2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI) | 978-1-6654-9704-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/SAMI54271.2022.9780728

Geometric Transformations and Tensor Product

Vaclav Skala
Dept. of Computer Science and Engineering
Faculty of Applied Sciences, University of West Bohemia
Pilsen, Czech Republic, Country
www.VaclavSkala.eu ORCID 0000-0001-8886-4281

***Abstract*—This contribution describes new useful geometric transformations using the tensor product. The geometric transformations are used widely in many applications, especially in CAD/CAM systems, systems for Civil Engineering, computer graphics, virtual and augmented reality, etc. The basic geometric transformations, e.g. rotation, translation, scaling, etc., are usually applied on points represented in homogeneous coordinates and described in many relevant books. However, there are also other primitives, e.g. lines, planes, normal "vector" of a line, plane, triangle, etc., on which some geometric transformations can be applied.**

***Index Terms*—Geometric transformations, geometric algebra, tensor product, outer product, inner product, homogeneous coordinates, computer graphics, computer vision, linear algebra, matrix operations**

I. INTRODUCTION

Geometric transformations play a very important role in many applications, especially in CAD/CAM systems, civil engineering systems, virtual reality, augmented reality, computer graphics, computer vision, etc. Geometric entities used are described in the Euclidean space, but mainly using the projective extension, see Yamaguchi [1]. It helps to formulate some algorithms more efficiently and leads to more robust computation as well, application of the principle of duality Johnson-1996 [2] and to interesting solutions of dual problems, such as:

- equivalence of the outer product (cross product) and the Gauss elimination method for $\mathbf{Ax} = \mathbf{b}$, Skala [3] [4] [5].
- volume, area, length computation, Skala [6],
- the duality between the operations "join" and "intersect" in E^2 and E^3, Skala [7],
- intersection computation of lines, planes, quadratic surfaces, etc. Calvet [8], Skala [9] [10] [11],
- line clipping algorithms in E^2 Skala [12] [13]. .

The geometric transformations using homogeneous coordinates, e.g. translation, rotation, scaling etc. are well known and described in Vince [14], Hughes [15], Foley [16], Agoston [17], [18], Ferguson [19], Salomon [20], Thomas [21], Angel [22].

The transformations are used for the point position change, in general. However, there are also other primitives like line, plane , etc. used and their position is changeable. Also, a triangle normal "vector" (it is actually a bivector Vince [23]) is to be transformed differently from the triangle vertices.

Supported by the University of West Bohemia - Institutional research support No.1311.

II. TENSOR PRODUCT

The tensor product WiKi [24] is not frequently used, however it is very useful. Generally, it is the non-commutative product on two vectors $\mathbf{v} = [v_1, v_2, \ldots, v_n]^T$ and $\mathbf{w} = [w_1, w_2, \ldots, w_n]^T$ defined as:

$$\mathbf{v} \otimes \mathbf{w} = \begin{bmatrix} v_1 w_1 & v_1 w_2 & \cdots & v_1 w_m \\ v_2 w_1 & v_2 w_2 & \cdots & v_2 w_m \\ \vdots & \vdots & \ddots & \vdots \\ v_n w_1 & v_n w_2 & \ldots & v_n w_m \end{bmatrix} \tag{1}$$

The tensor product is multilinear WiKi [25], that is a great advantage in its use for solving geometrical problems and can be also applied on functions Mochizuki [26].

III. GEOMETRIC TRANSFORMATIONS

The geometric transformation are described by different operators, i.e. $\pm$, $*$ in the E^3 space. It causes problems with the inverse operations formulations as matrices are used to represent such operations. The basic geometric transformation in the E^3 case can be described using homogeneous coordinates. It means that a point $\mathbf{X} = (X, Y, Z)$ is represented as $\boldsymbol{x} = [x, y, z : w]^T$ and the conversion is given, see Gunn [27], as:

$$X = \frac{x}{w} \quad , \quad Y = \frac{y}{w} \quad , \quad Z = \frac{z}{w} \quad , \quad w \neq 0 \tag{2}$$

Using the homogeneous coordinates, the basic geometric transformations can be described as follows, see Vince [14]:

A. Translation

The translation operation moves the given point $\boldsymbol{x}$ to a position $\boldsymbol{x} + (a, b, c)$ and the transformation is defined as:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \tag{3}$$

B. Rotation

This operation rotates the given point $\boldsymbol{x}$ by an angle φ to a new position $\boldsymbol{x'}$. The rotation transformation has the origin of the coordinates system as the reference points. In the following, the *right-hand* coordinates system is used.

The rotation in the xy plane, i.e. rotation around the z-axis, is defined as:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \tag{4}$$

The rotation in the yz plane, i.e. rotation around the x-axis, is defined as:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi & 0 \\ 0 & \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \tag{5}$$

The rotation in the zx plane, i.e. rotation around the y-axis, is defined as:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 & a \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \tag{6}$$

Note, that the sign "-" is on the 3^{rd} row due to the coordinate system orientation.

C. Scaling

The scaling operation scales coordinates of the given point $\boldsymbol{x}$ to a new position $\boldsymbol{x}$. The scaling transformation has the origin of the coordinates system as the reference points. The transformation is defined as:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & a \\ 0 & s_y & 0 & b \\ 0 & 0 & s_z & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \tag{7}$$

The geometric transformations can then be accumulated using the multiplication operation only. This is a great advantage for which the homogeneous coordinates are used for geometric transformations and GPU is optimized for those.

However, the above mentioned transformation matrices cannot be used for geometric transformations for lines, planes, normal of a triangle , etc.

IV. Lines in E^2 and planes in E^3

Lines and planes are infinite geometrical elements and can be described by parametric or implicit form. Such elements have to be handled differently.

The line p in the E^2 space is described in the implicit form as:

$$\begin{aligned} aX + bY + c = 0 \quad , \quad \mathbf{X} = (X, Y) \quad or \\ ax + by + cw = 0 \quad , \quad \mathbf{a}^T\ \boldsymbol{x} = 0 \end{aligned} \tag{8}$$

where $\boldsymbol{x} = [x, y : w]^T$ are a point coordinates in the homogeneous coordinates and $\mathbf{a} = [a, b : c]^T$ are coefficients of the line p.

The plane ρ in the E^3 space is described in the implicit form as:

$$\begin{aligned} aX + bY + cZ + d = 0 \quad , \quad \mathbf{X} = (X, Y, Z) \quad or \\ ax + by + cz + dw = 0 \quad , \quad \mathbf{a}^T\ \boldsymbol{x} = 0 \end{aligned} \tag{9}$$

where $\boldsymbol{x} = [x, y, z : w]^T$ are a point coordinates in the homogeneous coordinates and $\mathbf{a} = [a, b, c : d]^T$ are coefficients of the plane ρ.

V. Geometric Algebra

The vector algebra (Gibbs algebra) used nowadays uses two basic 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, while the outer product (the cross-product in the E^3) $\mathbf{c} = \mathbf{a} \wedge \mathbf{b}$, where $\mathbf{c}$ is a bivector and has a different properties than a vector as it represents an oriented area in n-dimensional space, in general, see Massey [28], Silagadze [29], Skala [30].

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

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b} \tag{10}$$

where $\mathbf{ab}$ is the *geometric product* and $\mathbf{a} \wedge \mathbf{b}$ is the *outer-product*, i.e. the *cross-product* in the case of E^3, and $\mathbf{a} \cdot \mathbf{b}$ is the *dot-product*, i.e. *scalar-product*, see
Generally, in the case of the n-dimensional space, vectors are defined as:

$$\mathbf{a} = (a_1\mathbf{e}_1, \ldots, a_n\mathbf{e}_n), \quad \mathbf{b} = (b_1\mathbf{e}_1, \ldots, b_n\mathbf{e}_n) \tag{11}$$

where $\{\mathbf{e}_i\}_{i=1}^n$ are the orthonormal basis vectors in E^n. The geometry algebra uses the following operations, including the inverse of a vector.

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} = \frac{1}{2}(\mathbf{ab} + \mathbf{ba}) \quad \mathbf{a} \wedge \mathbf{b} = -\mathbf{b} \wedge \mathbf{a} \\ \mathbf{a}^{-1} = \frac{\mathbf{a}}{\|\mathbf{a}\|^2} \\ \mathbf{e}_i\mathbf{e}_j = -\mathbf{e}_j\mathbf{e}_i \quad \mathbf{e}_i\mathbf{e}_i = 1 \end{aligned} \tag{12}$$

In the case of the E^3 space, the geometric product is defined as, see Vince [23] [31]:

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \times \mathbf{b} \tag{13}$$

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-vectors (vectors)	$\mathbf{e}_{123} = \mathbf{I}$	3-vector (pseudoscalar)
$\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c} = q$		q is a scalar value (pseudoscalar)	

TABLE I
GEOMETRIC PRODUCT PROPERTIES IN E^3

The significant advantage of the geometric algebra is, that it is more general than the Gibbs algebra and can handle all objects with dimensionality up to n. It means, that the description is unified, see Macdonald [32], Doran [33] and Halma [34]. It is a great advantage in engineering applications and solving geometric problems Perwass [35], Li [36], Dorst [37], Kanatani [38], Hildebrand [39].

VI. INTERSECTION COMPUTATION

Intersection computation of lines in E^2 space and planes in E^3 space is the fundamental operation in solutions of geometrical problems. The implicit description also enables the use of geometric algebra for the projective extension of the Euclidean space and the homogeneous coordinates. The intersection operation is *dual* to the *join* operation, Johnson [2].

The intersection and joining (union) operations using the homogeneous coordinates and the outer product $\wedge$ is computationally simple (actually, for the joining operator the $\vee$ is used sometimes).

A. Case E^2

Computation of a line p given by two points $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, i.e. the join operation, and computation of the intersection point $\boldsymbol{x}$ of two lines p_1, p_2, i.e. intersection as the dual problem, are given as:

$$\mathbf{p} = \boldsymbol{x}_1 \wedge \boldsymbol{x}_2 \quad , \quad \boldsymbol{x} = \mathbf{p}_1 \wedge \mathbf{p}_2 \tag{14}$$

where $\mathbf{p} = [a, b : c]^T$ are coefficients of the line in the implicit form and $\boldsymbol{x} = [x, y, z : w]^T$ are point intersection coordinates in the homogeneous coordinates. Using the determinant notation:

$$\mathbf{p} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \end{vmatrix} \quad , \quad \boldsymbol{x} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix} \tag{15}$$

where $\mathbf{i} = [1, 0, 0]^T$, $\mathbf{j} = [0, 1, 0]^T$, $\mathbf{k} = [0, 0, 1]^T$ are the basis vectors.

B. Case E^3

Computation of a plane ρ given by three points $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$ i.e. the join operation, and computation of the intersection point $\boldsymbol{x}$ of three planes ρ_1, ρ_2, ρ_3 i.e. intersection as the dual, problem are given as:

$$\boldsymbol{\rho} = \boldsymbol{x}_1 \wedge \boldsymbol{x}_2 \wedge \boldsymbol{x}_2 \quad , \quad \boldsymbol{x} = \boldsymbol{\rho}_1 \wedge \boldsymbol{\rho}_2 \wedge \boldsymbol{\rho}_3 \tag{16}$$

where $\boldsymbol{\rho} = [a, b, c : d]^T$ are coefficients of the plane in the implicit form and $\boldsymbol{x} = [x, y, z : w]^T$ are point intersection coordinates in the homogeneous coordinates.

Using the determinant notation:

$$\boldsymbol{\rho} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{l} \\ x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3 \end{vmatrix} \quad , \quad \boldsymbol{x} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{l} \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{vmatrix} \tag{17}$$

where $\mathbf{i} = [1, 0, 0, 0]^T$, $\mathbf{j} = [0, 1, 0, 0]^T$, $\mathbf{k} = [0, 0, 1, 0]^T$, $\mathbf{l} = [0, 0, 1, 0]^T$ are the basis vectors.

It should be noted that use of SSE4 instructions or GPU lead to significant speed-up. Also, the presented approach can be used for computation of the barycentric coordinates Skala [40], computation of the Plücker coordinates Skala [41], etc.

VII. NEW GEOMETRIC TRANSFORMATION

General linear transformations are more complex, especially if the *dot product* and *outer-product* (equivalent to the *cross-product* or *skew-product* in the E^3 case) are used.

A. Basic rules

In the case of the cross-product the following identity is valid, see Wiki [42]:

$$(\mathbf{Ma}) \times (\mathbf{Mb}) = det(\mathbf{M})(\mathbf{M}^{-1})^T(\mathbf{a} \times \mathbf{b}) \tag{18}$$

If $det(M) = 1$ for the transformation in the Eq.18 is simplified for the transformation $\mathbf{Q}$ with $det(Q) = 1$ to:

$$(\mathbf{Qa}) \times (\mathbf{Qb}) = \mathbf{Q}\ (\mathbf{a} \times \mathbf{b}) \tag{19}$$

However, for the n-dimensional space and the $outer-product$ applications, more general rules can be derived:

$$\begin{gathered}(\mathbf{Ma}) \wedge (\mathbf{Ma}_2) \wedge \ldots \wedge (\mathbf{Ma}_n) = \\ det(\mathbf{M})^{n-1}(\mathbf{M}^{-1})^T(\mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \ldots \wedge \mathbf{a}_n)\end{gathered} \tag{20}$$

The presented rules are important as it enables to handle geometric transformations with lines, planes and also normal vectors. It should be noted that the normal vector of a plane or triangle is actually a bivector and geometric transformation have to respect Eq.20.

B. Generalized Transformations

Let us consider the geometric basic geometric transformation within the context of the *cross-product* and *dot-product*.

- How transformation matrices for a line are specified, if it is translated, rotated or an-isotropic scaling is made, etc.
- How transformation matrices for a plane are specified, if it is translated, rotated or an-isotropic scaling is made, etc.
- How a normal of a plane, triangle are to be transformed if plane or a triangle is rotated, translated, etc.

To be more specific - what is the result of the following operations with vectors:

$$\begin{aligned} (\mathbf{Ra})(\mathbf{Sb}) &= ? \quad \text{geometric product} \\ (\mathbf{Ra}) \cdot (\mathbf{Sb}) &= ? \quad \text{inner product} \\ (\mathbf{Ra}) \wedge (\mathbf{Sb}) &= ? \quad \text{outer product} \end{aligned} \tag{21}$$

where $\cdot$ is the *dot-product*, i.e. the *scalar-product* and $\wedge$ is the *outer product*, i.e. the *cross-product*.

Let us consider transformation matrices $\mathbf{R}$ and $\mathbf{S}$ as follows:

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{bmatrix} \quad , \quad \mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \end{bmatrix} \tag{22}$$

C. Inner product

The result for the inner product, i.e. dot-product or scalar product, is simple as:

$$\begin{aligned}(\mathbf{Ra}) \cdot (\mathbf{Sb}) = (\mathbf{Ra})^T(\mathbf{Sb}) = \\ \mathbf{a}^T\mathbf{R}^T\mathbf{S}\ \mathbf{b} = \mathbf{a}^T\mathbf{M}\ \mathbf{b} \\ \mathbf{M} = \mathbf{R}^T\mathbf{S}\end{aligned} \tag{23}$$

The square matrix $\mathbf{M}$ represents non-isotropic deformation general.

D. Outer product

The outer product can be rewritten in the matrix form as:

$$(\mathbf{Ra}) \wedge (\mathbf{Sb}) = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \mathbf{r}_1 \cdot \mathbf{a} & \mathbf{r}_2 \cdot \mathbf{a} & \mathbf{r}_3 \cdot \mathbf{a} \\ \mathbf{s}_1 \cdot \mathbf{b} & \mathbf{s}_2 \cdot \mathbf{b} & \mathbf{s}_3 \cdot \mathbf{b} \end{vmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{24}$$

Solving the determinant values of the resulting vector $\mathbf{q} = [q_1, q_2, q_3]^T$ as:

$$q_1 = (\mathbf{r}_2 \cdot \mathbf{a})\ (\mathbf{s}_3 \cdot \mathbf{b}) - (\mathbf{s}_2 \cdot \mathbf{b})\ (\mathbf{r}_3 \cdot \mathbf{a}) \tag{25}$$

$$q_2 = -(\mathbf{r}_1 \cdot \mathbf{a})\ (\mathbf{s}_3 \cdot \mathbf{b}) + (\mathbf{s}_1 \cdot \mathbf{b})\ (\mathbf{r}_3 \cdot \mathbf{a}) \tag{26}$$

$$q_3 = (\mathbf{r}_1 \cdot \mathbf{a})\ (\mathbf{s}_2 \cdot \mathbf{b}) - (\mathbf{s}_1 \cdot \mathbf{b})\ (\mathbf{r}_2 \cdot \mathbf{a}) \tag{27}$$

The expression for the value q_1 can be rewritten using the *tensor-product* as:

$$q_1 = \mathbf{r}_2^T(\mathbf{a} \otimes \mathbf{b})\mathbf{s}_3 - \mathbf{s}_2^T(\mathbf{b} \otimes \mathbf{a})\mathbf{r}_3 \tag{28}$$

Let the matrix $\mathbf{T} = \mathbf{a} \otimes \mathbf{b}$, then $\mathbf{b} \otimes \mathbf{a} = \mathbf{T}^T$. The Eq.25 can be then rewritten as:

$$\begin{aligned}q_1 = \mathbf{r}_2^T\mathbf{T}\ \mathbf{s}_3 - \mathbf{s}_2^T\mathbf{T}^T\mathbf{r}_3 = \\ \mathbf{r}_2^T\mathbf{T}\ \mathbf{s}_3 - \mathbf{s}_3^T\mathbf{T}\ \mathbf{r}_2 =\end{aligned} \tag{29}$$

It should be noted that the q_1 value is a scalar value. This is a useful formula for the case, when the processed element is constant and transformation is changing. However, a typical situation is that the transformation is constant and processed primitives are different. In this case the q_1 can be expressed as:

$$\begin{aligned}q_1 = \mathbf{a}^T(\mathbf{r}_2 \otimes \mathbf{s}_3)\mathbf{b} - \mathbf{a}^T(\mathbf{s}_2 \otimes \mathbf{r}_3)\mathbf{b} = \\ \mathbf{a}^T\mathbf{Q}_1\ \mathbf{b} = \mathbf{a}^T\mathbf{Q}_{23}\ \mathbf{b}\end{aligned} \tag{30}$$

It should be noted, that the notation $\mathbf{Q}_1$ is equivalent to $\mathbf{Q}_{23}$. Similarly for the q_2, q_3 values represented by matrices $\mathbf{Q}_{31}$, $\mathbf{Q}_{12}$.

Application of the above presented approach is intended for the cases, when the transformations $\mathbf{R}$ and $\mathbf{S}$ are constant and lines are processed.

However, the above presented rules are valid in the cases of:

- the three dimensional vector space using the Euclidean space, i.e. $\mathbf{X} = (X, Y, Z)$,
- the projective extension of the two dimensional space using the homogeneous coordinates, i.e. $\boldsymbol{x} = [x, y : w]^T$.

For a higher dimension, the *geometric algebra* is to be used together with the *tensor product*, see Skala [43], using the *geometric product* Eq.13.
The geometric product is represented by the tensor product as:

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

It should be noted, that the resulting matrix contains the elements of the *inner-product* and *outer-product*. Let us consider again transformation matrices $\mathbf{R}$ and $\mathbf{S}$ as follows:

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_n \end{bmatrix} \quad , \quad \mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_n \end{bmatrix} \tag{31}$$

Note, that the row vectors $\mathbf{r}_i$, resp. $\mathbf{s}_i$, are the i-th row of the matrix $\mathbf{R}$, resp. $\mathbf{S}$.
Then the result of the *geometric product* can be represented as:

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b} \quad , \quad \mathbf{ab} \Leftrightarrow \mathbf{a} \otimes \mathbf{b} = \mathbf{a}\ \mathbf{Q}\ \mathbf{b} \tag{32}$$

where the matrix $\mathbf{Q} = \{q_{ij}\}$, $i, j = 1, \ldots, n$, $q_{ij} = \mathbf{e}_i\mathbf{e}_j$ and $\{\mathbf{e}_i\}_{i=1}^n$ are the orthonormal basis vectors in E^n.

In the case of the E^3 space, it should be noted that the matrix $\mathbf{Q}$ has the following combinations of the basis vectors:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{e}_1\mathbf{e}_1 & \mathbf{e}_1\mathbf{e}_2 & \mathbf{e}_1\mathbf{e}_3 \\ \mathbf{e}_2\mathbf{e}_1 & \mathbf{e}_2\mathbf{e}_2 & \mathbf{e}_2\mathbf{e}_3 \\ \mathbf{e}_3\mathbf{e}_1 & \mathbf{e}_3\mathbf{e}_2 & \mathbf{e}_3\mathbf{e}_3 \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{e}_1\mathbf{e}_2 & -\mathbf{e}_3\mathbf{e}_1 \\ -\mathbf{e}_1\mathbf{e}_2 & 1 & \mathbf{e}_2\mathbf{e}_3 \\ \mathbf{e}_3\mathbf{e}_1 & -\mathbf{e}_2\mathbf{e}_3 & 1 \end{bmatrix} \tag{33}$$

It should be noted that in the E^3 case, the right handed coordinate system has the orthonormal basis $\mathbf{e}_1\mathbf{e}_2$, $\mathbf{e}_2\mathbf{e}_3$, $\mathbf{e}_3\mathbf{e}_1$ and therefore the value of q_{13} results into the $-\mathbf{e}_3\mathbf{e}_1$ value.

It means, that the results of the $\mathbf{a} \otimes \mathbf{b}$ operations is:

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1b_1\mathbf{e}_1\mathbf{e}_1 & a_1b_2\mathbf{e}_1\mathbf{e}_2 & -a_1b_3\mathbf{e}_3\mathbf{e}_1 \\ -a_2b_1\mathbf{e}_1\mathbf{e}_2 & a_2b_2\mathbf{e}_2\mathbf{e}_2 & a_2b_3\mathbf{e}_2\mathbf{e}_3 \\ a_3b_1\mathbf{e}_3\mathbf{e}_1 & -a_3b_2\mathbf{e}_2\mathbf{e}_3 & a_3b_3\mathbf{e}_3\mathbf{e}_3 \end{bmatrix} \tag{34}$$

including the right-hand orientation of the coordinate system, resulting into the "-" sign in the matrix.
Note, that $\mathbf{e}_i\mathbf{e}_i = 1$ by definition and therefore:

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1b_1 & a_1b_2\mathbf{e}_1\mathbf{e}_2 & -a_1b_3\mathbf{e}_3\mathbf{e}_1 \\ -a_2b_1\mathbf{e}_1\mathbf{e}_2 & a_2b_2 & a_2b_3\mathbf{e}_2\mathbf{e}_3 \\ a_3b_1\mathbf{e}_3\mathbf{e}_1 & -a_3b_2\mathbf{e}_2\mathbf{e}_3 & a_3b_3 \end{bmatrix} \tag{35}$$

It can be seen, that the diagonal represents the *inner product*, while non-diagonal elements are related to the *outer product*, see AppendixA.

In the general n-dimensional space it means, that:

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

$$\mathbf{a} \wedge \mathbf{b} = \sum_{i,j=1 \& i \neq j}^{n,n} a_i \mathbf{e}_i b_j \mathbf{e}_j = \sum_{i,j=1 \& i > j}^{n,n} (a_i b_j - a_j b_i)\ \mathbf{e}_i \mathbf{e}_j \tag{36}$$

If different transformations $\mathbf{R}$ and $\mathbf{S}$ are applied on the vectors $\mathbf{a}$ and $\mathbf{b}$, then:

$$\begin{aligned} (\mathbf{Ra})(\mathbf{Sb}) &= (\mathbf{Ra}) \cdot (\mathbf{Sb}) + (\mathbf{Ra}) \wedge (\mathbf{Sb}) \\ (\mathbf{Ra})(\mathbf{Sb}) &\Leftrightarrow (\mathbf{Ra}) \otimes (\mathbf{Sb}) = \mathbf{a}\ \mathbf{W}\ \mathbf{b} \end{aligned} \tag{37}$$

where the matrix $\mathbf{W} = \{w_{ij}\}$, $i, j = 1, \dots, n$, $\mathbf{Q}$ is a matrix containing the basis vectors, too.
Note, that the elements w_{ij} of the matrix $\mathbf{W}$ are given as $\mathbf{e}_i \mathbf{e}_j$ not shown explicitly.

$$\begin{aligned} (\mathbf{Ra}) \otimes (\mathbf{Sb}) &= \mathbf{Q} = \mathbf{a}\ \mathbf{W}\ \mathbf{b} \\ q_{ij} = (\mathbf{r_i} \cdot \mathbf{a})\ (\mathbf{s_j} \cdot \mathbf{b}) &= \mathbf{a}\ (\mathbf{r_i} \otimes \mathbf{s_j})\ \mathbf{b} \\ \text{and} \quad w_{ij} &= (\mathbf{r_i} \otimes \mathbf{s_j}) \end{aligned} \tag{38}$$

where $w_{ij} = \mathbf{r_i} \otimes \mathbf{s_j}$, $i, j = 1, \dots, n$.

Using the dual algebraic adjustments using the multilinearity property WiKi [25] the dual formulation is formed as:

$$(\mathbf{Ra}) \otimes (\mathbf{Sb}) = (\mathbf{r_i} \cdot \mathbf{a})\ (\mathbf{s_j} \cdot \mathbf{b}) = \mathbf{r_i}\ (\mathbf{a} \otimes \mathbf{b})\ \mathbf{s_j}$$

$$\begin{aligned} (\mathbf{Ra}) \otimes (\mathbf{Sb}) &= \mathbf{Q} = \mathbf{r}\ \mathbf{W}\ \mathbf{s} \\ q_{ij} = (\mathbf{r_i} \cdot \mathbf{a})\ (\mathbf{s_j} \cdot \mathbf{b}) &= \mathbf{r_i}\ (\mathbf{a} \otimes \mathbf{b})\ \mathbf{s_j} \\ \text{and} \quad w_{ij} &= (\mathbf{a} \otimes \mathbf{b}) \end{aligned} \tag{39}$$

It should be noted that the diagonal of the matrix $\mathbf{W}$ contains elements of the *inner product*. The non-diagonal elements represent parts of bivectors of the given n-dimensional space.

This formulation Eq.39 has an advantage that for the given constant transformations $\mathbf{R}$ and $\mathbf{S}$ the matrix is $\mathbf{W}$ is *constant*.

As the direct impact of the above presented approach can be seen a new formulation of a line p in the E^3 space using the Plücker coordinates and the principle of duality, e.g. a line in E^3 is given by two points in the E^3 space or as an intersections of two planes in E^3. Due to the duality and use of the homogeneous coordinates, it can be represented by one equation, see Skala [3] [44] for details. Library for computations using the projective extension of the Euclidean space was described in Kaiser [45].

It can be seen, that the geometric algebra offers simplification and more general approach to solve geometrical problems, Vince [23].

VIII. Conclusion

In this contribution more general transformation for points, lines, planes have been introduced with a connection to a new *geometric product*, which enables to describe entities generally in the n-dimensional space, including projective extension of the Euclidean space. The transformations are based on the *inner product* and *outer product* represented by a matrix given as a *tensor product* of two vectors.

The proposed method uses *vector-vector* operation and therefore implementation of algorithms using the SSE4 or GPU benefit from an additional speed-up.

Acknowledgement

The author would like to thank colleagues at the University of West Bohemia for hints provided, to students and colleagues at the University of West Bohemia, Plzen and VSB-Technical University, Ostrava for their recent suggestions, to anonymous reviewers for their critical comments and recommendations that improved this paper significantly.

References

[1] F. Yamaguchi, *Computer-Aided Geometric Design*, 1st ed. Tokyo, Japan: Springer-Verlag Tokyo, 2002.
[2] M. Johnson, “Proof by duality: or the discovery of “new” theorems,” *Mathematics Today*, vol. December, pp. 138–153, 1996.
[3] V. Skala, “Extended cross-product and solution of a linear system of equations,” *Lecture Notes in Computer Science*, vol. 9786, pp. 18–35, 2016.
[4] ——, “Modified gaussian elimination without division operations,” *AIP Conference Proceedings*, vol. 1558, pp. 1936–1939, 2013.
[5] ——, “Computation in projective space,” *Proceedings of the 11th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems, MAMECTIS ’09, Proc. 8th WSEAS NOLASC ’09, Proc. 5th WSEAS CONTROL ’09*, pp. 152–157, 2009.
[6] ——, “Length, area and volume computation in homogeneous coordinates,” *International Journal of Image and Graphics*, vol. 6, no. 4, pp. 625–639, 2006.
[7] ——, “Duality and intersection computation in projective space with gpu support,” *International Conference on Applied Mathematics, Simulation, Modelling - Proceedings*, pp. 66–71, 2010.
[8] R. G. Calvet, *Treatise of Plane through Geometric Algebra*, 1st ed. Spain: Cerdanyola del Vallés, 2013.
[9] V. Skala, “Duality, barycentric coordinates and intersection computation in projective space with gpu support,” *WSEAS Transactions on Mathematics*, vol. 9, no. 6, pp. 407–416, 2010.
[10] ——, “Intersection computation in projective space using homogeneous coordinates,” *International Journal of Image and Graphics*, vol. 8, no. 4, pp. 615–628, 2008.
[11] ——, “A new approach to line - sphere and line - quadrics intersection detection and computation,” *AIP Conference Proceedings*, vol. 1648, pp. 1–4, 2015.
[12] ——, “A new approach to line and line segment clipping in homogeneous coordinates,” *Visual Computer*, vol. 21, no. 11, pp. 905–914, 2005.
[13] ——, “Optimized line and line segment clipping in e2 and geometric algebra,” *Annales Mathematicae et Informaticae*, vol. 52, pp. 199–215, 2020.
[14] J. A. Vince, *Geometric Algebra for Computer Graphics*, 1st ed. Santa Clara, CA, USA: Springer-Verlag TELOS, 2008.
[15] J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, J. D. Foley, S. K. Feiner, and K. Akeley, *Computer Graphics - Principles and Practice, 3rd Edition*. Addison-Wesley, 2014.
[16] J. D. Foley, A. van Dam, S. Feiner, and J. F. Hughes, *Computer graphics - principles and practice, 2nd Edition*. Addison-Wesley, 1990.
[17] M. K. Agoston, *Computer Graphics and Geometric Modelling: Mathematics*. Berlin, Heidelberg: Springer-Verlag, 2005.

[18] ——, *Computer Graphics and Geometric Modelling: Implementation & Algorithms*. Berlin, Heidelberg: Springer-Verlag, 2004.
[19] R. S. Ferguson, *Practical Algorithms for 3D Computer Graphics*, 2nd ed. USA: A. K. Peters, Ltd., 2013.
[20] D. Salomon, *Computer Graphics and Geometric Modeling*, 1st ed. Berlin, Heidelberg: Springer-Verlag, 1999.
[21] A. Thomas, *Integrated Graphic and Computer Modelling*, 1st ed. Springer Publishing Company, Incorporated, 2008.
[22] E. Angel and D. Shreiner, *Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL*, 6th ed. USA: Addison-Wesley Publishing Company, 2011.
[23] J. Vince, *Geometric Algebra: An Algebraic System for Computer Games and Animation*, 1st ed. Springer Publishing Company, Incorporated, 2009.
[24] Wikipedia, "Tensor product - Wikipedia, the free encyclopedia," 2021, [Online; accessed 7-October-2021]. [Online]. Available: https://en.wikipedia.org/wiki/Tensor_product
[25] ——, "Multilinear polynomial - Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Multilinear_polynomial
[26] N. Mochizuki, "The tensor product of function algebras," *Tohoku Mathematical Journal*, vol. 17, no. 2, pp. 139–146, 1965.
[27] C. Gunn, "Doing euclidean plane geometry using projective geometric algebra," *Advances in Applied Clifford Algebras*, vol. 27, no. 2, pp. 1203–1232, 2017.
[28] W. S. Massey, "Cross products of vectors in higher dimensional euclidean spaces," *The American Mathematical Monthly*, vol. 90, no. 10, pp. 697–701, 1983. [Online]. Available: http://www.jstor.org/stable/2323537
[29] Z. K. Silagadze, "Multi-dimensional vector product," *Journal of Physics A: Mathematical and General*, vol. 35, no. 23, p. 4949–4953, May 2002.
[30] V. Skala, "Projective geometry and duality for graphics, games and visualization," *SIGGRAPH Asia 2012 Courses, SA 2012*, 2012.
[31] J. Vince, *Introduction to the Mathematics for Computer Graphics*, 3rd ed. Berlin, Heidelberg: Springer-Verlag, 2010, https://link.springer.com/book/10.1007/978-1-4471-6290-2#toc.
[32] A. Macdonald, "A survey of geometric algebra and geometric calculus," *Advances in Applied Clifford Algebras*, vol. 27, no. 1, pp. 853–891, 2017.
[33] C. Doran, A. N. Lasenby, and J. Lasenby, "Conformal geometry, euclidean space and geometric algebra," *CoRR*, vol. cs.CG/0203026, 2002. [Online]. Available: https://arxiv.org/abs/cs/0203026
[34] A. Halma, *Interpolation in Conformal Geometric Algebra: Toward Unified Interpolation of Euclidean Motions in the Conformal Model of Geometric Algebra*, 1st ed. Moldova: Lap Lampert Publ., 2011.
[35] C. Perwass, *Geometric Algebra with Applications in Engineering*, 1st ed. Berlin: Springer-Verlag, 2009.
[36] H. Li, P. J. Olver, and G. Sommer, Eds., *Computer Algebra and Geometric Algebra with Applications*, ser. Lecture Notes in Computer Science, vol. 3519. Springer, 2005.
[37] L. Dorst and J. Lasenby, Eds., *Guide to Geometric Algebra in Practice*. Springer, 2011. [Online]. Available: https://doi.org/10.1007/978-0-85729-811-9
[38] K. Kanatani, *Understanding geometric algebra: Hamilton, Grassmann, and Clifford for computer vision and graphics*. CRC Press, 2015.
[39] D. Hildebrand, *Foundations of Geometric Algebra Computing*, 1st ed. Berlin: Springer-Verlag, 2013.
[40] V. Skala, "Barycentric coordinates computation in homogeneous coordinates," *Computers and Graphics (Pergamon)*, vol. 32, no. 1, pp. 120–127, 2008.
[41] ——, "Plücker coordinates and extended cross product for robust and fast intersection computation," *ACM International Conference Proceeding Series*, vol. 28-June-01-July-2016, pp. 57–60, 2016.
[42] Wikipedia, "Cross product - Wikipedia, the free encyclopedia," 2021, [Online; accessed 10-October-2021]. [Online]. Available: https://en.wikipedia.org/wiki/Cross_product
[43] V. Skala, S. Karim, and E. Kadir, "Scientific computing and computer graphics with gpu: Application of projective geometry and principle of duality," *International Journal of Mathematics and Computer Science*, vol. 15, no. 3, pp. 769–777, 2020.
[44] V. Skala, "Geometry, duality and robust computation in engineering," *WSEAS Transactions on Computers*, vol. 11, no. 9, pp. 275–293, 2012.
[45] V. Skala, J. Kaiser, and V. Ondracka, "Library for computation in the projective space," *APLIMAT 2007*, vol. 2007-January, pp. 125–130, 2007.

APPENDIX

The outer product $\mathbf{a} \wedge \mathbf{b}$ (cross-product $\mathbf{a} \times \mathbf{b}$) in E^3 can be also represented in a matrix form as:

$$\begin{gathered} \mathbf{a} \wedge \mathbf{b} = \mathbf{a} \times \mathbf{b} = \\ \lfloor \mathbf{a}_x \rfloor \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \mathbf{b} = \\ \lfloor \mathbf{b}_x^T \rfloor \mathbf{a} = \begin{bmatrix} 0 & b_3 & -b_2 \\ -b_3 & 0 & b_1 \\ b_2 & -b_1 & 0 \end{bmatrix} \mathbf{a} \end{gathered} \tag{40}$$

It should be noted that the matrix $\lfloor \mathbf{a}_x \rfloor$ is actually transposed as it is an operator applied on a vector $\mathbf{b}$.