Skala,V.: Plücker Coordinates and Extended Cross Product for Robust and Fast Intersection Computation, CGI 2016 & GACSE2016, CGI 2016 Proceedi
pp.57-60, doi>10.1145/2949035.2949050, Greece, 2016

Plücker Coordinates and Extended Cross Product for Robust and Fast Intersection Computation

Vaclav Skala
Department of Computer Science and Engineering
University of West Bohemia
Univerzitni 8, CZ 30614 Plzen, Czech Republic
http://www.VaclavSkala.eu

ABSTRACT

Many geometrically oriented problems lead to intersection computation or to its dual problems. In many cases the problem is reduced to intersection computation of two planes in E^3, e.g. intersection of two triangles. However in several cases triangles are given by vertices in the homogeneous coordinates. The usual approach is to convert coordinates to the Euclidean space and make intersection computation in the Euclidean space. This leads to extensive use of division operations and to decreased precision of computation. Another approach is an application of Plücker coordinates which are not commonly recognized in computer graphics or direct computing in the projective space. In this paper we present a relation between the extended cross product and the Plücker coordinates. The extended cross product is especially convenient for GPU application. Also a new formulation for the extended cross product using matrix notation in n-dimensional space is introduced. The presented approach leads to simple, robust and fast intersection computation of two planes on GPU. Also the advantage of the projective representation for geometrical problems solution is presented as it actually offers "doubled" mantissa length naturally and saves division operations.

CCS Concepts

- **Mathematics of computing → Mathematical analysis**
- **Computing methodologies→Computer vision 3D imaging.**

Keywords

Extended cross-product, projective space, geometric algebra, scientific computation; Plücker coordinates; computer graphics.

1. INTRODUCTION

Wide range of geometrical problems solutions are based on intersection computation, e.g. intersection of planes, triangles etc. Vast majority of algorithms use the Euclidean representation. However, the projective extension is more convenient as it offers simpler and robust solution of the given problem. A typical problem is a solution of a system of linear equations $\boldsymbol{Ax} = \boldsymbol{0}$ or $\boldsymbol{Ax} = \boldsymbol{b}$. It can be shown in the both cases, in which solutions in the Euclidean space are different, can be solved using projective representation by the extended cross product [9][12]. Another example is a computation of an intersection of two triangles in E^3. Many algorithms rely on projection to a plane of one triangle and an intersection line of planes on which the given triangle lies. The intersection of two planes can be symbolically written as $\boldsymbol{\rho}_1 \times \boldsymbol{\rho}_2$, where $\times$ means intersection, which is actually an application of the extended cross product.

Another significant advantage of the projective representation is that we get actually a doubled mantissa precission and therefore significant increase of the precision of computation. Also a problem of the Euclidean infinity disappears as the infinity is represented by the homogeneous coordinate $w = 0$. On the other hand a distance computation is more expensive [10].

2. EXTENDED CROSS PRODUCT

The scalar (dot) product and the cross product are often used in vector algebra and geometrical problems solutions. The "standard" cross-product of two vectors $\boldsymbol{a} = [a_1, a_2, a_3]^T$ and $\boldsymbol{b} = [b_1, b_2, b_3]^T$ is defined as:

$$\boldsymbol{a} \times \boldsymbol{b} = \det \begin{bmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix} \qquad (1)$$

where: $\boldsymbol{i} = [1,0,0]^T$, $\boldsymbol{j} = [0,1,0]^T$, $\boldsymbol{k} = [0,0,1]^T$. If a matrix form is needed, then we can write:

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \qquad (2)$$

In some applications the matrix form is more convenient. It will be used latter on to represent a relation between the extended cross product and the Plücker coordinates.

Let us introduce the extended cross-product of three vectors $\boldsymbol{a} = [a_1, \dots, a_n]^T$, $\boldsymbol{b} = [b_1, \dots, b_n]^T$ and $\boldsymbol{c} = [c_1, \dots, c_n]^T$, $n = 4$:

$$\boldsymbol{a} \times \boldsymbol{b} \times \boldsymbol{c} = \det \begin{bmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} & \boldsymbol{l} \\ a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix} \qquad (3)$$

$\boldsymbol{i} = [1,0,0,0]^T$, $\boldsymbol{j} = [0,1,0,0]^T$, $\boldsymbol{k} = [0,0,1,0]^T$, $\boldsymbol{l} = [0,0,0,1]^T$. It is actually related to the Grassmann's outer product.

The extended cross-product can be expressed in a matrix form as:

$$\boldsymbol{a} \times \boldsymbol{b} \times \boldsymbol{c} = (-1)^{n+1} \begin{bmatrix} 0 & -\delta_{34} & \delta_{24} & -\delta_{23} \\ \delta_{34} & 0 & -\delta_{14} & \delta_{13} \\ -\delta_{24} & \delta_{14} & 0 & -\delta_{12} \\ \delta_{23} & -\delta_{13} & \delta_{12} & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \qquad (4)$$

In this case δ_{ij} are sub-determinants with columns i, j of the matrix $\boldsymbol{T}$, which is defined as:

$$\boldsymbol{T} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \end{bmatrix} \qquad (5)$$

e.g. sub-determinant $\delta_{24} = \det \begin{bmatrix} a_2 & a_4 \\ b_2 & b_4 \end{bmatrix}$ etc.

ERRATA Eq.(18) shoud be read as [x3,y3,z3:w3] T

The extended cross-product for 5-dimensions is defined as:

$$\boldsymbol{a} \times \boldsymbol{b} \times \boldsymbol{c} \times \boldsymbol{d} = \det \begin{bmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} & \boldsymbol{l} & \boldsymbol{n} \\ a_1 & a_2 & a_3 & a_4 & a_5 \\ b_1 & b_2 & b_3 & b_4 & b_5 \\ c_1 & c_2 & c_3 & c_4 & c_5 \\ d_1 & d_2 & d_3 & d_4 & d_5 \end{bmatrix} \qquad (6)$$

$\boldsymbol{i} = [1,0,0,0,0]^T$, $\boldsymbol{j} = [0,1,0,0,0]^T$, $\boldsymbol{k} = [0,0,1,0,0]^T$,
$\boldsymbol{l} = [0,0,0,1,0]^T$, $\boldsymbol{n} = [0,0,0,0,0,1]^T$.
In spite of the "complicated" description above, this approach leads to a faster computation in the case of lower dimensions. It is also applicable to higher dimensions [13].

3. PROJECTIVE REPRESENTATION

Projective representation and its application for computation are considered to be "mysterious" or too complex. However, they are used naturally frequently in a form of fractions, e.g. a/b. We also know that fractions help us to express values, which cannot be expressed precisely due to the limited length of a mantissa, e.g. $1/3 = 0{,}33 \ldots\ldots .333\ldots. = 0.\bar{3}$. It should be noted that the projective representation actually doubles double precision of computation. In the following we will explore projective representation, actually rational fractions, and its applicability.

The projective extension of the Euclidean space is used commonly in computer graphics and computer vision mostly for geometric transformations. However, in computational sciences, the projective representation is not used, in general. This chapter shortly introduces basic properties and mutual conversions. More detailed description of projective representation and applications can be found in [8][7][11].

The given point $\boldsymbol{X} = (X, Y, Z)$ in the Euclidean space E^3 is represented in homogeneous coordinates as $\boldsymbol{x} = [x, y, z\colon w]^T$, $w \neq 0$. It can be seen that $\boldsymbol{x}$ is actually a line in the projective space P^3 with the origin excluded. Mutual conversions are defined as:

$$X = \frac{x}{w} \qquad Y = \frac{y}{w} \qquad Z = \frac{z}{w} \qquad (7)$$

where: $w \neq 0$ is the homogeneous coordinate. Note that the homogeneous coordinate w is actually a scaling factor with no physical meaning, while x, y, z are values with physical units in general.

The projective representation enables us nearly double precision of representation as the mantissa of x, resp. y, z and w are used for a value representation (all doubles). However we have to distinguish two different data types, i.e.

- Projective representation of a value $\boldsymbol{X} = (X_1, \ldots, X_n)$ is one dimensional array $\boldsymbol{x} = [x_1, \ldots, x_n\colon x_w]^T$, e.g. coordinates of a point, which is fixed to the origin of the coordinate system.
- Projective representation of a vector (in the mathematical meaning) $\boldsymbol{A} = (A_1, \ldots, A_n)$ is represented by one dimensional array $\boldsymbol{a} = [a_1, \ldots, a_n\colon a_w]^T$. In this case the homogeneous coordinate a_w is actually just a scaling factor and the vector should be handled as $\boldsymbol{a} = \left[\frac{a_1}{a_w}, \ldots, \frac{a_n}{a_w}\colon 0\right]^T$. Any vector is not fixed to the origin of the coordinate system and it is "movable".

An attention should be kept on the correctness of operations used.

A line in E^3can be expresses as a join of two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ or as an intersection of two planes ρ_1 and ρ_2 or in a parametric form as

$$\boldsymbol{x}(t) = \boldsymbol{x}_A + \boldsymbol{s}t \qquad (8)$$

Usually a conversion to the Euclidean space is made and then the line is computed. In this way we get a linear interpolation with a linear parameterization. If the Eq. (8) is applied directly in the projective space, i.e. including the homogeneous coordinate w we obtain a linear interpolation with a non-linear, but monotonous parameterization [10]. This can save many division operations.

4. PLÜCKER COORDINATES

The Plücker coordinates are used often in computer graphics and in mechanical engineering. In computer graphics and geometry are used quite rarely. The geometry the Plücker coordinates are used for determination of a line given by two points in the homogeneous coordinates without need to convert them to the Euclidean space. The advantage of that is that division operations are not need.

Let us consider a little bit more difficult problems formulated as follows:

- determine a line $\boldsymbol{x}(t) \in E^2$ if given by two points $\boldsymbol{x}_i$
- determine a line $\boldsymbol{x}(t) \in E^3$ if given by two planes $\boldsymbol{\rho}_i$

if the parametric form is required.

These problems formulation seem to be trivial if $w_i = 1$ or the division operation are allowed, i.e. solutions are expected in the Euclidean space directly.

On the other hand, a classic rule for robustness is to "postpone division operation to the last moment possible". Even if division is allowed, the 2[nd] case seems to be more difficult not only from the robustness point of view as the line is considered as an intersection of two planes, i.e. a common solution of their implicit equations.

In the following will describe a method for determination of a line in the E^3 space for those two possible cases without use of division directly in the projective space. The Plücker coordinates can help us to formalize and resolve this problem efficiently.

Let us consider two points in the homogeneous coordinates:

$$\boldsymbol{x}_1 = [x_1, y_1, z_1\colon w_1]^T \qquad \boldsymbol{x}_2 = [x_2, y_2, z_2\colon w_2]^T \qquad (9)$$

Then the Plücker coordinates l_{ij} of the anti-symmetric matrix $\boldsymbol{L}$ are defined as:

$$\boldsymbol{L} = \boldsymbol{x_1}\,\boldsymbol{x_2^T} - \boldsymbol{x_2}\,\boldsymbol{x_1^T}$$

$$\begin{aligned} l_{41} &= w_1 x_2 - w_2 x_1 & l_{23} &= y_1 z_2 - y_2 z_1 \\ l_{42} &= w_1 y_2 - w_2 y_1 & l_{31} &= z_1 x_2 - z_2 x_1 \\ l_{43} &= w_1 z_2 - w_2 z_1 & l_{12} &= x_1 y_2 - x_2 y_1 \\ l_{ii} &= 0 & l_{ij} &= -l_{ji} \end{aligned} \qquad (10)$$

Let us define two vectors $\boldsymbol{\omega}$ and $\boldsymbol{\nu}$ as:

$$\boldsymbol{\omega} = [l_{41}, l_{42}, l_{43}]^T \qquad \boldsymbol{\nu} = [l_{23}, l_{31}, l_{12},]^T \qquad (11)$$

It means that $\boldsymbol{\omega}$ represents the "directional vector", while $\boldsymbol{\nu}$ represents the "positional vector". It can be seen that for the Euclidean space ($w = 1$) we get:

$$\boldsymbol{X_2} - \boldsymbol{X_1} = \boldsymbol{\omega} \qquad \boldsymbol{X_1} \times \boldsymbol{X_2} = \boldsymbol{\nu} \qquad (12)$$

where $\boldsymbol{X}_i = \left({}^{x_i}/_{w_i}, {}^{y_i}/_{w_i}, {}^{z_i}/_{w_i}\right)$ are coordinates of points in the Euclidean space. The line p given by two points in the homogeneous coordinates is then given as:

ERRATA Eq.(18) shoud be read as [x3,y3,z3:w3] T

Skala,V.: Plücker Coordinates and Extended Cross Product for Robust and Fast Intersection Computation, CGI 2016 & GACSE2016, CGI 2016 Proceedi
pp.57-60, doi>10.1145/2949035.2949050, Greece, 2016

$$\boldsymbol{X}(t) = \frac{\boldsymbol{v} \times \boldsymbol{\omega}}{\|\boldsymbol{\omega}\|^2} + \boldsymbol{\omega} t \tag{13}$$

where $\boldsymbol{X} = (X, Y, Z)$ are coordinates of points on the line p.

Note, that the point $\boldsymbol{X}(t = 0)$ is the closest point to the origin of the coordinate system. If the projective representation is used, we can write:

$$\boldsymbol{x}(t) = [\boldsymbol{v} \times \boldsymbol{\omega} + \boldsymbol{\omega}\|\boldsymbol{\omega}\|^2 t : \|\boldsymbol{\omega}\|^2]^T \tag{14}$$

and if re-parameterization $\tau = \|\boldsymbol{\omega}\|^2 t$ is applied, then multiplication operations can be saved, as:

$$\boldsymbol{x}(\tau) = [\boldsymbol{v} \times \boldsymbol{\omega} + \boldsymbol{\omega}\tau : \|\boldsymbol{\omega}\|^2]^T \tag{15}$$

Now, the principle of duality can be used for computation of two planes intersection.

5. PRINCIPLE OF DUALITY

The projective representation offers also one very important property – principle of duality. The principle of duality in E^2 states that any theorem remains true when we interchange the words "point" and "line", "lie on" and "pass through", "join" and "intersection", "collinear" and "concurrent" and so on. Once the theorem has been established, the dual theorem is obtained as described above [1][6][9][15]. In other words, the principle of duality says that in all theorems it is possible to substitute the term "point" by the term "line" and the term "line" by the term "point" etc. in E^2 and the given theorem stays valid. Similar duality is valid for E^3 as well, i.e. the terms "point" and "plane" are dual etc. it can be shown that operations "join" a "meet" are dual as well. This helps a lot to solve some geometrical problems.

6. PLÜCKER COORDINATES AND CROSS-PRODUCT

Let three planes $\boldsymbol{\rho}_i = [a_i, b_i, c_i : d_i]^T = [\boldsymbol{n}_i^T : d_i]^T \in E^3$ are given. The intersection point $\boldsymbol{x}$ of those planes is given as [7][9][11]:

$$\mathbf{x} = \boldsymbol{\rho}_1 \times \boldsymbol{\rho}_2 \times \boldsymbol{\rho}_3 = \begin{bmatrix} 0 & -\delta_{34} & \delta_{24} & -\delta_{23} \\ \delta_{34} & 0 & -\delta_{14} & \delta_{13} \\ -\delta_{24} & \delta_{14} & 0 & -\delta_{12} \\ \delta_{23} & -\delta_{13} & \delta_{12} & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ b_3 \\ c_3 \\ d_3 \end{bmatrix} \tag{16}$$

where: δ_{ij} are sub-determinants w of the matrix $\boldsymbol{T}$:

$$\boldsymbol{T} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \end{bmatrix} \quad \text{e.g.} \quad \delta_{24} = \det \begin{bmatrix} b_1 & d_1 \\ b_2 & d_2 \end{bmatrix} \tag{17}$$

Due to the principle of duality, the plane $\boldsymbol{\rho}$ given by three points $\boldsymbol{x}_i = [x_i, y_i, z_i : w_i]^T \in E^3$ in the homogeneous coordinates is given as:

$$\boldsymbol{\rho} = \boldsymbol{x}_1 \times \boldsymbol{x}_2 \times \boldsymbol{x}_3 = \begin{bmatrix} 0 & -\delta_{34} & \delta_{24} & -\delta_{23} \\ \delta_{34} & 0 & -\delta_{14} & \delta_{13} \\ -\delta_{24} & \delta_{14} & 0 & -\delta_{12} \\ \delta_{23} & -\delta_{13} & \delta_{12} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ y_2 \\ z_3 \\ w_4 \end{bmatrix} \tag{18}$$

where: δ_{ij} are sub-determinants w of the matrix $\boldsymbol{T}$:

$$\boldsymbol{T} = \begin{bmatrix} x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \end{bmatrix} \quad \text{e.g.} \quad \delta_{24} = \det \begin{bmatrix} y_1 & w_1 \\ y_2 & w_2 \end{bmatrix} \tag{19}$$

It can be seen that $\delta_{ij} = -\delta_{ji}$ and values are "somehow" connected with the Plücker coordinates:

$$\begin{aligned} \delta_{12} &= x_1 y_2 - x_2 y_1 & \delta_{14} &= x_1 w_2 - x_2 w_1 \\ \delta_{23} &= y_1 z_2 - y_2 z_1 & \delta_{24} &= y_1 w_2 - y_2 w_1 \\ \delta_{13} &= x_1 z_2 - x_2 z_1 & \delta_{34} &= z_1 w_2 - z_2 w_1 \end{aligned} \tag{20}$$

Now we can see that the line expressed in the homogeneous coordinates is defined as:

$$\boldsymbol{x}(t) = [-(\delta_{14}, \delta_{24}, \delta_{34}) + (\delta_{23}, \delta_{31}, \delta_{12})t : (w_1 w_2)]^T \tag{21}$$

where $\boldsymbol{x}(0) = \boldsymbol{x}_1$ and $\boldsymbol{x}(1) = \boldsymbol{x}_2$.

In the Euclidean space we can write:

$$\boldsymbol{X}(t) = -\frac{(\delta_{12}, \delta_{24}, \delta_{34})}{w_1 w_2} + \frac{(\delta_{23}, \delta_{31}, \delta_{12})}{w_1 w_2} t \tag{22}$$

In the following we will demonstrate this approach on very simple geometrical problems like intersection of two planes.

7. INTERSECTION OF TWO PLANES

An intersection of two planes is often used in computer graphics and computer vision. Unfortunately in many cases available solutions are not robust or formula are neither simple nor convenient for GPU use. The intersection of two planes ρ_1 and ρ_1 in E^3 is seemingly a simple problem, but surprisingly computationally expensive, Figure 1.

Let us consider the "standard" solution in the Euclidean space and a solution using the projective approach.

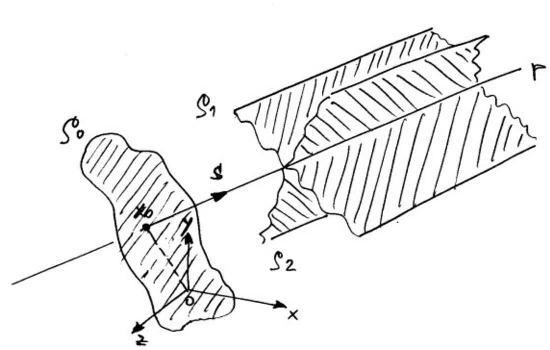

Figure 1: A line as the intersection of two planes

In the following a new formulation of two plane intersection is presented and as the projective space is used for formulation, the solution is quite simple.

7.1 Euclidean solution

Let two planes $\boldsymbol{\rho}_i = [a_i, b_i, c_i : d_i]^T = [\boldsymbol{n}_i^T : d_i]^T \in E^3$. Then the directional vector $\boldsymbol{s}$ of a parametric line $\boldsymbol{X}(t) = \boldsymbol{X}_0 + \boldsymbol{s}t$ is given by a cross product:

$$\boldsymbol{s} = \boldsymbol{n}_1 \times \boldsymbol{n}_2 \equiv [a_3, b_3, c_3]^T \tag{23}$$

The point $\boldsymbol{X}_0 \in E^3$ of the line is given as:

$$\begin{aligned} X_0 &= \frac{d_2 \begin{vmatrix} b_1 & c_1 \\ b_3 & c_3 \end{vmatrix} - d_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix}}{DET} \\ Y_0 &= \frac{d_2 \begin{vmatrix} a_3 & c_3 \\ a_1 & c_1 \end{vmatrix} - d_1 \begin{vmatrix} a_3 & c_3 \\ a_2 & c_2 \end{vmatrix}}{DET} \\ Z_0 &= \frac{d_2 \begin{vmatrix} a_1 & b_1 \\ a_3 & b_3 \end{vmatrix} - d_1 \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix}}{DET} \end{aligned} \tag{24}$$

where $DET = (\boldsymbol{n}_1 \times \boldsymbol{n}_2).\boldsymbol{s}$. It can be seen that the formula above is quite difficult to remember and its derivation is not simple. It should be noted that there is a severe problem with stability and robustness if a condition like $|DET| < eps$ is used. Also the formula is not convenient for GPU or SSE applications. There is another equivalent solution based on Plücker coordinates and duality application, see [8] [11].

ERRATA Eq.(18) shoud be read as [x3,y3,z3:w3] T

7.2 Duality application

Let us consider intersection of two planes again. The directional vector $\boldsymbol{s}$ of the line is given by Eq.(23):

$$s_x : s_y : s_z = \begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix} : \begin{vmatrix} c_1 & a_1 \\ c_2 & a_2 \end{vmatrix} : \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} \tag{25}$$

that is actually the ratio $l_{23} : l_{31} : l_{12}$ if the principle of duality is used, i.e. vector of $[a_i, b_i, c_i : d_i]^T$ instead of $[x_i, y_i, z_i, w_i]^T$ is used. Similarly, we can write for the $l_{41} : l_{42} : l_{43}$ values. It means that using the principle of duality we get a new formulation of the intersection of two planes as:

$$\boldsymbol{X}(t) = \frac{\boldsymbol{v} \times \boldsymbol{\omega}}{\|\boldsymbol{\omega}\|^2} + \boldsymbol{\omega} t \tag{26}$$

and if the projective notation is used, then:

$$\boldsymbol{x}(t) = [\boldsymbol{v} \times \boldsymbol{\omega} + \boldsymbol{\omega}\|\boldsymbol{\omega}\|^2 t : \|\boldsymbol{\omega}\|^2]^T \tag{27}$$

If $\|\boldsymbol{\omega}\| = 0$ then the given planes are parallel.

Now, let us explore a solution based on the projective solution directly.

7.3 Projective solution

Given two planes ρ_1 and ρ_2 again. Then the directional vector $\boldsymbol{s}$ of their intersection is given as:

$$\boldsymbol{s} = \boldsymbol{n}_1 \times \boldsymbol{n}_2 \triangleq [s_x, s_y, s_z : 0]^T \tag{28}$$

where $\triangleq$ means projectively equivalent.

We want to determine the point $\boldsymbol{x}_0$ of the line given as an intersection of those two planes. Let us consider a plane ρ_0 passing the origin of the coordinate system with the normal vector $\boldsymbol{n}_0$ equivalent to $\boldsymbol{s}$, see Figure 1. This plane ρ_0 is represented as:

$$\boldsymbol{\rho}_0 = [a_0, b_0, c_0 : 0]^T = [\boldsymbol{s}^T : 0]^T \tag{29}$$

Then the point $\boldsymbol{x}_0$ is simply determined as an intersection of three planes ρ_1, ρ_2, ρ_0 as:

$$\boldsymbol{x}_0 = \boldsymbol{\rho}_1 \times \boldsymbol{\rho}_2 \times \boldsymbol{\rho}_0 = [x_0, y_0, z_0 : w_0]^T \tag{30}$$

Now, the line given as the intersection of two planes is given as:

$$\begin{aligned} \boldsymbol{x}(t) &= [x_0, y_0, z_0 : w_0]^T + [s_x, s_y, s_z : 0]^T t \\ &\triangleq \left[[x_0, y_0, z_0]^{\mathrm{T}} + [s_x, s_y, s_z]^{\mathrm{T}} t : w_0\right]^T \end{aligned} \tag{31}$$

As the projective representation is used, there is no problem with infinity if the planes are parallel and with precision of computation as the mantissa of the coordinate and homogeneous coordinate are used.

The extended cross-product for 4D can be implemented as:

```
float4 cross_4D(float4 x1, float4 x2, float4 x3)
{ 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}
```

The proposed approach is simple, easy to understand, elegant and convenient for SEE and GPU applications as it uses vector-vector operations. It offers significant speed up as the "standard" cross-product is implemented in hardware as an instruction. Due to the implicit formulation using the projective extension of the Euclidean space the presented approach is recommendable also for solving other geometrical problems. This approach is closely related to the Clifford algebra and Grassmann algebra approaches.

Another approach to solve similar problems can be application of the geometric algebra [2][3][4].

8. CONCLUSION

A new relation between the Euclidean Plücker and projective representations is presented with an application to two planes intersection. A new method of two planes intersection using extended cross-product is presented. As the projective representation actually doubles mantissa, the proposed method is robust and fast. Also a problem with infinity is eliminated. The presented approach is convenient especially for vector-vector architectures and GPU computing.

9. ACKNOWLEDGMENTS

The author would like to thank to colleagues and recent students at the University of West Bohemia for their comments. Thanks belong also to anonymous reviewer for hints and corrections they made. This research was supported by the MSMT CR - project No. LH12181.

10. REFERENCES

[1] Coxeter,H.S.M.: Introduction to Geometry, J.Wiley, 1961.
[2] Dorst,L., Fontine,D., Mann,S.: Geometric Algebra for Computer Science, Morgan Kaufmann, 2007
[3] Gonzales Calvet,R.: Treatise of Plane Geometry through Geometric Algebra, 2007
[4] Hildenbrand,D.: Foundations of Geometric Algebra Computing, Springer Verlag, , 2012
[5] Kanatani,K.: Undestanding geometric Algebra, CRC Press, 2015
[6] Johnson,M.: Proof by Duality: or the Discovery of "New" Theorems, Mathematics Today, December 1996.
[7] Skala,V.: A New Approach to Line and Line Segment Clipping in Homogeneous Coordinates, The Visual Computer, Vol.21, No.11, pp.905 914, Springer Verlag, 2005
[8] Skala,V.: Length, Area and Volume Computation in Homogeneous Coordinates, International Journal of Image and Graphics, Vol.6., No.4, pp.625-639, 2006
[9] Skala,V.: Barycentric Coordinates Computation in Homogeneous Coordinates, Computers & Graphics, Elsevier, ISSN 0097-8493, Vol. 32, No.1, pp.120-127, 2008
[10] Skala,V.: Projective Geometry, Duality and Precision of Computation in Computer Graphics, Visualization and Games, Tutorial Eurographics 2013, Girona, 2013
[11] Skala,V.: Intersection Computation in Projective Space using Homogeneous Coordinates, Int.Journal on Image and Graphics, ISSN 0219-4678, Vol.8, No.4, pp.615-628, 2008
[12] Skala,V.: Modified Gaussian Elimination without Division Operations, ICNAAM 2013, Rhodos, Greece, AIP Conf.Proceedings, No.1558, pp.1936-1939, AIP Publ. 2013
[13] Skala,V.: Extended Cross-product and Solution of a Linear Systems of Equations, ICCSA Proc, accepted, Springer, 2016
[14] Vince,J.: Geometric Algebra for Computer Science, Springer, 2008
[15] Yamaguchi,F. Computer Aided Geometric Design: A totally Four Dimensional Approach, Springer Verlag, 2002

ERRATA Eq.(18) shoud be read as [x3,y3,z3:w3] T