Table 2. Relationship between points, lines and planes in $\mathbb{P}^2$ and $\mathbb{P}^3$.

Grassmann-Cayley algebra in $\mathbb{P}^2$			
point $\in$ line	point $\wedge$ line $= 0$	$\leftrightarrow$	point $\vee$ line $= 0$

Grassmann-Cayley algebra in $\mathbb{P}^3$			
point $\in$ line	point $\wedge$ line $= 0$	$\leftrightarrow$	point $\vee$ line $= 0$
point $\in$ plane	point $\wedge$ plane $= 0$	$\leftrightarrow$	point $\vee$ plane $= 0$

2 Least Square in Grassmann-Cayley Algebra

2.1 Grade and Least Square

A specific use of the outer product is to wedge a set of p-vectors (with same grade p) to build a k-vector $\mathbf{u}$:

$$\underset{k}{\mathbf{u}} = \underset{p}{\mathbf{x}_1} \wedge \underset{p}{\mathbf{x}_2} \wedge \cdots \wedge \underset{p}{\mathbf{x}_r} \tag{1}$$

with $\text{grade}(\mathbf{u}) = k = rp$.

The problem addressed in this paper is how to extend the number of p-vectors wedged together without changing the grade of $\mathbf{u}$:

$$\underset{k}{\mathbf{u}} \leftarrow \underset{p}{\mathbf{x}_1} \wedge \underset{p}{\mathbf{x}_2} \wedge \cdots \wedge \underset{p}{\mathbf{x}_r} \wedge \cdots \wedge \underset{p}{\mathbf{x}_s} \tag{2}$$

This problem can be seen as a least square problem where $\mathbf{u}$ is still of grade $k = rp$ and each $\mathbf{x}_i$ $(i = 1, ..., r, ..., s)$ contributes to build $\mathbf{u}$. This operation can be written in a more compact expression:

$$\mathbf{u} = \wedge^k_{\{\mathbf{x}_i\}_1^s} \tag{3}$$

where $k \leq sp$ is the expected grade of $\mathbf{u}$ and $\{\mathbf{x}_i\}$ the set of p-vector to wedge. It should be noted that the elements of $\{\mathbf{x}_i\}$ are not ordered and hence this operator looses the oriented property of the wedge product. While this operator may not be of big interest in $\bigwedge^k \mathbb{R}^n$ (the linear space of k-vectors of $\mathbb{R}^n$), it involves interesting properties in $\bigwedge^k \mathbb{P}^n$ where each k-vector is expressed in homogeneous coordinates. Indeed, as suggested by Table 1, there exist direct applications of this problem in computer vision, such as line fitting, line intersection, etc.

Some least square applications in geometric algebra has been investigated by Gebken et al. [5], applied to line and circle fitting in the conformal space of Euclidean 3D-space, but to our knowledge, nothing has been done concerning the least square formulation of the wedge product.

2.2 Linear Algebra Reformulation

The construction of a k-vector $\mathbf{u}$ as formulated in equation (1) involves:

$$\mathbf{u} \wedge \mathbf{x}_i = \mathbf{0} \qquad \forall i \in 1, ..., r \tag{4}$$

Each element $\mathbf{u} \wedge \mathbf{x}_i$ is a $(k+p)$-vector whose components are all zero. Due to the distributivity of the wedge product, the analytic form of the j^{th} component of the i^{th} relation $\mathbf{u} \wedge \mathbf{x}_i$ can be formulated as the dot product of $\mathbb{R}^{C_n^k}$ between $\mathbf{u}$ and a k-vector $\mathbf{a}_i^j$ such that equation (4) is equivalent to:

$$\mathbf{a}_i^{j\top} \cdot \mathbf{u} = 0 \qquad \forall i \in 1, ..., r \;\text{ and }\; \forall j \in 1, ..., C_n^{k+p} \tag{5}$$

These C_n^{k+p} equations have the following matrix form:

$$\underbrace{\begin{bmatrix} \mathbf{a}_1^{1\top} \\ \mathbf{a}_1^{2\top} \\ \vdots \\ \mathbf{a}_1^{C_n^{k+p}\top} \end{bmatrix}}_{\mathtt{A}_i} \mathbf{u} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{6}$$

Combining all the elements of $\{\mathbf{x}_i\}$ together leads to:

$$\underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_r \end{bmatrix}}_{\mathtt{A}} \mathbf{u} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{7}$$

Hence the k-vector $\mathbf{u}$ is the right-nullspace of a matrix $\mathtt{A}$ of size $\left(r.C_n^{k+p} \times C_n^k\right)$ build exclusively with the set $\{\mathbf{x}_i\}$.

2.3 Least Square and Wedge Product

The extension of equation (7) to the overdetermined case presented on equation (2) leads to a similar formulation where we want to minimize the residual $\|\mathtt{A}\mathbf{u}\|_2$ in a linear least square sense. This residual corresponds to the L_2 norm of the vector composed by all the components of all vectors $\mathbf{u} \wedge \mathbf{x}_i$. Again, the k-vector $\mathbf{u}$ is the right-nullspace of the matrix $\mathtt{A}$ and can be numerically computed with a Singular Value Decomposition subject to $\|\mathbf{u}\|_2 = 1$, the result is the right-singular vector associated to the smallest singular value of $\mathtt{A}$. The constraint $\|\mathbf{u}\|_2 = 1$ is not a problem in $\bigwedge^k \mathbb{P}^n$ where each element is defined up to a non-zero scale factor, but would be a problem in $\bigwedge^k \mathbb{R}^n$. The form of the matrix $\mathtt{A}$ will be detailed from section 3. Moreover, the size of $\mathtt{A}$ becomes $\left(s.C_n^{k+p} \times C_n^k\right)$, however we will see that in lines and planes case, C_n^{k+p} is often equal to 1.

3 Line, Plane and Hyper-Plane Fitting

The least square formulation of the wedge product presented in Section 2.3 has a direct application to line fitting in $\mathbb{P}^2$, plane fitting in $\mathbb{P}^3$ and n-dimensional hyperplane fitting in $\mathbb{P}^n$. All these problems are solved following the same approach presented on the next parts.

3.1 Line Fitting in $\mathbb{P}^2$

Let $\{\mathbf{x}_i\}_{i=1,\dots,m}$ a set of m points of $\mathbb{P}^2$ of the form $\mathbf{x}_i = x_i\mathbf{e_1} + y_i\mathbf{e_2} + w_i\mathbf{e_3}$. The line $\mathbf{l}$ fitting these points is a 2-vector and hence is formulated:

$$\mathbf{l} = l_1\mathbf{e_{12}} + l_2\mathbf{e_{13}} + l_3\mathbf{e_{23}}$$

The distance to minimize between a point $\mathbf{x}_i$ and the line $\mathbf{l}$ is represented by:

$$\mathbf{l} \wedge \mathbf{x}_i = (w_i l_1 - y_i l_2 + x_i l_3)\ \mathbf{e_{123}} \tag{8}$$

Hence, the least square constraint applied on $\mathbf{l}$ is:

$$w_i l_1 - y_i l_2 + x_i l_3 = 0 \tag{9}$$

Thus, as specified by equation (5), the relation $\mathbf{x}_i \wedge \mathbf{l} = 0$ can be expressed in term of a set of dot products $\mathbf{a}_i^\top \cdot \mathbf{l} = 0$ with $\mathbf{a}_i = (w_i, -y_i, x_i)^\top$. Here, only a single dot product is required. The associated system to solve is:

$$\underbrace{\begin{bmatrix} w_1 & -y_1 & x_1 \\ w_2 & -y_2 & x_2 \\ \vdots & \vdots & \vdots \\ w_m & -y_m & x_m \end{bmatrix}}_{\mathtt{A}} \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

It is noteworthy to see that equation (8) corresponds to the perpendicular Euclidean distance between $\mathbf{x}_i$ and $\mathbf{l}$ under 2 conditions, namely the line should be expressed in its normal Hessian form, i.e. $\sqrt{l_2^2 + l_3^2} = 1$ and the finite point should be expressed as $(x_i/w_i, y_i/w_i, 1)^\top$. Numerically, the first constraint is not satisfied since the SVD involves $\|\mathbf{l}\|_2 = 1$, however this constraint is applied to all points $\mathbf{x}_i$ and hence is just a common scale factor to the computed distances to minimize. Moreover, the SVD minimizes $\|\mathtt{A}\mathbf{l}\|_2$, i.e. the square root of a sum of squared distances, that has the same minimum as the sum of squared distances which is a standard least square residual.

It is also interesting to see that equation (9) is the homogeneous form of a line in $\mathbb{P}^2$ (up to the order and sign of the components of $\mathbf{l}$) commonly used in computer vision. A usual precaution is a pre-conditioning of the matrix $\mathtt{A}$ before the numerical computation of its right nullspace. Indeed, the first column of $\mathtt{A}$ is composed of $w_i = 1$ and the two last columns contain x_i and y_i that can be much larger than w_i, resulting in non-negligible numerical instability. Hartley and Zisserman [6, p. 107] suggest to perform a data normalization consisting in a translation of the points so that their centroid is at the origin, followed by scale so that the average distance from the origin is $\sqrt{2}$, leading to an "average" point $= (1, 1, 1)^\top$. This operation can be performed by the following normalization matrix $\mathtt{T}$:

$$\mathtt{T} = \begin{bmatrix} 1 & \frac{\overline{y}\sqrt{2}}{d} & -\frac{\overline{x}\sqrt{2}}{d} \\ 0 & -\frac{\sqrt{2}}{d} & 0 \\ 0 & 0 & \frac{\sqrt{2}}{d} \end{bmatrix}$$

where $(\overline{x}, \overline{y}, 1)^\top$ is the centroid coordinates of the data and d is the average distance of the points from the centroid:

$$d = \frac{1}{m} \sum_{i=1}^{m} \sqrt{(x_i - \overline{x})^2 + (y_i - \overline{y})^2}$$

The normalized system to solve is $\mathtt{A}\mathtt{T}\mathtt{T}^{-1}\mathbf{l} = \mathbf{0}$. Thus, the normalized data $\widehat{\mathtt{A}}$ is obtained by $\widehat{\mathtt{A}} = \mathtt{A}\mathtt{T}$ and the system to solve becomes $\widehat{\mathtt{A}}\,\widehat{\mathbf{l}} = \mathbf{0}$. Finally, the line coefficients are given by $\mathbf{l} = \mathtt{T}\widehat{\mathbf{l}}$. Any other conditioning methods also may succeed to provide correct results. As depicted in Figure 1, the data normalization is not optional.

3.2 Plane Fitting in $\mathbb{P}^3$

Plane fitting from points in $\mathbb{P}^3$ is very similar to line fitting in $\mathbb{P}^2$. The set $\{\mathbf{x}_i\}_{i=1,\dots,m}$ is composed of points $\mathbf{x}_i \in \mathbb{P}^3$ with $\mathbf{x}_i = x_i\mathbf{e_1} + y_i\mathbf{e_2} + y_i\mathbf{e_3} + w_i\mathbf{e_4}$. The plane π is a 3-vector of the form:

$$\pi = \pi_1\mathbf{e_{123}} + \pi_2\mathbf{e_{124}} + \pi_3\mathbf{e_{134}} + \pi_4\mathbf{e_{234}}$$

The least square constraint on the $\mathbf{x}_i$ is:

$$\pi \wedge \mathbf{x}_i = (\pi_1 w_i - \pi_2 z_i + \pi_3 y_i - \pi_4 x_i)\ \mathbf{e_{1234}} = 0\ \mathbf{e_{1234}}$$

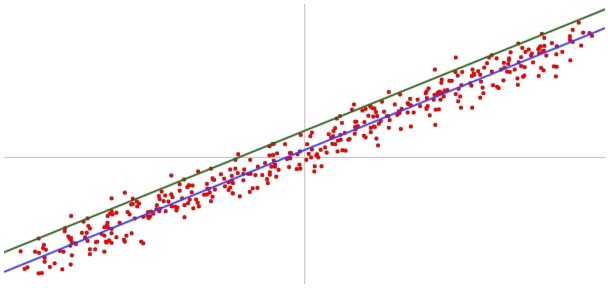

Fig. 1. Line fitting in $\mathbb{P}^2$ on a set of 2d points with Gaussian noise. Blue line: the fitting line with data normalization. Green line: without normalization. This is a critical situation where the range of the point distribution is very large compared to the distance between the line and the origin.

Hence, the system to solve is:

$$\begin{bmatrix} w_1 & -z_1 & y_1 & -x_1 \\ w_2 & -z_2 & y_2 & -x_2 \\ \vdots & \vdots & \vdots & \vdots \\ w_m & -z_m & y_m & -x_m \end{bmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Again, this is the Hessian form of a plan in $\mathbb{P}^3$ and its computation involves data normalization before solving the system. The normalization of [6] should be performed with a scaling of $\sqrt{3}$ on x,y and z. Any other conditioning methods also may succeed to provide correct results.

3.3 General Form in $\mathbb{P}^n$

The generalization of line fitting in $\mathbb{P}^2$ and plane fitting in $\mathbb{P}^3$ is n-dimensional hyperplane fitting in $\mathbb{P}^n$. Let $\{\mathbf{x}_i\}_{i=1,\dots,m}$ be a set of m points of $\mathbb{P}^n$ with components $\mathbf{x}_i = \sum_{j=1}^{n+1} x_{i,j}\mathbf{e_j}$. Given an hyperplane $\mathbf{h} \in \bigwedge^n \mathbb{P}^n$, each point $\mathbf{x}_i$ should satisfy $\mathbf{h} \wedge \mathbf{x}_i = \sum_{j=1}^{n+1} (-1)^{j+1} h_j x_{i,n+2-j}\ \mathbf{e_{123\cdots n+1}} = 0\ \mathbf{e_{123\cdots n+1}}$, which is also a single dot product. The matrix form of the corresponding system is $\mathtt{A}\mathbf{h} = \mathbf{0}$ with:

$$\mathtt{A} = \begin{bmatrix} x_{1,n+1} & -x_{1,n} & \cdots & (-1)^{j+1}x_{1,j} & \cdots & (-1)^n x_{1,1} \\ x_{2,n+1} & -x_{2,n} & \cdots & (-1)^{j+1}x_{2,j} & \cdots & (-1)^n x_{2,1} \\ x_{3,n+1} & -x_{3,n} & \cdots & (-1)^{j+1}x_{3,j} & \cdots & (-1)^n x_{3,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m,n+1} & -x_{m,n} & \cdots & (-1)^{j+1}x_{m,j} & \cdots & (-1)^n x_{m,1} \end{bmatrix}$$

Data normalization presented in [6] should include a scaling of $\sqrt{n}$ on the component $x_{i,j}$, with $j \in 1, ..., n$ with the following normalization matrix $\mathtt{T}$:

$$\mathtt{T} = \frac{\sqrt{n}}{d} \begin{bmatrix} \frac{d}{\sqrt{n}} & \overline{x_n} & \cdots & (-1)^{n-j}\overline{x}_j & \cdots & (-1)^{n-1}\overline{x}_1 \\ 0 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & (-1)^{n-j+1} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & (-1)^n \end{bmatrix}$$

4 Line Fitting in $\mathbb{P}^3$

Line fitting in $\mathbb{P}^3$ slightly differs from line fitting in $\mathbb{P}^2$. Indeed, a line in $\mathbb{P}^3$ is a 2-vector with $C_4^2 = 6$ components:

$$\mathbf{l} = l_1\mathbf{e_{12}} + l_2\mathbf{e_{13}} + l_3\mathbf{e_{14}} + l_4\mathbf{e_{23}} + l_5\mathbf{e_{24}} + l_6\mathbf{e_{34}}$$

These 6 components correspond to the Plücker coordinates of a line of $\mathbb{P}^3$ in the following order:

$$L = \{\mathbf{u} : \mathbf{v}\} = \{(l_3, l_5, l_6)^\top : (-l_4, l_2, -l_1)^\top\}$$

where $\mathbf{u}$ and $\mathbf{v}$ should satisfy $\mathbf{u}^\top \cdot \mathbf{v} = \mathbf{0}$ and $\mathbf{u} \neq \mathbf{0}$. The vector $\mathbf{u}$ is the tangent vector of the homogeneous line and the vector $\mathbf{v}$ is its moment. In Euclidean space, the line can be defined as the set of points $\mathbf{p} = \mathbf{m} + \alpha\mathbf{u}$ where $\mathbf{m} = (\mathbf{v} \times \mathbf{u}, \mathbf{u}^\top.\mathbf{u})^\top$ is the nearest point of $\mathbf{l}$ from the origin of the referential frame. Moreover, the perpendicular Euclidean distance between the line $\mathbf{l}$ and a 3D point $\mathbf{x} = (x, y, z)^\top$ is $\text{dist}(\mathbf{x}, \mathbf{l}) = \|(\mathbf{x} \times \mathbf{u} - \mathbf{v}, \mathbf{u}^\top.\mathbf{u})^\top\|_2$.

A point $\mathbf{x} = x\mathbf{e_1} + y\mathbf{e_2} + z\mathbf{e_3} + w\mathbf{e_4}$ lies on the line $\mathbf{l}$ if $\mathbf{x} \wedge \mathbf{l} = \mathbf{0}$. This expression can be developed as:

$$\begin{aligned} \mathbf{x} \wedge \mathbf{l} = &(xl_4 - yl_2 + zl_1)\mathbf{e_{123}} \\ &+ (xl_5 - yl_3 + wl_1)\mathbf{e_{124}} \\ &+ (xl_6 - zl_3 + wl_2)\mathbf{e_{134}} \\ &+ (yl_6 - zl_5 + wl_4)\mathbf{e_{234}} \\ &= \mathbf{0} \end{aligned}$$

As presented in equation (6), this expression can be formulated in a matrix form:

$$\begin{bmatrix} z & -y & 0 & x & 0 & 0 \\ w & 0 & -y & 0 & x & 0 \\ 0 & w & -z & 0 & 0 & x \\ 0 & 0 & 0 & w & -z & y \end{bmatrix} \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{10}$$

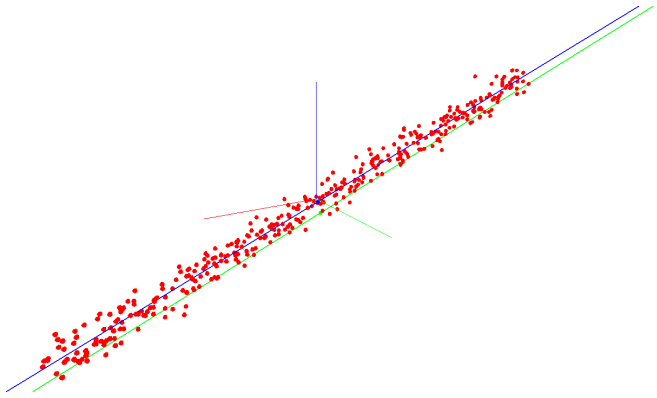

Fig. 2. Line fitting in $\mathbb{P}^3$ on a set of 3d points with Gaussian noise. Blue line: the fitting line with data normalization. Green line: without normalization. This is a critical situation where the range of the point distribution is very large compared to the distance between the line and the origin.

The three last lines of equation (10) correspond to the three components of the vector whose norm is $\text{dist}(\mathbf{x}, \mathbf{l})$. The first line does not have any geometric meaning (in a Plücker sense) and can be ignored. Hence, a line fitting a set of m points $\{\mathbf{x}_i\}_{i=1,\dots,m}$ can be computed with the following system:

$$\begin{bmatrix} w_1 & 0 & -y_1 & 0 & x_1 & 0 \\ 0 & w_1 & -z_1 & 0 & 0 & x_1 \\ 0 & 0 & 0 & w_1 & -z_1 & y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_m & 0 & -y_m & 0 & x_m & 0 \\ 0 & w_m & -z_m & 0 & 0 & x_m \\ 0 & 0 & 0 & w_m & -z_m & y_m \end{bmatrix} \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{11}$$

Each point is expressed as $(x_i/w_i, y_i/w_i, z_i/w_i, 1)^\top$. Numerically, equation (11) minimizes the square root of a sum of squared distances that has the same minimum as the sum of squared distances which is a standard least square residual. This system is solved subject to $\|\mathbf{l}\|_2 = 1$ that is not a normalized Plücker line, hence all the distances are computed up to a common scale factor however the final normalized result is not altered.

Data normalization is performed in two steps. First, translate the input points of $\mathbf{t} = (-\overline{x}, -\overline{y}, -\overline{z})^\top$ so that their centroid is at the origin. Next, scale the data with an isotropic scale s so that the average distance between the points and the origin is $\sqrt{3}$. Then, compute the right nullspace of the matrix of equation (11) build from the normalized data. Finally, apply the inverse transformation to the

line $\mathbf{l}$. In Plücker coordinates, this inverse transformation is formulated as:

$$\begin{aligned} &\text{Result of equation (11) :} && \widehat{L} = \{\mathbf{u} : \mathbf{v}\} \\ &\text{Unormalized result :} && L = \left\{\mathbf{u} : \frac{\mathbf{v}}{s} - \mathbf{t} \times \frac{\mathbf{u}}{\|\mathbf{u}\|_2}\right\} \end{aligned}$$

As shown in Figure 2, data normalization is not optional.

5 Intersections

This section presents the computation of intersection between lines in $\mathbb{P}^2$ and $\mathbb{P}^3$, planes in $\mathbb{P}^3$ and hyperplanes in $\mathbb{P}^n$. Intersection computation is the dual problem of data fitting presented in section 3 and 4 and hence introduces the dual operator of equation (3), that is $\mathbf{u} = \vee_{\{\mathbf{x}_i\}_1^s}^k$. However line intersection is known to be difficult since the cost function under L_2-norm related to the least square method is not convex. The following methods will provide a good estimation of the intersection, but for an optimal solution, the reader should refer to methods such as Lu and Hartley [7] who introduce a practical method to solve this problem with the L_2-norm. Kanatani et al. [8] propose an alternative method that can be applied to both line fitting and line intersection problems. Dorst et al. [9, equation 11.20] as well as Valkenburg and Alwesh in [10, chapter 7] present a closed form of this problem using geometric algebra.

5.1 Intersection of Lines in $\mathbb{P}^2$, Planes in $\mathbb{P}^3$ and n-Dimensional Hyperplane in $\mathbb{P}^n$

This section is a generalization of the least square intersection of lines in $\mathbb{P}^2$ and planes in $\mathbb{P}^3$ to n-dimensional hyperplane in $\mathbb{P}^n$. Let $\{\pi_i\}_{i=1,\dots,m}$ be a set of m hyperplanes of $\mathbb{P}^n$ such that $\pi_i = \sum_{j=1}^n \pi_{i,j}\bar{\mathbf{e}}_{\mathbf{n-j+1}}$. The intersection of these hyperplanes in a least square sense is the point $\mathbf{x} = \sum_{i=1}^{n+1} x_i$. The least square constraint on the system is:

$$\pi_{\mathbf{i}} \vee \mathbf{x} = \sum_{i=1}^{n+1} (-1)^{i,j+1} \pi_{n+2-j} x_j \; \bar{\mathbf{e}}_{\mathbf{123\cdots n+1}} = 0 \; \bar{\mathbf{e}}_{\mathbf{123\cdots n+1}}$$

The matrix form of the corresponding system is:

$$\begin{bmatrix} -\pi_{1,n+1} & \pi_{1,n} & \cdots & (-1)^j \pi_{1,n+2-j} & \cdots & (-1)^{n+1}\pi_{1,1} \\ -\pi_{2,n+1} & \pi_{2,n} & \cdots & (-1)^j \pi_{2,n+2-j} & \cdots & (-1)^{n+1}\pi_{2,1} \\ -\pi_{3,n+1} & \pi_{3,n} & \cdots & (-1)^j \pi_{3,n+2-j} & \cdots & (-1)^{n+1}\pi_{3,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\pi_{m,n+1} & \pi_{m,n} & \cdots & (-1)^j \pi_{m,n+2-j} & \cdots & (-1)^{n+1}\pi_{m,1} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where the hyperplanes should be expressed in the Hessian normal form, i.e. $\|(x_1, \cdots, x_n)^\top\|_2 = 1$.

5.2 Line Intersection in $\mathbb{P}^3$

Let $\{\mathbf{l}_i\}_{i=1,\dots,m}$ be a set of m lines of the form $\mathbf{l}_i = l_{i,1}\mathbf{e_{12}} + l_{i,2}\mathbf{e_{13}} + l_{i,3}\mathbf{e_{14}} + l_{i,4}\mathbf{e_{23}} + l_{i,5}\mathbf{e_{24}} + l_{i,6}\mathbf{e_{34}}$. A point $\mathbf{x} \in \mathbb{P}^3$ that lies on the line $\mathbf{l}_i$ satisfies:

$$\begin{aligned}\mathbf{l}_i \wedge \mathbf{x} = &(-xl_{i,4} + yl_{i,2} - zl_{i,1})\mathbf{e_{123}} \\ &+ (-xl_{i,5} + yl_{i,3} - wl_{i,1})\mathbf{e_{124}} \\ &+ (-xl_{i,6} + zl_{i,3} - wl_{i,2})\mathbf{e_{134}} \\ &+ (-yl_{i,6} + zl_{i,5} - wl_{i,4})\mathbf{e_{234}} \\ &= \mathbf{0}\end{aligned}$$

Like for equation (10), the element in $\mathbf{e_{123}}$ component can be ignored, such that the extension to the set of lines of this equation takes the following matrix form:

$$\begin{bmatrix} -l_{1,5} & l_{1,3} & 0 & -l_{1,1} \\ -l_{1,6} & 0 & l_{1,3} & -l_{1,2} \\ 0 & -l_{1,6} & l_{1,5} & -l_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ -l_{m,5} & l_{m,3} & 0 & -l_{m,1} \\ -l_{m,6} & 0 & l_{m,3} & -l_{m,2} \\ 0 & -l_{m,6} & l_{m,5} & -l_{m,4} \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

A good data normalization would be to translate the lines such their intersection is at the origin, however since this is specifically this intersection we are looking for, this approach is not possible in practice. A good numerical alternative is to scale the lines with a "relatively small" scale factor s so that the intersection is near the origin. Then, the intersection found should be scaled back by the scale factor $1/s$. In Plücker coordinates, the scale of factor s of a line is formulated as:

$$\begin{aligned}\text{Input line}: &\quad L = \{\mathbf{u} : \mathbf{v}\} \\ \text{Scaled line}: &\quad \widehat{L} = \{\mathbf{u} : s\mathbf{v}\}\end{aligned}$$

This intersection estimation is specially interesting to compute the intersection of 2 lines of $\mathbb{P}^3$ since there is no native operator in Grassmann-Cayley algebra for this purpose.

6 Conclusion

This paper presents an extension of the wedge and anti-wedge products of Grassmann-Cayley algebra that reduces the grade of a k-vector in a least square sense. These two new operators, applied on vectors expressed in homogeneous coordinates, have direct applications to line, plane and hyperplane fitting and intersection. We show that all of these problems have a direct reformulation in linear algebra that corresponds to their original formulation in projective geometry. These two operators also open some new perspectives on methods where decreasing the grade of an object may have some applications.

References

1. Peter Doubilet, Gian-Carlo Rota and Joel Stein: On the foundations of combinatorial theory: Ix combinatorial methods in invariant theory. Studies in Applied Mathematics, Volume 53, number 3, pages 185–216 (1974).
2. Marilena Barnabei, Andrea Brini and Gian-Carlo Rota: On the exterior calculus of invariant theory. Journal of Algebra, Volume 96, numeber 1, pages 120–160 (1985).
3. Stefan Carlsson: The Double Algebra: An Effective Tool for Computing Invariants in Computer Vision. In Applications of Invariance in Computer Vision, Lecture Notes in Computer Science 825, Proceedings of 2nd-joint Europe-US workshop, pages 145-164, Azores (1993)
4. Wolfgang Förstner, Ansgar Brunn and Stephan Heuel: Statistically testing uncertain geometric relations. In Mustererkennung 2000, pages 1726, Springer, Berlin (2000).
5. Christian Gebken, Christian Perwass and Gerald Sommer: Parameter Estimation from Uncertain Data in Geometric Algebra. In Advances in Applied Clifford Algebras, Volume 18, Issue 3-4, pp. 647-664 (2008).
6. Richard I. Hartley and Andrew Zisserman: Multiple View Geometry in Computer Vision, second edition. In Cambridge University Press (2004).
7. Fangfang Lu and Richard Hartley: A fast optimal algorithm for L_2 triangulation. In Proceedings of the 8th Asian conference on Computer vision - Volume Part II (ACCV'07), pp. 279-288, Japan (2007).
8. Kenichi Kanatani, Yasuyuki Sugaya and Hirotaka Niitsuma: Optimization without Minimization Search: Constraint Satisfaction by Orthogonal Projection with Applications to Multiview Triangulation. In IEICE Transactions, volume 93-D, pp. 2836-2845 (2010).
9. Leo Dorst, Daniel Fontijne and Stephen Mann: Geometric algebra for computer science: an object-oriented approach to geometry. Morgan Kaufmann series in computer graphics, Revised First Edition, Elsevier San Francisco (2009).
10. Leo Dorst and Joan Lasenby: Guide to geometric algebra in practice. Springer (2011).