Barycentric and Plücker coordinates using projective Geometric algebra

Vaclav Skala
Dept. of Computer Science and Engineering,
University of West Bohemia, Pilsen, Czech Republic
http://www.VaclavSkala.eu ORCID[0000-0001-8886-4281]

Abstract: *This paper presents the computation of the barycentric coordinates and Plücker coordinates using the projective extension of the Euclidean space and geometric algebra. Using the projective extension, it also presents a relationship between linear systems of equations* ***Ax=b*** *and* ***Ax=0*** *using the projective extension. An application of the principle of duality enables solving dual problems efficiently. The given approach uses vector notation leading to efficient implementation on GPU or efficient use of SSE instructions. As the presented approach is based on projective notation, the division operation is postponed and the proposed method leads to higher computational robustness.*

Keywords: barycentric coordinates, Plücker coordinates, principle of duality, outer product, geometric algebra

INTRODUCTION

Linear algebra and geometry are closely related research fields and many algorithms have been developed. Geometric calculus evolved from Euclid's geometry(300 BC), Descartes geometry(1637), Hamilton's Algebra of quaternions(1843), Grassmann's Extensive algebra(1844), Cayley's Matrix algebra (1854), Clifford's algebra(1878), Gibbs Vector algebra(1881), Ricci's Tensor calculus(1890) and Pauli&Dirac's Spin algebra to Geometric Algebra&Calculus, which was formulated by Hesteness[12] as Space-time algebra in 1996 [1].
Since then the Geometric Algebra (GA) has developed to the universal multi-dimensional calculus, see Calvet[5], Macdonald[21], Kanatani[17], Gunn[10]. The geometric algebra is used in many fields, e.g. physics Doran[6], computer graphics Dorst[7][8], Hildebrand[13], Vince[30][31], electrical engineering Joot[16], Esch[9], geometry Calvet[5], motion interpolation Halma[11] robotics Bayro-Corrochano[4][2], quantum computing Alves[1], applications Li[20], Perwass[23] etc.
The geometric algebra was extended to the Conformal Geometric Algebra(CGA), see Doran[6], Bayro-Corrochano[3], Li[19], Hildenbrand[14], etc. [2]

Today's linear algebra uses the Gibbs vector algebra and Cayley's matrix notation, which leads to problems if multi-dimensional formulation is to be used.

1 GEOMETRIC ALGEBRA

The vector algebra (Gibbs algebra) used nowadays uses two fundamental operations on two vectors $\mathbf{a},\mathbf{b}$ in E^n, i.e. the inner product (scalar product or dot product) $c = \mathbf{a}\cdot\mathbf{b}$, where c is a scalar value

[1] http://geocalc.clas.asu.edu/html/Evolution.html
[2] A brief introduction to the CGA: https://en.wikipedia.org/wiki/Conformal_geometric_algebra

and outer product $\mathbf{c} = \mathbf{a} \wedge \mathbf{b}$ (the cross-product $\mathbf{c} = \mathbf{a} \times \mathbf{b}$ in E^3), [3] where $\mathbf{c}$ is a bivector and it has different properties than a vector as it represents an oriented area in n-dimensional space.

The Geometric Algebra (GA) uses a "new" product called Geometric product defined as:

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b} \tag{1}$$

where $\mathbf{ab}$ is a geometric product.

In the case of the n-dimensional space, vectors are defined as $\mathbf{a} = (a_1\mathbf{e}_1 + ... + a_n\mathbf{e}_n)$, $\mathbf{b} = (b_1\mathbf{e}_1 + ... + b_n\mathbf{e}_n)$ and the $\mathbf{e}_i$ vectors form orthonormal basis vectors in E^3 then we get:

1	0-vector (scalar)	$\mathbf{e}_{12}, \mathbf{e}_{23}, \mathbf{e}_{31}$	2-vectors (bivectors)
$\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3,$	1-vector (vectors)	$\mathbf{e}_{123}$	3-vector (pseudoscalar)

It can be easily proved that the following operations are valid, including an inverse of a vector.

$$\mathbf{a} \cdot \mathbf{b} = \frac{1}{2}(\mathbf{ab} + \mathbf{ba}) \qquad \mathbf{a} \wedge \mathbf{b} = -\mathbf{b} \wedge \mathbf{a} \qquad \mathbf{a}^{-1} = \mathbf{a}/||\mathbf{a}||^2 \tag{2}$$

It can be seen, that geometric algebra is *anti-commutative* and the "pseudoscalar" I in E^3 has the basis $\mathbf{e}_1\mathbf{e}_2\mathbf{e}_3$, i.e.

$$\mathbf{e}_i\mathbf{e}_j = -\mathbf{e}_j\mathbf{e}_i \qquad \mathbf{e}_i\mathbf{e}_i = 1 \qquad \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3 = I \qquad \mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c} = q \tag{3}$$

where q is a scalar value and a short notation $\mathbf{e}_i\mathbf{e}_j = \mathbf{e}_{ij}$ can be used

In general, the geometric product is represented as:

$$\mathbf{ab} = \sum_{i,j=1}^{n,n} a_i\mathbf{e}_i b_j\mathbf{e}_j \qquad \mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n,n} a_i\mathbf{e}_i b_i\mathbf{e}_i \tag{4}$$

$$\mathbf{a} \wedge \mathbf{b} = \sum_{i,j=1 \& i \neq j}^{n,n} a_i\mathbf{e}_i b_j\mathbf{e}_j = \sum_{i,j=1, \& i>j}^{n} (a_ib_j - a_jb_i)\mathbf{e}_i\mathbf{e}_j \tag{5}$$

It is not a "friendly user" notation for a practical application and causes problems in practical implementations, primarily due to the anti-commutativity of the geometric product.

However, the geometric product can be easily represented by the tensor product, which can be represented by a matrix. As the homogeneous coordinates will be used in the following, the tensor product for the 4-dimensional case is presented [4] :

$$\mathbf{ab} \underset{repr}{\Longleftrightarrow} \mathbf{ab}^T = \mathbf{a} \otimes \mathbf{b} = \mathbf{Q} = \begin{bmatrix} a_1b_1\mathbf{e}_1\mathbf{e}_1 & a_1b_2\mathbf{e}_1\mathbf{e}_2 & a_1b_3\mathbf{e}_1\mathbf{e}_3 & a_1b_4\mathbf{e}_1\mathbf{e}_4 \\ a_1b_2\mathbf{e}_2\mathbf{e}_1 & a_2b_2\mathbf{e}_2\mathbf{e}_2 & a_2b_3\mathbf{e}_2\mathbf{e}_3 & a_2b_4\mathbf{e}_2\mathbf{e}_4 \\ a_1b_3\mathbf{e}_3\mathbf{e}_1 & a_3b_2\mathbf{e}_3\mathbf{e}_2 & a_3b_3\mathbf{e}_3\mathbf{e}_3 & a_3b_4\mathbf{e}_3\mathbf{e}_4 \\ a_1b_4\mathbf{e}_4\mathbf{e}_1 & a_4b_2\mathbf{e}_4\mathbf{e}_2 & a_4b_3\mathbf{e}_4\mathbf{e}_3 & a_4b_4\mathbf{e}_4\mathbf{e}_4 \end{bmatrix} = \mathbf{B} + \mathbf{U} + \mathbf{D} \tag{6}$$

where $\mathbf{B} + \mathbf{U} + \mathbf{D}$ are *B*ottom triangular, *U*pper triangular, *D*iagonal matrices, a_4, b_4 are the homogeneous coordinates, i.e. actually w_a, w_b (will be explained later), and the operator $\otimes$ means the anti-commutative tensor product.

[3] Massey[22] and Silagadze[24] use multi-dimensional cross-product term

[4] The vector basis $\mathbf{e}_i\mathbf{e}_j$, etc. will not be used explicitly

2 PROJECTIVE EXTENSION AND PRINCIPLE OF DUALITY

Let us consider the projective extension of the Euclidean space and the use of homogeneous coordinates. [5].

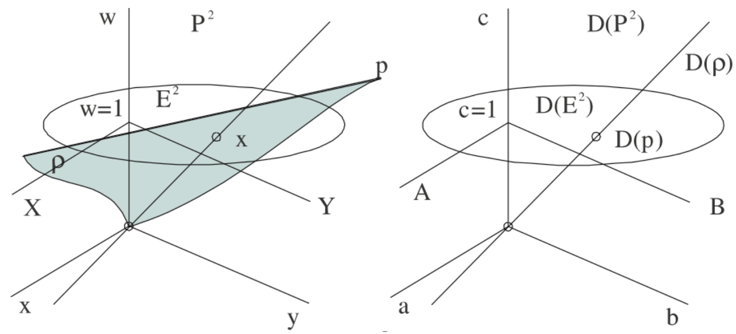


Figure 1: Projective extension and dual space

It uses homogeneous coordinates and two equivalent forms can be found:

- the form $[x_1,\dots,x_n : x_w]$ is mostly used in computer graphics-related fields, namely $[x,y:w]$ in the case of P^2, resp. $[x,y,z:w]$ in the case of P^3, where w is the homogeneous coordinate.
- the form $[x_0 : x_1,\dots,x_n]$ is used in the mathematical fields and the x_0 is the homogeneous coordinate. This form has the advantage that the homogeneous coordinate is on the first position.

It should be noted that ":" is used to emphasize that the x_w, resp x_0 has a different meaning as it is the "scaling factor", i.e. without a physical unit, while $x_1,\dots,x_n$ has different physical units, e.g. meters[m] etc.

The mutual conversion between the Euclidean space and projective space is given as:

$$X_i = \frac{x_i}{x_0} \quad x_0 \neq 0 \quad \text{, resp.} \quad X_i = \frac{x_i}{x_w} \quad x_w \neq 0 \quad , \quad i = 1,\dots,n \tag{7}$$

where X_i are coordinates in the Euclidean space.
In the case of the E^2 space

$$X = \frac{x}{x_0} \quad Y = \frac{Y}{x_0} \quad x_0 \neq 0 \quad \text{, resp.} \quad X = \frac{x}{w} \qquad Y = \frac{y}{w} \quad w \neq 0 \tag{8}$$

where (X,Y), resp.$[x,y:w]$ are coordinates in the Euclidean space E^2, resp.in the projective space P^2. The extension to the E^3, resp. E^n space is straightforward, see Vince[31], Yamaguchi[32].
The geometrical interpretation of the Euclidean ($x_w = 1$, resp. $x_0 = 1$) and the projective spaces is presented at Fig.1.
It should be noted, that a distance of a point $X = (X,Y)$, i.e. $x = [x,y:w]^T$ from a line in the E^2 is defined as

$$dist = \frac{aX + bY + c}{\sqrt{a^2+b^2}} = \frac{ax+by+cw}{w\sqrt{a^2+b^2}} \tag{9}$$

where (a,b) is the normal vector (actually it is a bivector) of the line.

[5]The concept of the projective extension for the CAD/CAM systems was deeply described in Yamaguchi[32]

2.1 Inner and outer products

The *inner product* and *outer product*, i.e. the *dot-product* and *cross-product* in the E^3, are known. However, if the projective extension of the Euclidean space is used, there are slightly different interpretations.

Let us consider vectors $\mathbf{a} = [a_1, a_2, a_3 : a_4]^T$ and $\mathbf{b} = [b_1, b_2, b_3 : b_4]^T$ in the projective space. They represents actually vectors $(a_1/a_4, a_2/a_4, a_3/a_4)$ and $(b_1/b_4, b_2/b_4, b_3/b_4)$ in the E^3 space. It can be seen, that the diagonal of the matrix $\mathbf{Q}$ actually represents the inner product in the projective representation:

$$\mathbf{a} \cdot \mathbf{b} = [(a_1b_1 + a_2b_2 + a_3b_3) : a_4b_4]^T \triangleq \frac{a_1b_1 + a_2b_2 + a_3b_3}{a_4b_4} \tag{10}$$

where $\triangleq$ means projective equivalence. The inner product represents the trace $tr(\mathbf{Q})$ of the matrix $\mathbf{Q}$ and $\mathbf{a} \cdot \mathbf{b}$ means a scalar value expressed using homogeneous coordinates.

The outer product in the E^3 —textitvector space is represented respecting anti-commutativity as:

$$\mathbf{a} \wedge \mathbf{b} \underset{\text{repr}}{\Longleftrightarrow} \sum_{i,j=1\&i>j}^{3,3} (a_ib_j\mathbf{e}_i\mathbf{e}_j - b_ia_j\mathbf{e}_i\mathbf{e}_j) = \sum_{i,j\&i>j}^{3,3} (a_ib_j - b_ia_j)\mathbf{e}_i\mathbf{e}_j \tag{11}$$

where $\mathbf{a}, \mathbf{b} \in E^3$ vector space.

However, if the projective extension is used,

$$\mathbf{a} \wedge \mathbf{b} = \underset{\text{repr}}{\Longleftrightarrow} \sum_{i,j=1\&i>j}^{4,4} (a_ib_j\mathbf{e}_i\mathbf{e}_j - b_ia_j\mathbf{e}_i\mathbf{e}_j) \triangleq \frac{\sum_{i,j\&i>j}^{3,3}(a_ib_j - b_ia_j)\mathbf{e}_i\mathbf{e}_j}{a_4b_4\mathbf{e}_4\mathbf{e}_4} \tag{12}$$

where $\mathbf{e}_i\mathbf{e}_j = 1$. It means, that the result of the outer product $\mathbf{c} = \mathbf{a} \wedge \mathbf{b}$ is represented as $\mathbf{c} = [c_1, \ldots, c_3 : c_4]^T$, where $(c_1, \ldots, c_3)$, i.e. by a bivector (normal vector) of a plane in E^3, while $c_4 = a_4b_4$ is actually a scaling factor.

It should be noted, that the outer product can be used for a solution of a linear system of equations $\mathbf{Ax} = \mathbf{b}$ or $\mathbf{Ax} = \mathbf{0}$, too.

2.2 Principle of duality

The principle of duality is essential principle, in general. Its application in geometry in connection with the implicit representation using projective geometry brings some new formulations or even new ones, see Johnson[15].

The duality principle for basic geometric entities and operators are presented by Tab.1 and Tab.2. It the E^2 case, a point is dual to a line and vice versa, the intersection of two lines is dual to a union of two points, i.e. line given by two points, similarly for the E^3 case.

Table 1: Duality of geometric entities

Duality of geometric entities						
Point in E^2	$\overset{}{\underset{\text{DUAL}}{\Longleftrightarrow}}$	Line in E^3		Point in E^3	$\underset{\text{DUAL}}{\Longleftrightarrow}$	Plane in E^3

Table 2: Duality of operators

Duality of operators		
Union $\cup$	$\underset{\text{DUAL}}{\Longleftrightarrow}$	Intersection $\cap$

3 COMPUTATION WITH HOMOGENEOUS REPRESENTATION

The geometric algebra (GA) presented above has been formulated for vectors in the Euclidean space, as presented above. However, the concept can be extended using the projective extension of the Euclidean space. It enables handling geometric entities like points, lines and planes, efficiently.

3.1 SOLUTION OF LINEAR SYSTEM OF EQUATIONS

A solution of a linear system of equations is a part of linear algebra and is used in many computational systems. It should be noted, that linear equations $\mathbf{Ax} = \mathbf{b}$ can be transformed to an implicit the homogeneous system, i.e. to the form $\mathbf{B}\xi = \mathbf{0}$, where $\mathbf{B} = [\mathbf{A}|-\mathbf{b}]$, $\boldsymbol{\xi} = [\xi_1, ..., \xi_n : \xi_w]^T$,
$x_i = \xi_i \,/\, \xi_w$, $i = 1, ..., n$.

As the solution of a linear system of equations is equivalent to the outer product (generalized cross-vector) of vectors formed by row vectors $\mathbf{a}_i$ of the matrix $\mathbf{B}$, the solution of the system is defined as:

$$\boldsymbol{\xi} = \mathbf{a}_1 \wedge \mathbf{a}_2 \wedge ... \wedge \mathbf{a}_n \qquad [\mathbf{A}|-\mathbf{b}]\boldsymbol{\xi} = 0 \qquad \mathbf{a}_i = [a_{i,1}, \dots, a_{i,n}, -b_i] \tag{13}$$

which is equivalent to a solution of the linear system of equations:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad \text{, i.e.} \quad \begin{bmatrix} a_{11} & \cdots & a_{1n} & -b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & -b_n \end{bmatrix} \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_n \\ \xi_w \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{14}$$

It is a significant result as a solution of a linear system of equations is formally the same for systems for the both cases, i.e. $\mathbf{Ax} = \mathbf{0}$ and $\mathbf{Ax} = \mathbf{b}$.
As the solution is formally determined, the formal linear operators can be used for further symbolic processing using formula manipulation, as the geometry algebra is multi-linear. Even more, it is capable to handle more complex objects generally in the n-dimensional space, i.e. oriented surfaces, volumes, etc.

However, more general rules can be derived for the n-dimensional space and the *outer product* application in Euclidean space. Let a matrix $\mathbf{M}$ is a $n \times n$ non-singular matrix representing a geometric transformation, see the Eq.15.

$$(\mathbf{Ma}) \wedge (\mathbf{Ma}_2) \wedge \ldots \wedge (\mathbf{Ma}_n) = det(\mathbf{M})^{n-1} (\mathbf{M}^{-1})^T (\mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \ldots \wedge \mathbf{a}_n) \tag{15}$$

In the case pro of the projective extension of the Euclidean space, the Eg.15 is simplified to Eq.16 due to implicit representation, as the $det(\mathbf{M})^{n-1}$ is only a multiplicative constant.

$$\begin{aligned}(\mathbf{Ma}) \wedge (\mathbf{Ma}_2) \wedge \ldots \wedge (\mathbf{Ma}_n) &= det(\mathbf{M})^{n-1} (\mathbf{M}^{-1})^T (\mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \ldots \wedge \mathbf{a}_n) \\ &\triangleq (\mathbf{M}^{-1})^T (\mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \ldots \wedge \mathbf{a}_n)\end{aligned} \tag{16}$$

where $\triangleq$ means projective equivalence as we use the implicit formulation.

Now, it is possible to use the Functional analysis approach: "Let L is a linear operator, then the following operation is valid....". As there are many linear operators like derivation, integration, Fourier and Laplace transforms etc., there is a wide variety of applications of those to the formal solution of the linear system of equations, i.e. $L(\xi)$.
However, it is necessary to respect that in the case of the projective representation specific care is to be taken for deriving rules for derivation etc., as a fraction is to be processed; similarly to other operators.

3.2 Intersections and unions

The direct consequence of the principle of duality is that the intersection point $\mathbf{x}$ of two lines $\mathbf{p}_1, \mathbf{p}_2$, resp. a line $\mathbf{p}$ passing two given points $\mathbf{x}_1, \mathbf{x}_2$, is given as:

$$\mathbf{x} = \mathbf{p}_1 \wedge \mathbf{p}_2 \underset{\text{DUAL}}{\Longleftrightarrow} \mathbf{p} = \mathbf{x}_1 \wedge \mathbf{x}_2 \tag{17}$$

where $\mathbf{p}_i = [a_i, b_i : c_i]^T$, $\mathbf{x} = [x, y : w]^T$ (w is the homogeneous coordinate), $i = 1, 2$; similarly in the dual case.

In the case of the E^3 space, a point is dual to a plane and vice versa. It means that the intersection point $\mathbf{x}$ of three planes $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \boldsymbol{\rho}_3$, resp. a plane $\boldsymbol{\rho}$ passing three given points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ is given as:

$$\mathbf{x} = \boldsymbol{\rho}_1 \wedge \boldsymbol{\rho}_2 \wedge \boldsymbol{\rho}_3 \underset{\text{DUAL}}{\Longleftrightarrow} \boldsymbol{\rho} = \mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \mathbf{x}_3 \tag{18}$$

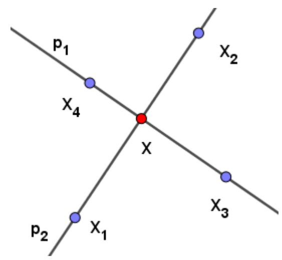
where $\mathbf{x} = [x, y, z : w]^T$, $\boldsymbol{\rho}_i = [a_i, b_i, c_i : d_i]^T$, $i = 1, 2, 3$.

It can be seen that the above formulae is equivalent to the "extended" cross-product, which in natively supported by GPU architecture. For an intersection computation, we get:

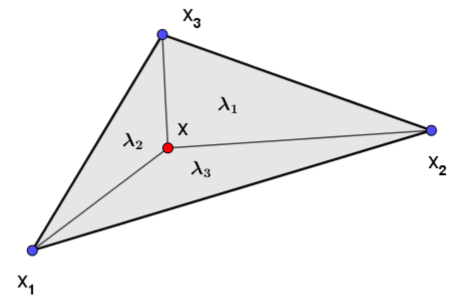
$$\mathbf{x} = \mathbf{p}_1 \wedge \mathbf{p}_2 = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_w \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix} \qquad \mathbf{x} = \boldsymbol{\rho}_1 \wedge \boldsymbol{\rho}_2 \wedge \boldsymbol{\rho}_3 = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_w \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix} \tag{19}$$

Due to the principle of duality, a dual problem solution is given as:

$$\mathbf{p} = \mathbf{x}_1 \wedge \mathbf{x}_2 = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_w \\ x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \end{bmatrix} \qquad \boldsymbol{\rho} = \mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \mathbf{x}_3 = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_w \\ x_1 & y_1 & z_1 & w_1 \\ x_1 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3 \end{bmatrix} \tag{20}$$



(a) Duality of a point and a line in E^2



(b) Barycentric coordinates in E^2

Figure 2: Duality in E^2: Lines and points, union and intersection, barycentric coordinates

The above-presented formulae prove the strength of the geometric algebra approach [6].
There is a natural question: What is the more convenient computation of the geometric product, as computation with the outer product, i.e. extended cross-product, using basis vector approach is not simple. Fortunately, the geometric product of $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2$, resp. of $\mathbf{x}_1$ and $\mathbf{x}_2$ vectors using homogeneous coordinates given as anti-commutative tensor product is given as:

$\boldsymbol{\rho}_1\boldsymbol{\rho}_2$	a_2	b_2	c_2	d_2
a_1	a_1a_2	a_1b_2	a_1c_2	a_1d_2
b_1	b_1a_2	b_1b_2	b_1c_2	b_1d_2
c_1	c_1a_2	c_1b_2	c_1c_2	a_1d_2
d_1	d_1a_2	d_1b_2	d_1c_2	d_1d_2

$\mathbf{x}_1\mathbf{x}_2$	x_2	y_2	z_2	w_2
x_1	x_1x_2	x_1y_2	x_1z_2	x_1w_2
y_1	y_1x_2	y_1y_2	y_1z_2	y_1w_2
z_1	z_1x_2	z_1y_2	z_1z_2	x_1w_2
w_1	w_1x_2	w_1y_2	w_1z_2	w_1w_2

3.3 Plücker coordinates

A line in the E^3 space is given as an intersection of two planes or in a parametric form, see Eq.21:

$$\begin{aligned}\rho_1 &: a_1X + b_1Y + c_1Z + d_1 = 0 \\ \rho_2 &: a_2X + b_2Y + c_2Z + d_2 = 0\end{aligned} \quad \text{, or} \quad \mathbf{X}(t) = \mathbf{X}_1 + (\mathbf{X}_2 - \mathbf{X}_1)\, t \tag{21}$$

where: $\rho_1 : \mathbf{n}_1^T\mathbf{X} + d_1 = 0$ and $\rho_2 : \mathbf{n}_2^T\mathbf{X} + d_2 = 0$.
The question is how to compute a line $\mathbf{p} \in E^3$ given as an intersection of two planes $\boldsymbol{\rho}_1$, $\boldsymbol{\rho}_2$, which is dual to a line determination given by two points $\mathbf{x}_1$, $\mathbf{x}_2$ as those problems are dual.

The parametric solution can be easily obtained using standard Plücker coordinates [7]. The above-given formula is difficult to derive [8] and not easy to understand and computation is complex.

$$q(t) = \frac{\omega \times \mathbf{v}}{||\omega||^2} + \omega t \qquad \mathbf{L} = \mathbf{x}_1\mathbf{x}_2^T - \mathbf{x}_2\mathbf{x}_1^T \quad \omega = [l_{41}, l_{42}, l_{43}]^T \quad \mathbf{v} = [l_{23}, l_{31}, l_{12}]^T \tag{22}$$

In 1871, Klein[18] derived that $\omega\mathbf{v} = 0$, i.e. there is a dimension reduction, see Skala[25] for details.

[6] See Skala[28][29]
[7] The "reference" point of a line is the closest point to the origin of the coordinate system, which is a substantial property, e.g. in robotics and mechanical engineering
[8] https://en.wikipedia.org/wiki/Plücker_coordinates

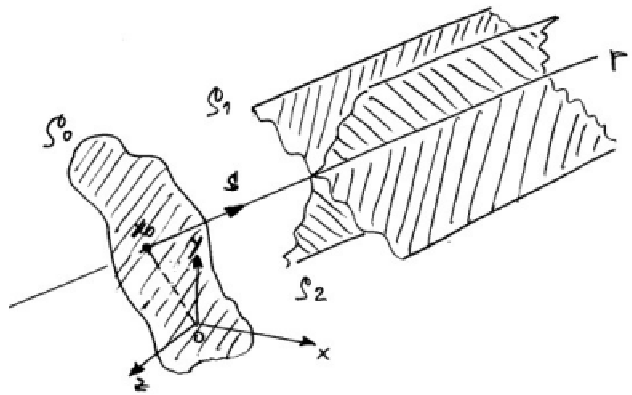


Figure 3: Plücker coordinates

However, using the outer product the formulation is easy and easy to understand, see Fig.3:

$$\mathbf{s} = \mathbf{n}_1 \wedge \mathbf{n}_2 \qquad \boldsymbol{\rho}_0 = [\mathbf{s}^T : 0]^T \qquad \mathbf{x}_0 = \boldsymbol{\rho}_1 \wedge \boldsymbol{\rho}_2 \wedge \boldsymbol{\rho}_0 \tag{23}$$

where $\mathbf{s}$ is the directional vector of and $\mathbf{x}_0$ is a "reference" point of a line, which is the closest point to the origin.

For the intersection of two planes, the principle of duality can be applied directly.

However, using geometric algebra, the principle of duality and projective representation, we can directly write:

$$\mathbf{p} = \rho_1 \wedge \rho_2 \underset{\text{DUAL}}{\Longleftrightarrow} \mathbf{p} = \mathbf{x}_1 \wedge \mathbf{x}_2 \tag{24}$$

It can be seen, that the formula given above keeps the duality in the final formulae, too.

From the formal point of view, the geometric product for the both cases is given as:

$$\boldsymbol{\rho}_1\boldsymbol{\rho}_2 \underset{\text{repr}}{\Longleftrightarrow} \boldsymbol{\rho}_1 \otimes \boldsymbol{\rho}_2 = \begin{bmatrix} a_1a_2 & a_1b_2 & a_1c_2 & a_1d_2 \\ b_1a_2 & b_1b_2 & b_1c_2 & b_1d_2 \\ c_1a_2 & c_1b_2 & c_1c_2 & c_1d_2 \\ d_1a_2 & d_1b_2 & d_1c_2 & d_1d_2 \end{bmatrix} \tag{25}$$

The dual problem formulation:

$$\mathbf{x}_1\mathbf{x}_2 \underset{\text{repr}}{\Longleftrightarrow} \mathbf{x}_1 \otimes \mathbf{x}_2 = \begin{bmatrix} x_1x_2 & x_1y_2 & x_1z_2 & x_1w_2 \\ y_1x_2 & y_1y_2 & y_1z_2 & y_1w_2 \\ z_1x_2 & z_1y_2 & z_1z_2 & z_1w_2 \\ w_1x_2 & w_1y_2 & w_1z_2 & w_1w_2 \end{bmatrix} \tag{26}$$

It means that we have computation of the Plücker coordinates for both cases, i.e. for the computation of a line $\mathbf{p} = \boldsymbol{\rho}_1 \wedge \boldsymbol{\rho}_2$ given as an intersection of two planes in E^3 and a line given by two points, i.e. as a union of two points, in E^3 as $\mathbf{p} = \mathbf{x}_1 \wedge \mathbf{x}_2$ using the projective representation and the principle of duality. It should be noted that the given approach offers: significant simplification of computation of the Plücker coordinates as it is simple and easy to derive and explain, uses vector-vector operations, which is especially convenient for SSE and GPU application one code sequence for the both cases.

The Plücker coordinates are also in mechanical engineering applications, especially in robotics, due to their simple displacement and momentum specifications. In other fields simple explanation and derivation are important arguments for GA approach application.

3.4 Barycentric coordinates

The barycentric coordinates are often used in many applications, not only in geometry. The barycentric coordinates computation, see Fig.2b, leads to a solution of a system of linear equations.

$$X_1\lambda_1 + X_2\lambda_2 + \lambda_3 X_3 = X \qquad Y_1\lambda_1 + Y_2\lambda_2 + \lambda_3 Y_3 = Y \qquad \lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{27}$$

In the matrix form:

$$\begin{bmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad \text{, resp.} \quad \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ w \end{bmatrix} \tag{28}$$

where $\mathbf{X} = (X, Y) \in E^2$ and $\mathbf{x} = [x, y : w]^T \in P^2$, i.e. in the projective space.
However, a solution of linear system equations is equivalent to the outer product application, as explained above; Skala[25][26]. Therefore, it is possible to compute the barycentric coordinates using the outer product, which is recommendable especially for the GPU oriented applications.

Let us consider the E^2 case and the barycentric interpolation between three points (a triangle vertices) given generally in the projective space as $\mathbf{x}_i = [x_i, y_i : w_i]^T$, $i = 1, \ldots, 3$ & $w_i \neq 0$, of the given triangle, and vectors:

$$\boldsymbol{\xi} = [x_1, x_2, x_3, x] \qquad \boldsymbol{\eta} = [y_1, y_2, y_3, y] \qquad \boldsymbol{\omega} = [w_1, w_2, w_3, w] \tag{29}$$

Then the barycentric coordinates μ in the homogeneous coordinates of the point $\mathbf{x} = [x, y : w]^T$ are given as:

$$\begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \\ \boldsymbol{\omega} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{, i.e.} \quad \boldsymbol{\mu} = \boldsymbol{\xi} \wedge \boldsymbol{\eta} \wedge \boldsymbol{\omega} \tag{30}$$

where $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3 : \mu_w]^T$ and the barycentric coordinates in the Euclidean space λ are given as:

$$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3) = (-\frac{\mu_1}{\mu_w}, -\frac{\mu_2}{\mu_w}, -\frac{\mu_3}{\mu_w}) \tag{31}$$

Similarly, for other dimensions, see Skala[27] for details. How simple and elegant solution!

It can be seen, that the presented computation of barycentric coordinates is simple and convenient for GPU or SSE applications. As we have assumed from the very beginning, there is no need to convert the coordinates of points from homogeneous coordinates to Euclidean coordinates. As a direct consequence of that, we save a lot of division operations and increase the robustness of the computation.

4 CONCLUSION

This contribution briefly presents geometry algebra, which is not generally known and used. However, it offers simple and efficient solutions to many computational problems if combined with the principle of duality and projective notation.
As the result of this contribution, a new formulation of the Plücker coordinates, often used in mechanical engineering and robotics, is given. As the operations are based on standard linear algebra formalism, it is simple to use. The presented approach supports direct GPU application with significant speed-up and parallelism potential. Also, the approach is applicable to d-dimensional problem solutions, as geometric algebra is multi-dimensional.
The presented approach efficiently computes the barycentric coordinates of a point in the given convex simplex, the Plücker coordinates of a line given by two points or two planes in the E^3 space. As the division operation is postponed, higher robustness of computation can be achieved.

References

[1] R. Alves, D. Hildenbrand, J. Hrdina, and C. Lavor. An online calculator for quantum computing operations based on geometric algebra. *Advances in Applied Clifford Algebras*, 32(1), 2022.

[2] E. Bayro-Corrochano. *Geometric algebra applications vol. II: Robot modelling and control*. Springer International Publishing, 2020.

[3] E. Bayro-Corrochano, L. Reyes-Lozano, and J. Zamora-Esquivel. Conformal geometric algebra for robotic vision. *Journal of Mathematical Imaging and Vision*, 24(1):55–81, 2006.

[4] E. Bayro-Corrochano and G. Scheuermann. *Geometric algebra computing: In engineering and computer science*. Springer London, 2010.

[5] R. G. Calvet. *Treatise of Plane through Geometric Algebra*. Cerdanyola del Vallés, Spain, 1 edition, 2013.

[6] C. Doran, A. N. Lasenby, and J. Lasenby. Conformal geometry, euclidean space and geometric algebra. *CoRR*, cs.CG/0203026, 2002.

[7] L. Dorst, D. Fontijne, and S. Mann. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009.

[8] L. Dorst and J. Lasenby, editors. *Guide to Geometric Algebra in Practice*. Springer, 2011.

[9] J. Esch. Geometric algebra for electrical and electronic engineers. *Proceedings of the IEEE*, 102(9):1338–1339, 2014.

[10] C. Gunn. Doing Euclidean plane geometry using projective geometric algebra. *Advances in Applied Clifford Algebras*, 27(2):1203–1232, 2017.

[11] A. Halma. *Interpolation in Conformal Geometric Algebra: Toward Unified Interpolation of Euclidean Motions in the Conformal Model of Geometric Algebra*. Lap Lampert Publ., Moldova, 1 edition, 2011.

[12] D. Hestenes. *Space-Time Algebra*. Springer/Birkhauser, Berlin, Germany, 2015.

[13] D. Hildebrand. *Foundations of Geometric Algebra Computing*. Springer-Verlag, Berlin, 1 edition, 2013.

[14] D. Hildenbrand. Geometric computing in computer graphics using conformal geometric algebra. *Computers and Graphics (Pergamon)*, 29(5):795–803, 2005.

[15] M. Johnson. Proof by duality: or the discovery of "new" theorems. *Mathematics Today*, December:138–153, 1996.

[16] P. Joot. *Geometric Algebra for Electrical Engineers: Multivector Electromagnetism*. CreateSpace Independent Publishing Platform, 2019.

[17] K. Kanatani. *Understanding geometric algebra: Hamilton, Grassmann, and Clifford for computer vision and graphics*. CRC Press, 2015.

[18] F. Klein. Notiz, betreffend den zusammenhang der liniengeometric mit der mechanik starrer körper. *Mathematische Annalen*, 4(3):403–415, 1871.

[19] H. Li. Conformal geometric algebra - a new framework for computational geometry. *Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/Journal of Computer-Aided Design and Computer Graphics*, 17(11):2383–2393, 2005.

[20] H. Li, P. J. Olver, and G. Sommer, editors. *Computer Algebra and Geometric Algebra with Applications*, volume 3519 of *Lecture Notes in Computer Science*. Springer, 2005.

[21] A. Macdonald. A survey of geometric algebra and geometric calculus. *Advances in Applied Clifford Algebras*, 27(1):853–891, 2017.

[22] W. S. Massey. Cross products of vectors in higher dimensional Euclidean spaces. *The American Mathematical Monthly*, 90(10):697–701, 1983.

[23] C. Perwass. *Geometric Algebra with Applications in Engineering*. Springer-Verlag, Berlin, 1 edition, 2009.

[24] Z. K. Silagadze. Multi-dimensional vector product. *Journal of Physics A: Mathematical and General*, 35(23):4949–4953, May 2002.

[25] V. Skala. A new approach to line and line segment clipping in homogeneous coordinates. *Visual Computer*, 21(11):905–914, 2005.

[26] V. Skala. Length, area and volume computation in homogeneous coordinates. *International Journal of Image and Graphics*, 6(4):625–639, 2006.

[27] V. Skala. Barycentric coordinates computation in homogeneous coordinates. *Computers and Graphics (Pergamon)*, 32(1):120–127, 2008.

[28] V. Skala. Intersection computation in projective space using homogeneous coordinates. *International Journal of Image and Graphics*, 8(4):615–628, 2008.

[29] V. Skala. Projective geometry and duality for graphics, games and visualization. *SIGGRAPH Asia 2012 Courses, SA 2012*, 2012.

[30] J. Vince. *Geometric Algebra: An Algebraic System for Computer Games and Animation*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[31] J. A. Vince. *Geometric Algebra for Computer Graphics*. Springer-Verlag TELOS, Santa Clara, CA, USA, 1 edition, 2008.

[32] F. Yamaguchi. *Computer-Aided Geometric Design*. Springer-Verlag Tokyo, Tokyo, Japan, 1 edition, 2002.

Acknowledgement

The authors would like to thank to colleagues at the Shandong University in Jinan (China) and University of West Bohemia in Pilsen (Czech Rep.) for stimulating of this work and discussions made.
This research was supported by the University of West Bohemia - Institutional research support (project No.1311).

Appendix

The GPU implementation of the outer product for the E^3 case using the homogeneous coordinate is quite simple. It should be noted that only 4 clocks for the *outer product* and 4 clocks for the *inner product* are needed.

```
float4 a;
a.x = dot(x1.yzw, cross(x2.yzw, x3.yzw));
a.y = - dot(x1.xzw, cross(x2.xzw, x3.xzw));
a.z = dot(x1.xyw, cross(x2.xyw, x3.xyw));
a.w = - dot(x1.xyz, cross(x2.xyz, x3.xyz));
return a;
```

or more compactly as:

```
float4 cross_4D(float4 x1, float4 x2, float4 x3)
return(
    dot(x1.yzw, cross(x2.yzw, x3.yzw)),
    - dot(x1.xzw, cross(x2.xzw, x3.xzw)),
    dot(x1.xyw, cross(x2.xyw, x3.xyw)),
    - dot(x1.xyz, cross(x2.xyz, x3.xyz))
);
```