IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 14, 2021, accepted July 7, 2021, date of publication July 16, 2021, date of current version July 30, 2021.
*Digital Object Identifier 10.1109/ACCESS.2021.3097756*

# A Survey on Quaternion Algebra and Geometric Algebra Applications in Engineering and Computer Science 1995–2020

**EDUARDO BAYRO-CORROCHANO, (Senior Member, IEEE)**
Department of Electrical Engineering and Computer Science, CINVESTAV Guadalajara, Guadalajara Campus, Zapopan, Jalisco 45119, Mexico
e-mail: eduardo.bayro@cinvestav.mx

This work was supported by the Conacyt Project under Grant A1-S-10412.

**ABSTRACT** Geometric Algebra (GA) has proven to be an advanced language for mathematics, physics, computer science, and engineering. This review presents a comprehensive study of works on Quaternion Algebra and GA applications in computer science and engineering from 1995 to 2020. After a brief introduction of GA, the applications of GA are reviewed across many fields. We discuss the characteristics of the applications of GA to various problems of computer science and engineering. In addition, the challenges and prospects of various applications proposed by many researchers are analyzed. We analyze the developments using GA in image processing, computer vision, neurocomputing, quantum computing, robot modeling, control, and tracking, as well as improvement of computer hardware performance. We believe that up to now GA has proven to be a powerful geometric language for a variety of applications. Furthermore, there is evidence that this is the appropriate geometric language to tackle a variety of existing problems and that consequently, step-by-step GA-based algorithms should continue to be further developed. We also believe that this extensive review will guide and encourage researchers to continue the advancement of geometric computing for intelligent machines.

**INDEX TERMS** Geometric algebra, Clifford algebra, quaternion algebra, screw theory, signal processing, electrical engineering and power systems, geometric and quantum computing, image processing, computer vision, graphic engineering, artificial intelligence, machine learning, neural networks, control engineering, robotics, biomedical engineering, and biotechnology.

## I. INTRODUCTION

The recent reviews "Applications of Clifford's Geometric Algebra" by E. Hitzer, T. Nitta and Y. Kuroe [135], "Geometric Algebra in Signal and Image Processing: A survey" by R. Wang, K. Wang, W.Cao and X.Wang [233] and complementary "A survey of quaternion neural networks" by T. Parcollet, M. Morchid, M. and G. Linarés [194] are very useful to understand the progress in the area of geometric algebra. Bhatti *et al.* [52] provide a detailed review of GA in different fields of AI and computer vision regarding its applications and the current developments in geospatial research. In our survey, we extend and complete these reviews discussing works in several areas of computer science and engineering from 1995 to 2020.

The associate editor coordinating the review of this manuscript and approving it for publication was Sunil Karamchandani.

Since 1995, Eduardo Bayro-Corrochano and Joan Lasenby pioneered in the application of geometric algebra to computer vision [14]–[18], [152], [153], [153], [155]–[157], [213], Bayro *et al.* in robotics [15], [19], [33], [74] and Bayro *et al.* in neurocomputing [20]–[24]. In the beginning, their articles were hardly accepted and not understood as they proposed a new geometric approach using exclusively the geometric algebra framework which was fairly unknown by the community and completely different from the usual methods based on matrix algebra, vector calculus, and tensor algebra. Thanks to their persistence in solving challenging problems in those fields, geometric algebra was recognized and many researchers worldwide started using and developing new algorithms for real-time applications.

## II. GEOMETRIC ALGEBRA

In the 1870s, William Kingdon Clifford introduced his geometric algebra, building on earlier works of Sir William

Rowan Hamilton and Hermann Gunther Grassmann. Clifford intended to describe the geometric properties of vectors, planes, and higher-dimensional objects.

Most physicists encounter the algebra in the guise of Pauli and Dirac matrix algebras of quantum theory. Many roboticists or computer graphic engineers use quaternions for 3D rotation estimation and interpolation, as it is too difficult for them to formulate homogeneous transformations of high-order geometric entities using a point-wise approach. They resort often to tensor calculus for multi-variable calculus. Since robotics and engineering make use of the developments of mathematical physics, many beliefs are automatically inherited; for instance, some physicists come away from a study of Dirac theory with the view that Clifford's algebra is inherently quantum-mechanical.

For the last two decades, researchers have been using different geometric algebras $G_{p,q,r}$ depending upon the problem in question. It is very important to formulate the problem using a geometric algebra framework with the correct metric. Next, we will describe which geometric algebras are suitable to tackle different problems in Engineering and Computer Science.

The geometric algebra is a geometric interpretation of Clifford algebras, so Clifford algebra is not the same as geometric algebra, the reader should resort to [53] for an introduction to Clifford algebras. The section on geometric algebras explains most of the concepts of Clifford algebra, however in a geometric way for example to model the geometric entities and the Lie groups.

In subsection IV-A, we outline briefly the fundamentals of the most relevant geometric algebras mentioned above which were used mostly in the works of the last 25 years.

### A. THE GEOMETRIC ALGEBRA

Geometric algebra (GA), a geometric interpretation of Clifford algebra [53], is a coordinate-free approach to geometry based on the algebra of Clifford and Grassmann. Suppose a $n$-dimensional space $\mathbb{R}^n$, equipped with an orthonormal basis vectors $\{e_i\}$, $i = 1, \ldots, n$; such that $e_i \cdot e_j = \delta_{i,j}$. The previously mentioned vectors lead to the multivector basis for the entire geometric algebra

$$\{1; e_i; e_i \wedge e_j; e_i \wedge e_j \wedge e_k; \cdots ; I = e_1 \wedge \cdots \wedge e_n\} \quad (1)$$

where $I$ is known as the pseudoscalar. The geometric algebra of a $n$-dimensional space $\mathbb{R}^n$ is denoted by $G_n$, but alternatively, it can be written by $G_{p+q+r}$. Here, $p$, $q$ and $r$ represent the number of unit orthonormal basis vectors that square to 1, $-1$ and 0, respectively. For simplicity, if $q$ or $r$ are zero, we will not denote them, e.g. $G_3$, $G_{3,1}$, $G_{4,1}$ except for motor algebra $G^+_{3,0,1}$. Since for a $n$-dimensional space $\mathbb{R}^n$ the correspondent $G_n$ has $2^n$ multivector bases, thus $G_n \in \mathbb{R}^{2n}$.

Additionally to scalar multiplication and vector addition, $G_n$ contemplates a noncommutative product that is associative and distributive, this product is the geometric or Clifford product and for vectors $a, b$ can be expressed as:

$$ab = a \cdot b + a \wedge b \quad (2)$$

The right part of (2) shows two operators, the first is the inner product (symmetric part) and it is common in vector calculus; the second is the exterior product or wedge product (antisymmetric part), where the exterior product is an associative, distributive and anticommutative multiplication with multilinear functions. The elements produced by the geometric product of $k$ linear independent vectors span the $k - vector$ space, and it is expressed by $\bigwedge^k V^n$. Each element of this space is named $k - vector$ and it is written as $\langle A \rangle_k$, where $k$ indicates the grade. The linear combination of $k$-vectors is called a multivector and it is denoted as follows:

$$A = \langle A \rangle_1 + \langle A \rangle_2 + \ldots + \langle A \rangle_n \quad (3)$$

In addition to (2), the geometric product of two multivectors $A_r$ and $B_s$, where $r$ and $s$ stand for the grade of each multivector, is:

$$A_r B_s = \langle AB \rangle_{r+s} + \langle A \rangle_{r+s-2} + \ldots + \langle A \rangle_{r-s}, \quad (4)$$

where $\langle A_r B_s \rangle_t$, denotes the t-grade part of the multivector $A_r B_s$, for more details see [41].

### B. 3D GEOMETRIC ALGEBRA FOR THE EUCLIDEAN 3D SPACE

For the case of the Euclidean 3D space $\mathbb{R}^3$, we choose the geometric algebra $G_3$, which has $2^3 = 8$ elements given by

$$\underbrace{1}_{scalar}, \quad \underbrace{\{e_1, e_2, e_3\}}_{vectors}, \quad \underbrace{\{e_2e_3, e_3e_1, e_1e_2\}}_{bivectors}, \quad \underbrace{\{e_1e_2e_3\} \equiv I}_{trivector}.$$

The highest-grade algebraic element for the 3D space is a trivector called pseudoscalar $I \equiv e_1e_2e_3$, which squares to $-1$ and which commutes with the scalars and bivectors in the 3D space. In the algebra of three-dimensional space we can construct a trivector $\boldsymbol{a} \wedge \boldsymbol{b} \wedge \boldsymbol{c} = \lambda I$, where the vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are in general position and $\lambda \in \mathbb{R}$. Note that no 4-vectors exist since there is no possibility of sweeping the volume element $\boldsymbol{a} \wedge \boldsymbol{b} \wedge \boldsymbol{c}$ over a fourth dimension.

Multiplication of the three basis vectors $e_1$, $e_2$, and $e_3$ by $I$ results in the three basis bivectors $e_2e_3 = Ie_1, e_3e_1 = Ie_2$, and $e_1e_2 = Ie_3$. These simple bivectors rotate vectors in their own plane by 90°, for example, $(e_1e_2)e_2 = e_1$, $(e_2e_3)e_2 = -e_3$, etc. Identifying the unit vectors $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ of quaternion algebra with $Ie_1, -Ie_2, Ie_3$ allows us to write the famous Hamilton relations $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$. Since the $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are really bivectors, it comes as no surprise that they represent 90° rotations in orthogonal directions and provide a system well suited for the representation of general 3D rotations. Rotors are isomorphic with quaternions. Be aware, the quaternion and rotor follow the left-hand and the right-hand rotation rule respectively.

In geometric algebra a *rotor* (short name for rotator), $\boldsymbol{R}$, is an even-grade element of the Euclidean algebra of 3D-space. If Q= $\{r_0, r_1, r_2, r_3\} \in G_3$ represents a unit quaternion, then the rotor which performs the same rotation is simply given by

$$\boldsymbol{R} = \underbrace{r_0}_{scalar} + \underbrace{r_1 e_{2,3} - r_2 e_{3,1} + r_3 e_{1,2}}_{bivectors}. \tag{5}$$

The rotor algebra $G_3^+$ is therefore a subset of the Euclidean geometric algebra of three-dimensional space.

Consider in $G_3$ two nonparallel vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ which are referred to the same origin. In general, a rotation operation of a vector $\boldsymbol{a}$ toward the vector $\boldsymbol{b}$ can be performed by two *reflections*, respective to the unit vector axes $\boldsymbol{n}$ and $\boldsymbol{m}$. Thus, the unit rotor can be computed as the geometric product of two unit vectors,

$$\boldsymbol{R} = \boldsymbol{mn} = \boldsymbol{m}\cdot\boldsymbol{n} + \boldsymbol{m}\wedge\boldsymbol{n}. \tag{6}$$

The components of equation (6) correspond to the scalar and bivector terms of an equivalent quaternion in $G_3$, and thus $\boldsymbol{R} \in G_3^+$. This even subalgebra corresponds to the algebra of rotors.

Considering the scalar and the bivector terms of the rotor of equation (6), we can further write the Euler representation of a 3D rotation with angle $\theta$ in the left-hand sense, as follows:

$$\boldsymbol{R} = e^{\frac{\theta}{2}\bar{\boldsymbol{r}}_n} = cos(\frac{\theta}{2}) + sin(\frac{\theta}{2})\bar{\boldsymbol{r}}_n, \tag{7}$$

where $\bar{\boldsymbol{r}}_n$ is the unitary rotation axis-vector spanned by the bivector basis $e_2e_3$, $e_3e_1$, and $e_1e_2$. The polar representation of a rotor given in equation (7) is possible, because the rotor as a Lie group can be expressed in terms of the Lie algebra of bivectors: The orbits on the Lie group manifold describe the evolution of the actions of rotors. The bivector $\bar{\boldsymbol{r}}_n$ corresponds to the Lie algebra operator tangent to an orbit or geodesic.

A rotor is isomorph with a quaternion. As a result, we can embed quaternions in the more comprehensive mathematical system offered by geometric algebra. In contrast to the quaternion theory, in geometric algebra, the quaternions or rotors have a clear geometric interpretation due to the spatial representation of the rotations which can be described by using reflections with respect to planes. The next subsection is devoted to quaternion algebra including more details useful for applications in image processing.

### C. QUATERNION ALGEBRA

This survey includes various articles which use quaternion algebra $\mathbb{H}$. We review those works and compare them with those which use geometric algebra, particularly because quaternion algebra $\mathbb{H}$ is isomorph to the subalgebra of rotors $G_3^+$. Rotors have a geometric interpretation and as rotor algebra can be embedded in high-order geometric algebras to deal with more complex Lie groups. Example rotors can be related with translators for SE(3) in motor algebra $G_{3,0,1}^+$ which is isomorph to dual quaternions or in conformal geometric algebra $G_{4,1}$. So the researchers have a much-generalized way to use SO(3) for rotors as with quaternions.

As a result, the readers can see for their future works they can use rotors instead of quaternions.

The even geometric subalgebra or rotor algebra $G_3^+$ (bivector basis) is isomorphic to the quaternion algebra $\mathbb{H}$, which is an associative, non-commutative, four-dimensional algebra that consists of one real element and three imaginary elements.

$$q = a + bi + cj + dk, \quad a, b, c, d \in \mathbb{R} \tag{8}$$

The units $i$, $j$ and $k$ obey the relations $i^2 = j^2 = -1$, $ij = k$. The imaginary elements of $\mathbb{H}$ are related to the bivector basis of $G_3^+$ as follows $i \to e_{23}, j \to e_{13}, k \to e_{12}$, where $e_{23}, e_{13}, e_{12}$ are the bivector bases. Another important property of $\mathbb{H}$ is the phase concept. A polar representation of $q$ is

$$q = |q|e^{i\phi}e^{k\psi}e^{j\theta}, \tag{9}$$

where $|q| = \sqrt{q\bar{q}}$ where $\bar{q}$ is a conjugate of $q = a - bi - cj - dk$ and the angles $(\phi, \theta, \psi)$ represent the three quaternionic phases.

The quaternion product $q_a q_b$ is equivalent with the geometric product of $G_3^+$. Given two quaternions $q_a = a + \boldsymbol{a} = a + a_x i + b_y j + a_z k$ and $q_b = b + \boldsymbol{b} = b + b_x i + b_y j + b_z k$ where $a, a_x, a_y, a_z, b, b_x, b_y, b_z \in \mathbb{R}$, their quaternion product is given by

$$\begin{aligned} q_c &= c + \boldsymbol{c} = q_a q_b \\ &= (ab - \boldsymbol{a}\cdot\boldsymbol{b}) + (a\boldsymbol{b} + b\boldsymbol{a} + \boldsymbol{a}\times\boldsymbol{b}), \end{aligned} \tag{10}$$

where $\cdot$ and $\times$ are vector inner and cross product respectively.

The rotations in $\mathbb{R}^3$ are expressed by the subgroup $SO(3)$. Where in quaternions, the rotor operator is illustrated as

$$q = cos\frac{\theta}{2} + sin\frac{\theta}{2}\boldsymbol{v}, \tag{11}$$

where $\theta$ is the rotation angle and $\boldsymbol{v}$ is an arbitrary axis of rotation expanded as $\boldsymbol{v} = v_x i + v_y j + v_z k$. On the other hand, the conjugate of the rotor is:

$$\widetilde{q} = cos\frac{\theta}{2} - sin\frac{\theta}{2}\boldsymbol{v}. \tag{12}$$

Given 3D point $\boldsymbol{p}$ expressed as the imaginary part of a quaternion $\boldsymbol{p} = x_x i + x_y j + x_z k$, the rotation of $\boldsymbol{p}$ around an arbitrary axis $\boldsymbol{v}$ with angle $\theta$ can be computed as follows

$$\boldsymbol{y} = y_x i + y_y j + y_z k = qp\widetilde{q}. \tag{13}$$

### D. MOTOR ALGEBRA

Usually, problems of robotics are treated in algebraic systems of 2D and 3D space. In the case of 3D rigid motion or Euclidean transformation, we are confronted with a nonlinear mapping. However, if we employ homogeneous coordinates in 4D geometric algebra we can linearize the rigid motion in 3D Euclidean space. That is why we choose three basis vectors which square to one and a fourth vector which squares to zero—to provide dual copies of the multivectors of the 3D space. In other words, we extend the Euclidean geometric algebra $G_3$ to the special or degenerated geometric algebra $G_{3,0,1}^+$. The word *motor* is an abbreviation of "moment and vector." Clifford introduced motors with the name bi-quaternions [68]. Motors are isomorphic to dual quaternions, with the necessary condition $I^2 = 0$. They can be found in the special 4D even subalgebra of $G_{3,0,1}$. This even subalgebra is denominated by $G^+_{3,0,1}$ and is only spanned via a bivector basis, as follows:

$$\underbrace{1}_{scalar}, \underbrace{e_2e_3, e_3e_1, e_1e_2, e_4e_1, e_4e_2, e_4e_3}_{6 bivectors}, \underbrace{I}_{unit\ pseudoscalar}$$

This kind of basis structure also allows us to represent spinors, which are composed of scalar and bivector terms. Motors, then, are also spinors. As such, they represent a special kind of rotor, because a Euclidean transformation includes both rotation and translation.

Since a rigid motion consists of the rotation and translation transformations, it should be possible to split a motor multiplicatively in terms of these two spinor transformations, which we will call a rotor and a *translator*. In the following discussion, we will denote all bivector components of a spinor by slant bold lowercase letters and with slant bold uppercase letters: screw lines $\boldsymbol{L}$, rotors $\boldsymbol{R}$, translators $\boldsymbol{T}$ and motors $\boldsymbol{M}$. Let us now express this procedure algebraically. First of all, let us consider a simple rotor in its *Euler representation* for a rotation with an angle $\theta$,

$$\boldsymbol{R} = a_0 + a_1e_2e_3 + a_2e_3e_1 + a_3e_1e_2 = a_c + a_s\boldsymbol{n}, \quad (14)$$

where $\boldsymbol{n}$ is the unit 3D bivector of the rotation axis spanned by the bivector basis $e_2e_3$, $e_3e_1$, $e_1e_2$, and $a_c$, $a_s \in \mathbb{R}$. Now, dealing with the rotor of a screw motion, the rotation-axis vector should be represented as a screw-axis line. For that, we must relate the rotation axis to a reference coordinate system at the distance $\boldsymbol{t}_c$. A 3D translation in motor algebra is represented by a spinor $\boldsymbol{T}_c$ called a translator. If we apply a translator from the left to rotor $\boldsymbol{R}$, and then apply the translator's conjugate from the right, we get a modified rotor,

$$\boldsymbol{R}_s = \boldsymbol{T}_c\boldsymbol{R}\widetilde{\boldsymbol{T}}_c = a_0 + \boldsymbol{a} + I(\boldsymbol{a} \times \boldsymbol{t}_c). \quad (15)$$

where $\boldsymbol{t}_c$ is the 3D vector of translation spanned by the vector basis $e_1$, $e_2$, $e_3$. Then, expressing the last equation in Euler terms, we get the spinor representation,

$$\boldsymbol{R}_s = a_0 + a_s\boldsymbol{n} + Ia_s\boldsymbol{n}\wedge\boldsymbol{t}_c == cos(\frac{\theta}{2}) + sin(\frac{\theta}{2})\boldsymbol{l}. \quad (16)$$

This result is interesting because the new rotor $\boldsymbol{R}_s$ can now be applied with respect to an axis line $\boldsymbol{l} = \boldsymbol{n} + I\boldsymbol{m}$ expressed in dual terms of direction $\boldsymbol{n}$ and moment $\boldsymbol{m} = \boldsymbol{n}\wedge\boldsymbol{t}_c$. Now, to define the motor finally, let us slide the distance $\boldsymbol{t}_s = d\boldsymbol{n}$ along the rotation-axis line $\boldsymbol{l}$. Since a motor is applied from the left and conjugated from the right, we should use the half of $\boldsymbol{t}_s$ in the spinor expression of $\boldsymbol{T}_s$ when we define the motor:

$$\boldsymbol{M} = \boldsymbol{T}_s\boldsymbol{R}_s = (a_c - Ia_s\frac{d}{2}) + (a_s + Ia_c\frac{d}{2})\boldsymbol{l}. \quad (17)$$

Note that this expression of the motor makes explicit the unit line bivector of the screw-axis line $\boldsymbol{l}$.

Now let us express a motor using Euler representation. By substituting the constants $a_c = cos(\frac{\theta}{2})$ and $a_s = sin(\frac{\theta}{2})$ in the motor equation (17), we get

$$\boldsymbol{M} = \boldsymbol{T}_s\boldsymbol{R}_s = cos(\frac{\theta}{2} + I\frac{d}{2}) + sin(\frac{\theta}{2} + I\frac{d}{2})\boldsymbol{l}, \quad (18)$$

which is a dual-number representation of the spinor. Now, let us analyze the resultant expressions,

$$\begin{aligned} \boldsymbol{R} &= cos(\frac{\theta}{2}) + sin(\frac{\theta}{2})\boldsymbol{n} \\ \boldsymbol{R}_s &= cos(\frac{\theta}{2}) + sin(\frac{\theta}{2})\boldsymbol{l} \\ \boldsymbol{M} &= cos(\frac{\theta}{2} + I\frac{d}{2}) + sin(\frac{\theta}{2} + I\frac{d}{2})\boldsymbol{l}. \quad (19) \end{aligned}$$

We can see that the rotation axis $\boldsymbol{n}$ of the simple rotor $\boldsymbol{R}$ is changed to a rotation-axis line, so that $\boldsymbol{R}_s$ now rotates about an axis line $\boldsymbol{l}$. And in the motor expression, the information for sliding distance $d$ is now made explicit in terms of dual arguments of the trigonometric functions. It is also interesting to note that the expression for the motor using dual angles simply extends the expression of $\boldsymbol{R}_s$.

If we expand the exponential function of the dual bivectors using a Taylor series, the result will follow the general expression $e^{\alpha+I\beta} = e^{\alpha} + Ie^{\alpha}\beta = e^{\alpha}(1 + I\beta)$. Once again, we obtain the motor expression as the spinor

$$e^{\boldsymbol{l}\frac{\theta}{2}+I\frac{\boldsymbol{t}_s}{2}} = (1 + I\frac{\boldsymbol{t}_s}{2})e^{\boldsymbol{l}\frac{\theta}{2}} = \boldsymbol{T}_s\boldsymbol{R}_s = \boldsymbol{M}, \quad (20)$$

where $I\frac{\boldsymbol{t}_s}{2} = I\frac{1}{2}(t_1e_2e_3 + t_2e_3e_1 + t_3e_1e_2) = \frac{1}{2}(t_1e_4e_1 + t_2e_4e_2 + t_3e_4e_3)$.

If we want to express the motor using only rotors in a dual spinor representation, we proceed as follows:

$$\boldsymbol{M} = \boldsymbol{T}_s\boldsymbol{R}_s = (1 + I\frac{\boldsymbol{t}_s}{2})\mathbf{R}_s = \boldsymbol{R}_s + I\boldsymbol{R}'_s. \quad (21)$$

### *E. THE GEOMETRIC ALGEBRAS OF 3D AND 4D SPACES FOR COMPUTER VISION*

For the modeling of the image plane, we use $G_3$, which has the standard *Euclidean signature*. We will show that if we choose to map between projective space $\mathbb{P}^3$ and 3D Euclidean space or projective plane $\mathbb{P}^2$ via the projective split, we are then forced to use the 4D geometric algebra $G_{3,1}$ for the projective space.

The Lorentzian 4D algebra $G_{3,1}$ has a Minkowski metric with $e_i^2 = +1$ and for $i = 1, 2, 3$ $e_4^2 = -1$ and generates the following multivector basis:

$$\underbrace{1}_{scalar}, \underbrace{e_k}_{4vectors}, \underbrace{e_2e_3, e_3e_1, e_1e_2, e_4e_1, e_4e_2, e_4e_3}_{6bivectors}, \quad (22)$$

$$\underbrace{Ie_k}_{4trivectors}, \underbrace{I}_{pseudoscalar}. \quad (23)$$

The pseudoscalar is $I = e_1e_2e_3e_4$, with

$$I^2 = (e_1e_2e_3e_4)(e_1e_2e_3e_4) = -(e_3e_4)(e_3e_4) = -1. \quad (24)$$

The fourth basis vector, $e_4$, can also be seen as a selected direction for the *projective split*. When we use homogeneous coordinates, a general point in $\mathbb{E}^3$ given by $\boldsymbol{x} = xe_1+ye_2+ze_3$ becomes the point $\mathbf{X} = (Xe_1 + Ye_2 + Ze_3 + We_4)$ in $R^4$, where $x = X/W$, $y = Y/W$ and $z = Z/W$. Now, using $\underline{f_p}$, the linear map of $\mathbf{X}$ onto $\mathbf{X}'$ is given by

$$\mathbf{X}' = \sum_{i=1}^{3}\{(\alpha_i X + \beta_i Y + \delta_i Z + \epsilon_i W)e_i\} + (\tilde{\alpha} X + \tilde{\beta} Y + \tilde{\delta} Z + \tilde{\epsilon} W)e_4. \tag{25}$$

The coordinates of the vector in the image plane $\mathbb{P}^2$, $\boldsymbol{x}' = x'e_1 + y'e_2 + z'e_3 \in G_3$ which correspond to $\mathbf{X}'$ are given by

$$x' = \frac{\alpha_1 X + \beta_1 Y + \delta_1 Z + \epsilon_1 W}{\tilde{\alpha} X + \tilde{\beta} Y + \tilde{\delta} Z + \tilde{\epsilon} W} = \frac{\alpha_1 x + \beta_1 y + \delta_1 z + \epsilon_1}{\tilde{\alpha} x + \tilde{\beta} y + \tilde{\delta} z + \tilde{\epsilon}},$$
$$y' = \frac{\alpha_2 x + \beta_2 y + \delta_2 z + \epsilon_2}{\tilde{\alpha} x + \tilde{\beta} y + \tilde{\delta} z + \tilde{\epsilon}}, \quad z' = \frac{\alpha_3 x + \beta_3 y + \delta_3 z + \epsilon_3}{\tilde{\alpha} x + \tilde{\beta} y + \tilde{\delta} z + \tilde{\epsilon}}.$$

### 1) INCIDENCE ALGEBRA

In $G_{3,1}$, consider the line $L = \mathbf{X}_1 \wedge \mathbf{X}_2$ intersecting the plane $\Phi = \mathbf{Y}_1 \wedge \mathbf{Y}_2 \wedge \mathbf{Y}_3$. We can compute the intersection point $\boldsymbol{X}$ using a *meet* operation, as follows:

$$L \cap \Phi = (\mathbf{X}_1 \wedge \mathbf{X}_2) \cap (\mathbf{Y}_1 \wedge \mathbf{Y}_2 \wedge \mathbf{Y}_3) = L \cap \Phi = L^* \cdot \Phi, \tag{26}$$

where $\cap$ stands for the meet operator for the intersection and $L^*$ is the dual of $L$ and the vectors $\boldsymbol{X}_i$, $\boldsymbol{Y}_i \in G_{3,1}$. Since,

$$L^* \cdot \Phi = (LI^{-1}) \cdot \Phi = -(LI) \cdot \Phi, \tag{27}$$

we can expand the meet to compute the intersecting point $\boldsymbol{X} \in G_{3,1}$

$$\begin{aligned} L \cap \Phi &= -(LI) \cdot (\mathbf{Y}_1 \wedge \mathbf{Y}_2 \wedge \mathbf{Y}_3) \\ &= -\{(LI) \cdot (\mathbf{Y}_2 \wedge \mathbf{Y}_3)\}\mathbf{Y}_1 + \{(LI) \cdot (\mathbf{Y}_3 \wedge \mathbf{Y}_1)\}\mathbf{Y}_2 \\ &\quad + \{(LI) \cdot (\mathbf{Y}_1 \wedge \mathbf{Y}_2)\}\mathbf{Y}_3. \\ \boldsymbol{X} &= [\mathbf{X}_1\mathbf{X}_2\mathbf{Y}_2\mathbf{Y}_3]\mathbf{Y}_1 + [\mathbf{X}_1\mathbf{X}_2\mathbf{Y}_3\mathbf{Y}_1]\mathbf{Y}_2 + [\mathbf{X}_1\mathbf{X}_2\mathbf{Y}_1\mathbf{Y}_2]\mathbf{Y}_3. \end{aligned}$$

The line of intersection of two planes, $\Phi_1 = \mathbf{X}_1 \wedge \mathbf{X}_2 \wedge \mathbf{X}_3$ and $\Phi_2 = \mathbf{Y}_1 \wedge \mathbf{Y}_2 \wedge \mathbf{Y}_3$, can be computed via the meet of $\Phi_1$ and $\Phi_2$ as follows

$$\Phi_1 \cap \Phi_2 = (\mathbf{X}_1 \wedge \mathbf{X}_2 \wedge \mathbf{X}_3) \cap (\mathbf{Y}_1 \wedge \mathbf{Y}_2 \wedge \mathbf{Y}_3). \tag{28}$$

This expression can be expanded as

$$\begin{aligned} \Phi_1 \cap \Phi_2 &= \Phi_1{}^* \cdot (\mathbf{Y}_1 \wedge \mathbf{Y}_2 \wedge \mathbf{Y}_3) \\ &= -\{(\Phi_1 I) \cdot \mathbf{Y}_1\}(\mathbf{Y}_2 \wedge \mathbf{Y}_3) + \{(\Phi_1 I) \cdot \mathbf{Y}_2\}(\mathbf{Y}_3 \wedge \mathbf{Y}_1) \\ &\quad + \{(\Phi_1 I) \cdot \mathbf{Y}_3\}(\mathbf{Y}_1 \wedge \mathbf{Y}_2). \end{aligned} \tag{29}$$

Once again, the join covers the entire space and so the dual is easily formed. We can show that $(\Phi_1 I) \cdot \mathbf{Y}_i \equiv -[\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Y}_i]$, so that the meet yields the intersection line $\boldsymbol{L} \in G_{3,1}$ or bivector.

$$\begin{aligned} \boldsymbol{L} = \Phi_1 \cap \Phi_2 &= [\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Y}_1](\mathbf{Y}_2 \wedge \mathbf{Y}_3) \\ &\quad + [\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Y}_2](\mathbf{Y}_3 \wedge \mathbf{Y}_1) + [\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Y}_3](\mathbf{Y}_1 \wedge \mathbf{Y}_2). \end{aligned} \tag{30}$$

### 2) PROJECTIVE GEOMETRY AND COMPUTER VISION

The algebraic computing of geometric entities is carried out using the Incidence Algebra which is formulated using the frameworks of the geometric algebras $G_{3,1}$ and $G_3$.

The projection of any world point $\mathbf{X}$ onto the image plane is notated $\boldsymbol{x}$ and is given by the intersection of line $\mathbf{A}_0 \wedge \mathbf{X}$ with the plane $\Phi_A$. Thus,

$$\boldsymbol{x} = (\mathbf{A}_0 \wedge \mathbf{X}) \cap (\mathbf{A}_1 \wedge \mathbf{A}_2 \wedge \mathbf{A}_3). \tag{31}$$

We can now expand the meet to get

$$\boldsymbol{x} = X_j\{[\mathbf{A}_0 \wedge e_j \wedge \mathbf{A}_2 \wedge \mathbf{A}_3]\mathbf{A}_1 + [\mathbf{A}_0 \wedge e_j \wedge \mathbf{A}_3 \wedge \mathbf{A}_1]\mathbf{A}_2 + [\mathbf{A}_0 \wedge e_j \wedge \mathbf{A}_1 \wedge \mathbf{A}_2]\mathbf{A}_3\}. \tag{32}$$

Since $\boldsymbol{x} = x^k\mathbf{A}_k$, the previous equation implies that $\boldsymbol{x} = X_j P_{jk} \mathbf{A}_k$ and therefore that

$$x^k = P_{jk} X_j,$$

where

$$P_{jk} = [\mathbf{A}_0 \wedge e_j \wedge L_k^A] \equiv [\phi_k \wedge e_j] = -t_{kj}, \tag{33}$$

since $Ie_j \wedge e_k = -I\delta_{jk}$. The matrix $P$ takes $\mathbf{X}$ to $\boldsymbol{x}$ and is therefore the standard camera projection matrix. If we define a set of vectors $\{\phi_A^j\}$, $j = 1, 2, 3$, which are the duals of the planes $\{\phi_j^A\}$—that is, $\phi_A^j = \phi_j^A I^{-1}$—it is then easy to see that

$$\phi_A^j = -\phi_j^A I = -[t_{j1}e_i + t_{j2}e_j + t_{j3}e_k + t_{j4}e_l]. \tag{34}$$

Thus, we see that the projected point $\boldsymbol{x} = x^j\mathbf{A}_j$ may be given by

$$x^j = \mathbf{X} \cdot \phi_A^j \qquad \text{or} \qquad \boldsymbol{x} = (\mathbf{X} \cdot \phi_A^j)\mathbf{A}_j. \tag{35}$$

That is, the coefficients in the image plane are formed by projecting $\mathbf{X}$ onto the vectors formed by taking the duals of the optical planes. This is, of course, equivalent to the matrix formulation

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \phi_A^1 \\ \phi_A^2 \\ \phi_A^3 \end{bmatrix} X. \tag{36}$$

The elements of the camera matrix are therefore simply the coefficients of each optical plane in the coordinate frame of the world point. They encode the intrinsic and extrinsic camera parameters.

Next, we consider the projection of world lines of the projective space $\mathbb{P}^3$ onto the image plane $\mathbb{P}^2$. Suppose we have a world line $L = \mathbf{X}_1 \wedge \mathbf{X}_2$ joining the points $\mathbf{X}_1$ and $\mathbf{X}_2$. If $\boldsymbol{x}_1 = (\mathbf{A}_0 \wedge \mathbf{X}_1) \cap \Phi_A$ and $\boldsymbol{x}_2 = (\mathbf{A}_0 \wedge \mathbf{X}_2) \cap \Phi_A$ (i.e., the intersections of the optical rays with the image plane), then the projected line in the image plane is clearly given by

$$l = \boldsymbol{x}_1 \wedge \boldsymbol{x}_2.$$

Since we can express $l$ in the bivector basis for the plane, we obtain

$$l = l^j L_j^A,$$

where $L_1^A = \mathbf{A}_2 \wedge \mathbf{A}_3$, $L_2^A = \mathbf{A}_3 \wedge \mathbf{A}_1$, $L_3^A = \mathbf{A}_1 \wedge \mathbf{A}_2$. From our previous expressions for projections given in equation (35), we see that we can also write $l$ as follows:

$$l = \boldsymbol{x}_1 \wedge \boldsymbol{x}_2 = (\mathbf{X}_1 \cdot \phi_A^j)(\mathbf{X}_2 \cdot \phi_A^k)\mathbf{A}_j \wedge \mathbf{A}_k \equiv l^p L_p^A, \quad (37)$$

which tells us that the *line coefficients* $\{l^j\}$ are

$$\begin{aligned} l^1 &= (\mathbf{X}_1 \cdot \phi_A^2)(\mathbf{X}_2 \cdot \phi_A^3) - (\mathbf{X}_1 \cdot \phi_A^3)(\mathbf{X}_2 \cdot \phi_A^2) \\ l^2 &= (\mathbf{X}_1 \cdot \phi_A^3)(\mathbf{X}_2 \cdot \phi_A^1) - (\mathbf{X}_1 \cdot \phi_A^1)(\mathbf{X}_2 \cdot \phi_A^3) \\ l^3 &= (\mathbf{X}_1 \cdot \phi_A^1)(\mathbf{X}_2 \cdot \phi_A^2) - (\mathbf{X}_1 \cdot \phi_A^2)(\mathbf{X}_2 \cdot \phi_A^1). \end{aligned} \quad (38)$$

Utilizing the fact that the join of the duals is the dual of the meet, we are then able to deduce identities of the following form for each $l^j$

$$l^1 = (\mathbf{X}_1 \wedge \mathbf{X}_2) \cdot (\phi_A^2 \wedge \phi_A^3) = (\mathbf{X}_1 \wedge \mathbf{X}_2) \cdot (\phi_2^A \cap \phi_3^A)^* = L \cdot (L_1^A)^*.$$

We therefore obtain the general result,

$$l^j = L \cdot (L_j^A)^* \equiv L \cdot L_A^j, \quad (39)$$

where we have defined $L_A^j$ to be the dual of $L_j^A$. Thus, we have once again expressed the projection of a line $L$ onto the image plane by contracting $L$ with the set of lines dual to those formed by intersecting the optical planes.

If we express the world and image lines as bivectors, $L = \alpha_j e_j + \tilde{\alpha}_j I e_j$ and $L_A^p = \beta_j e_j + \tilde{\beta}_j I e_j$, we can write the previous equations as a matrix equation:

$$\boldsymbol{l} = \begin{bmatrix} l^1 \\ l^2 \\ l^3 \end{bmatrix} \equiv P_L \bar{\boldsymbol{l}}$$

$$= \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} & u_{16} \\ u_{21} & u_{22} & u_{23} & u_{24} & u_{25} & u_{26} \\ u_{31} & u_{32} & u_{33} & u_{34} & u_{35} & u_{36} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \tilde{\alpha}_1 \\ \tilde{\alpha}_2 \\ \tilde{\alpha}_3 \end{bmatrix}, \quad (40)$$

where $\bar{l}$ is the vector of *Plücker coordinates* $[\alpha_1, \alpha_2, \alpha_3, \tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_3]$ and the matrix $P_L$ contains the $\beta$ and $\tilde{beta}$'s, that is, information about the camera configuration. Bayro ([41], Chap. 12) presents the n-view geometry of computer vision, where the equations of the essential and fundamental matrices, the trifocal, and quadrifocal tensors are given using the geometric algebras $G_{3,1}$ and $G_3$ for the projective space and projective plane respectively.

### F. CONFORMAL GEOMETRIC ALGEBRA

In Conformal Geometric Algebra (CGA), the Euclidean vector space $\mathbb{R}^n$ is represented in $\mathbb{R}^{n+1,1}$. This space has an orthonormal vector basis given by $\{e_1, \ldots, e_n, e_{n+1}, e_{n+2}\}$ with the properties $e_i^2 = 1$, $i = 1, \ldots, n$, $e_{n+1}^2 = 1$, $e_{n+2}^2 = -1$, $e_i \cdot e_{n+1} = e_i \cdot e_{n+2} = e_{n+1} \cdot e_{n+2} = 0$, $i = 1, \ldots, n$. A more detailed description of conformal geometric algebra can be found in [160] and [39], [41].

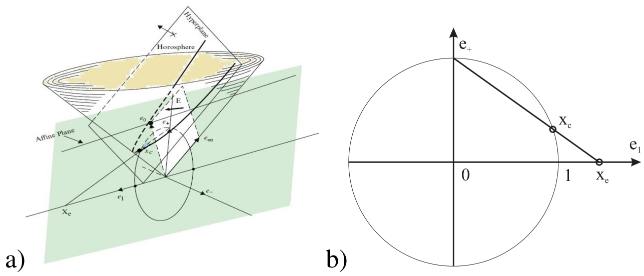



**FIGURE 1.** a) Null cone, hyperplane, horosphere and affine plane for the 1D case. b) Stereographic projection for the 1D case.

**TABLE 1.** Representation of entities in conformal geometric algebra.

| IPNS | OPNS |
|---|---|
| $x_c = x + \frac{x^2}{2} e_\infty + e_0$ | $x^* = S_1 \wedge S_2 \wedge S_3 \wedge S_4$ |
| $s = c_e + \frac{c_e^2 - \rho^2}{2} e_\infty + e_0$ | $s^* = X_1 \wedge X_2 \wedge X_3 \wedge X_4$ |
| $\pi = n - d e_\infty$<br>$n = (x_1 - x_2) \wedge (x_1 - x_3)$<br>$d = (x_1 \wedge x_2 \wedge x_3)\, I_e$ | $\pi^* = e_\infty \wedge X_1 \wedge X_2 \wedge X_3$ |
| $L = \pi_1 \wedge \pi_2$<br>$L = n I_e - e_\infty m I_e$<br>$n = x_1 - x_2$<br>$m = x_1 \wedge x_2$ | $l^* = e_\infty \wedge X_1 \wedge X_2$ |
| $Z = S_1 \wedge S_2$<br>$Z = \pi \wedge S$ | $z^* = X_1 \wedge X_2 \wedge X_3$ |

A *null basis* $\{e_0, e_\infty\}$ (origin and point at infinity) is computed as follows:

$$e_0 = \frac{(e_{n+1} - e_{n+2})}{2}, \quad e_\infty = e_{n+1} + e_{n+2}, \quad (41)$$

with the properties $e_0^2 = e_\infty^2 = 0$ and $e_\infty \cdot e_0 = -1$. $\boldsymbol{E} = e_\infty \wedge e_0$ is the Minkowski plane. The unit Euclidean pseudo-scalar is $I_e := e_1 \wedge e_2 \wedge e_3$, and the conformal pseudoscalar $I_c = I_e \boldsymbol{E}$ is used for computing the inverse and duals of multivectors. For a detailed treatment of conformal geometric algebra for robot vision see [42].

Conformal geometry is equivalent to stereographic projection in Euclidean space.

#### 1) POINTS, LINES, PLANES AND SPHERES

The representation of a 3D point in the geometric algebra $G_{4,1}$ is given by

$$x_c = x_e + \frac{1}{2} x_e^2 e_\infty + e_0. \quad (42)$$

The line can be obtained in its Inner Product Null Space (IPNS) form as

$$\boldsymbol{L} = nI_E - e_\infty \boldsymbol{m} = (\boldsymbol{n} + e_\infty \boldsymbol{m}), \tag{43}$$

where the bivector $\boldsymbol{n}$ is the orientation and the vector $\boldsymbol{m}$ the moment of the line. The IPNS equation of the plane is given by

$$\pi = \boldsymbol{n}I_E - e_\infty d = n - e_\infty d; \tag{44}$$

where the vector $n$ stands for the orientation of the plane and $d$ the Hesse distance. The IPNS equation for the sphere is given by

$$s = p_c - \frac{1}{2}\rho^2 e_\infty = p_e + (\frac{p_e^2 - \rho^2}{2})e_\infty + e_0. \tag{45}$$

Using the equation dual for the sphere, we write the constraint for a point lying on a sphere:

$$x_c \wedge s^* = x_c \wedge (s \cdot I_c) = 0. \tag{46}$$

The advantage of the dual form is that the Outer Product Null Space (OPNS) equations of the circle or of the sphere can be directly computed from three or four points respectively as follows:

$$\begin{aligned} z^* &= x_{c_1} \wedge x_{c_2} \wedge x_{c_3} \\ s^* &= x_{c_1} \wedge x_{c_2} \wedge x_{c_3} \wedge x_{c_4}. \end{aligned} \tag{47}$$

If we replace in equations (47,47) one of these points for the point at infinity, we get the OPNS equation of the line and plane respectively

$$\begin{aligned} l^* &= x_{c_1} \wedge x_{c_2} e_\infty, \\ \pi^* &= x_{c_3} \wedge x_{c_1} \wedge x_{c_2} \wedge e_\infty \\ &\quad + x_{e_3} \wedge x_{e_1} \wedge x_{e_2} \wedge e_\infty + ((x_{e_3} - x_{e_1}) \wedge (x_{e_2} - x_{e_1}))\boldsymbol{E}. \end{aligned}$$

Subsection II-E1 presents the algebra of incidence in the 4D geometric algebra $G_{3,1}$ using points, lines, and planes. As an extension, the incidence algebra in the conformal geometric algebra $G_{4,1}$ uses points, lines, planes, circles, and spheres of the OPNS, for more details see [41].

### 2) RIGID TRANSFORMATIONS

The translation of a geometric entity $Q \in G_{4,1}$ can be carried out by two reflections with parallel planes $\pi_1$ and $\pi_2$ as follows

$$Q' = \underbrace{(\pi_2 \pi_1)}_{T_a} Q \underbrace{\left(\pi_1^{-1} \pi_2^{-1}\right)}_{\tilde{T}_a}, \; T_a = 1 + \tfrac{1}{2} a e_\infty = e^{-\frac{a}{2} e_\infty},$$

with $a = 2dn$, $d$ the distance of translation, and $n$ the direction of translation.

A rotation is the product of two reflections between non-parallel planes $\pi_1$ and $\pi_2$ that cross the origin. The rotation is then defined by

$$Q' = \underbrace{(\pi_2 \pi_1)}_{R_\theta} Q \underbrace{\left(\pi_1^{-1} \pi_2^{-1}\right)}_{\tilde{R}_\theta}. \tag{48}$$

Computing the geometric product of the normal of the planes $n_1$ and $n_2$ yields

$$R_\theta = n_2 n_1 = \cos\left(\theta/2\right) - \sin\left(\theta/2\right)\boldsymbol{n} = e^{-\theta \boldsymbol{n}/2}, \tag{49}$$

with the unit bivector $\boldsymbol{n} = n_1 \wedge n_2$, and $\theta$ twice the angle between $\pi_1$ and $\pi_2$.

The screw motion called a motor is a composition of a translation and a rotation, both related to an arbitrary axis $L$. The motor is defined as

$$\begin{aligned} \boldsymbol{M} &= \cos\left(\frac{\theta + e_\infty d}{2}\right) + \boldsymbol{L} \sin\left(\frac{\theta + e_\infty d}{2}\right) \\ &= \boldsymbol{T}_s \boldsymbol{R}_s = \boldsymbol{R}_s + e_\infty \boldsymbol{R}'_s = \exp^{-\theta \boldsymbol{L}}, \end{aligned} \tag{50}$$

where the screw line is $\boldsymbol{L} = \boldsymbol{n} + e_\infty \boldsymbol{m}$. Therefore, a motor transformation for an geometric entity $Q \in G_{4,1}$ is given by

$$Q' = \boldsymbol{T}_s \boldsymbol{R}_s Q \tilde{\boldsymbol{R}}_s \tilde{\boldsymbol{T}}_s = \boldsymbol{M} Q \tilde{\boldsymbol{M}}. \tag{51}$$

A more detailed description of conformal geometric algebra can be found in [41]. Table 2 shows a resume of the relevant geometric algebras and their possible application domains.

**TABLE 2. Geometric algebras and possible applications domains.**

| Geometric Algebra | Applications |
|---|---|
| $G_n$, $G_{p,q,r}$ | GAs for euclidean or pseudo-euclidean metric |
| $G_3$ | 3D Euclidean GA for 3D motion projective plane |
| $G_3^+$ | Rotors for SO(3) computer vision, signal processing, model, control robotics, neural networks |
| $\mathbb{H} \equiv G_3^+$ | Quaternions for signal processing, Quaternion Fourier Transforms, neural networks |
| $G_{3,0,1}^+$ | Motors for SE(3) computer vision, screw theory, model control robotics, Newton-Euler dynamics, interpolation, Extended Kalman Filter, Neural networks, graphics |
| $G_{3,1}$, $G_{4,1}$ | CGAs for computer vision, screw theory l model control robotics, Euler-Lagrange dynamics, interpolation, medical robotics, neural networks, graphics, GIS |

The reader can resort to many books on geometric algebra and applications, see [246] to [256]. Appendix I presents a glossary of the most used variables in GA. Appendix II includes references to popular software packages to learn about GA and develop algorithms using the geometric algebra framework.

## III. APPLICATIONS IN ENGINEERING AND COMPUTER SCIENCE

In this section, we review several works from 1995 to 2020 on applications of GA in different areas of engineering and computer science. Based on these inquiries, we carried out a study of the research tendencies and a prognosis for new developments in near future. The results are presented in the final section. Section II outlines geometric algebra and explain the different geometric algebras $G_{p,q,r}$. In the next review of works, there are some which involve Clifford algebra. Geometric algebra is a geometric interpretation of Clifford algebras, thus it is not the same as Clifford algebra. The reader should resort to [53] for an introduction to Clifford algebras. The section on geometric algebras explains most of the concepts of Clifford algebra however in a geometric way for example to model the geometric entities and the Lie groups.

### A. ELECTROMAGNETISM: THE MAXWELL EQUATIONS

In the theory of electromagnetism, we find the Space-Time Algebra (STA) [118] in $G_{1,3}$, which provides a unifying framework for all four Maxwell equations. This STA method is presented by S. Gull, C. Doran, and A. Lasenby [Chaps. 8,11 [11]] to treat electromagnetic waves, including boundaries, propagation in layered media, and tunneling. On the other hand, in the alternative framework called Algebra of Physical Space (APS) in $Cl_{3,0}$ W. Baylis uses parabivectors $scalars + bivectors$ to treat polarized electromagnetic waves [Chaps. 17,18 [11]].

Chappel *et al.* [70] show how Heaviside reduced the Maxwell's ten field equations to four. However, using bivectors and the trivector in $G_3$ instead of the Heaviside-Gibbs vectors, they have a more appropriate representation for the essential definitions of electromagnetic theory; like the field, current, energy in fields, conservation of energy, force, potential function, Lorentz gauge, and potential equation.

### B. ELECTRICAL ENGINEERING

The development of the electrical power theory in the geometric algebra framework provides a new methodology to explain and to formulate the power flow in electrical systems of any type; this is due to the flexibility and generalization of the geometric language of GA for the multivector representation of the power flow in the cases of sinusoidal and non-sinusoidal systems. The works by Menti [176], Castro-Nuñez [62]–[64], [66], [177], Montoya [177], [178], Castilla and Bravo [59]–[61], Lev-Ari [181] and Petroianu [198] proved the power of the geometric language of GA for the analysis of power systems, namely a better understanding of the power balances.

As explained in previous studies [62], [63], [177], GA applied to sinusoidal, non-sinusoidal, linear, and nonlinear circuits is a promising technique to describe the power flow in terms of the energy conservation principle. The decomposition of the current into in-phase and quadrature components $\boldsymbol{I} = \boldsymbol{I}_c + \boldsymbol{a}$ has contributed to the development of methods for quadrature RMS current compensation [65]. However, Montoya *et al.* [178] found inconsistencies in the power formulation and reformulated the power theory based on GA (GAPoT) to correct such inconsistencies and shortcomings and define the total current through a decomposition considering the active current suggested by Fryze.

### C. SIGNAL PROCESSING

GA applications in signal processing include terahertz spectroscopy which allows efficient processing of time-domain signals [238]. Furthermore, GA applications in anisotropic materials and metamaterials which provide a more general description for metamaterials that can encourage new innovative procedures [170] as in the case of electromagnetic conductors [188].

Arsenovic [9] presented an ingenious method based on conformal geometric algebra to modernize the transmission line theory. The author formulates explicitly the relationships between the Smith Chart and Riemann Sphere. Circuit operations, for example, the addition of impedance and admittance, the impedances of the lines can be changed using rotations in CGA. In general, the majority of relationships in transmission line formulation can be linearized. The author solved some classical impedance matching problems.

### D. IMAGE PROCESSING

#### 1) FILTERING ALGORITHMS

There are many recent works on adaptive filtering applied to signal processing. A serious problem is that the performance of the filtering algorithm decreases due to the non-Gaussian noise. Using higher-order statistics, Constantinides [223] proposed an adaptive filtering algorithm called the least mean kurtosis (LMK). Chen *et al.* [73] introduced the QLMK for 3D and 4D signal processing, where its cost function is formulated in terms of the negative kurtosis of the error signal for adapting smoothly to non-Gaussian data. In addition, the QLMK algorithm estimates recursively multivectors representing rotations in 3D and 4D dimensions. Took *et al.* [226] developed the quaternion least mean square (QLMS) algorithm and the augment QLMS algorithm (AQLMS) which works adaptive filters for 3D and 4D signal processing. Mandic *et al.* [171] proposed the LMS algorithm in terms of the HR calculus, which computes the derivatives of the cost function just in the quaternion domain. Wang *et al.* [234] suggested a new Least-Mean Kurtosis adaptive filtering algorithm developed in geometric algebra. It is called GA-LMK and it represents the multidimensional signal as a multivector. The GA-LMK algorithm minimizes the cost function using the negated kurtosis of the error signal. B. Lopez and G. Lopes [167] formulated a novel adaptive filters called the Geometric Algebra Adaptive Filters (GAAFs), the approach is a minimization problem of a deterministic cost function in the GA framework. B. Lopez *et al.* [166] used the GA framework to design a novel adaptive filtering method to tackle the rotation estimation problem. Al-Nuaimi *et al.* [4] used GA-LMS to recover the 6 DOF alignment of two-point clouds in correspondence.

#### 2) FEATURE EXTRACTION ALGORITHMS

Feature extraction in image analysis is a fundamental procedure for low-level image processing like image segmentation, image registration, image mosaicking, motion recognition, pattern recognition, classification, and neurocomputing. Usually, in multi-channel image processing, the authors converted the images into grayscale images and then process them separately. This approach does not consider the natural correlation of each image channel, thus lowering the quality of the feature extraction method.

To tackle this drawback, Li *et al.* [182] suggested the GA-SIFT method which extends in the GA framework the SIFT procedure for multi-spectral images. GA-SIFT uses spectral and spatial information and it detects the feature points both in the spatial space and in the spectral space improving greatly the processing. On the other hand, Wang *et al.* [232] developed the GA-SURF by extending the traditional SURF using the geometric algebra framework. The GA-SURF utilizes a box filter to approximate the high complexity of the second Laplace derivative improving the calculation efficiency. For real-time multi-spectral image processing, Wang *et al.* [235] suggested a fast and rotated algorithm denoted as GA-ORB, which extracts near in real-time multi-spectral image interest points. The authors show that GA-ORB for extracting and matching interest points outperforms many current algorithms w.r.t. distinctiveness and robustness and it is even much faster.

Spatio-Temporal Interest Points (STIPs) were developed for extracting image interest points. The method was extended to extract points in the 3D space as well. STIPs are seen as a key invariant feature in video processing to detect object motion or action recognition directly without the need for foreground segmentation or background modeling. Wang *et al.* [231] proposed the GA-STIP utilizing the traditional 3D Harris algorithm and the Gaussian multi-resolution pyramid in the geometric algebra framework for scale-space of multi-channel video processing. GA-STIP is robust for feature extraction from a multi-channel video. It shows a good performance to recognize human activities.

It is known that lines and edges are key features in biological visual systems. Moya-Sánchez and Bayro-Corrochano [179] utilized the atomic function-based Riez transform in a multi-scale approach to extract characteristics of object symmetric shapes from images and to build feature-signature vectors for object classification. Bernal-Marin and Bayro-Corrochano [48] used the 2D and 3D Hough transform in conformal geometric algebra to construct 3D geometric maps utilizing lines and planes. This kind of representation permits to search for geometric constraints useful in robotics in the 3D visual space for re-localization, exploration, decision, navigation, and obstacle avoidance.

### E. COMPUTER VISION

For computer vision algorithms in the geometric algebra framework, authors use the 3D and 4D geometric algebras $G_3, G_{3,1}$ for the image plane $\mathbb{P}^2$ and the projective space $\mathbb{P}^3$ to formulate the fundamental concepts of projective geometry, projective split and theory of invariants relevant for computer vision. Authors show the incidence-algebra operations meet (intersection), join a duality principle to compute intersections or joins of geometric entities as points, lines, and planes. In $G_3$ and $G_{3,1}$ they present the analysis of monocular, binocular, and trinocular geometries equivalent to the theory of n-view geometry. The extension of the geometric algebras $G_3$ and $G_{3,1}$ to the conformal geometric algebras $G_{3,1}$ and $G_{4,1}$ augments the D.O.F. of the mathematical system so that one can represent also circles, cones, and spheres and extend the Lie groups to the conformal group which includes the rigid, affine and projective transformations. Even though that the CGA uses a quadratic model of the horosphere, one can get the Euclidean metric via the inner product of null vectors. Having more geometric entities and more bivectors to formulate the conformal transformation, authors could propose new models and find new geometric constraints to tackle complex problems in computer vision and robot vision.

Next, we review interesting contributions using $G_3$ $G_{3,1}$ and $G_{3,1}$ and $G_{4,1}$ to model the image plane and the 3D visual space. A seminal paper on projective geometry using Clifford algebra was written by Hestenes and Ziegler [125]. It stimulated further research in the nineties in the areas of projective geometry, projective invariant theory, and computer vision. Lasenby *et al.* [152] introduced the application of geometric algebra for the estimation of structure and motion. Later on, Bayro-Corrochano and Lasenby [153] utilized the geometric algebra framework for expressing projective invariants using n cameras. Bayro-Corrochano and Banarer [29] formulated projective invariants formed from points, lines, and planes using two and three cameras and in the geometric algebras $G_{3,1}$ and $G_3$. The authors presented interesting applications of projective invariants for tasks like visual guided grasping, camera self-localization, and reconstruction of shape and motion.

Lasenby et al [154] showed that using the mathematical framework of conformal geometric algebra a 5-dimensional representation of 3-dimensional space – they can provide an elegant covariant approach to geometry. In this language, objects such as spheres, circles, lines, and planes are simply elements of the algebra and can be transformed and intersected with ease. In addition, rotations, translation, dilations, and inversions all become rotations in our 5-dimensional space;

Bayro-Corrochano, Reyes-Lozano and Zamora-Esquivel [30] proposed geometric methods for visually guided robotics. The authors CGA present interesting methods useful for kinematics, dynamics, and projective geometry problems.

Ortegon-Aguilar and Bayro-Corrochano [186] presented the estimation of 2D and 3D transformations in terms of the Lie group $GL(n)$ or one of its subgroups useful for monocular estimation of object 3D rigid motion and for the estimation of the involved affine or projective transformations during monocular region tracking.

Debaecker *et al.* [75] proposed a new cone-based camera model using the called ''coxels'' a contraction of pixels of fields of view as volumetric cones. The model uses conformal geometric algebra that allows representing cones efficiently using twists. As to applications the authors considered the use of this model to derive robust motions estimation by redefining the notion of pixel's intersections. A calibration method is also presented which provides all cameras with parameter linking pixels to metric cones aperture.

Altamirano-Gómez and Bayro-Corrochano [5] developed an approach for a local voting process using global perceptual considerations at a low level and a global voting process based on clusters of salient geometric entities. The method represents objects in noisy images in terms of circles and lines even though the shapes are distorted by noise, or the contours are incomplete, or they are illusory or non-linear.

Ghina-El Mir, Saint-Jean and Bertier [86] formulated a model for image representation based on CGA to encode perspective distortions using the Minkowski metric in space $\mathbb{R}^{1,1}$.

Hrdina and Návrat [138] solved a generalized binocular vision problem using CGA. The position of two cameras is determined by arbitrary Euclidean transformations depending on certain degrees of freedom. These transformations are represented by a motor and projection unknown parameters to derive general equations in CGA for the 3D object reconstruction involving two camera projection image planes. The procedure was needed to solve a system of non-linear equations.

Ron Goldman *et al.* [104] showed how to compute perspective projection in 3D using rotations and spherical inversions in the homogeneous and conformal models in CGA. Their method started with a view direction and the distance to the projection plane and constructed the eye point from the information. The proposed perspective projection allowed to place the eyepoint arbitrarily in space, view direction, and the projection plane.

Leo Dorst [83] explicitly showed how various known projective transformations (translations, rotations, scalings, perspectivities, Lorentz transformations) are represented in geometric meaningful parameterizations of the rotors in terms of bivectors.

López-González *et al.* [164] used the randomized Hough transform for detecting in terms of k-blades lines and circles in images and planes, circles and spheres applying conformal geometric algebra framework $G_{4,1}$.

Gunn [110] introduced the dual projective Clifford algebra $\mathbb{P}(\mathbb{R}^{n,0,1})$ (PGA) as the most promising homogeneous (1-up) candidate for euclidean geometry. He compared PGA and the popular 2-up model CGA, restricting attention to flat geometric primitives, and showed that in this domain they exhibit the same formal feature set. The author thereby established that PGA is the smallest structure-preserving euclidean GA. He compared the two algebras in more detail, concerning several practical criteria, including implementation of kinematics and rigid body mechanics. Then, they extended the comparison to include euclidean sphere primitives.

In another article, Gunn [111] presented an introduction to projective geometric algebra (PGA), a modern, coordinate-free framework for doing euclidean geometry. PGA features: uniform representation of points, lines, and planes; robust, parallel-safe join and meet operations; compact, polymorphic syntax for euclidean formulas and constructions; a single intuitive sandwich form for isometries; native support for automatic differentiation; and tight integration of kinematics and rigid body mechanics.

### F. INTEGRAL TRANSFORMS

An integral transform $T$ maps a function $f$ from its original function space $V$ into another function space via integration, where some of the properties of the original function are more easily characterized and manipulated than in the original function space. Using the inverse transform $T^{-1}$, the function $G = Tf$ can be mapped back to the original function space $V$.

#### 1) CLIFFORD-QUATERNION FOURIER AND WAVELET TRANSFORMS

Since 1990 authors published different works using quaternion algebra to formulate integral transforms in the frequency domain. In a pioneering article, Chernov [71] used the quaternion algebra framework to speed up the computing of the 2D discrete, complex-valued, Fourier transform.

T. Ell [85] applied the QFT for the analysis of 2D linear, time-invariant, partial-differential systems. In this approach, the $\psi$ phase component of the polar representation of the QFT was utilized to assess the stability of the system. T. Ell in [112] also published a current overview of quaternion Fourier transform (QFT). The author presented definitions, their relations, inversion, linearity, convolution, correlation, and modulation.

Later on, Bülow [55] introduced the quaternionic phase concept and contributed to the clarification of the theory and practice of the QFT.

Sangwine [205], [206] showed that QFTs found rich applications in color image processing; he related the colors R, G, B with the quaternion coefficients of i, j, k. The color images are then transformed in a holistic manner using the QFT instead of separately transforming each color channel by the 2D Fourier transform.

Inspired on the concept of the Mellin transform, M. Bahri, E. Hitzer, and S. Adji in [38], pp. 93–106 proposed in $Cl_{2,0}$ the 2D windowed Clifford Fourier transforms for local multivector-image analysis. Girard *et al.* [101] formulated the processing of analytic video (2D+t) signals using the $Cl_{0,3}$ Clifford algebra framework. The authors show that $Cl_{0,3}$ can be represented in terms of double quaternions $q_1 + q_2$, where with$^2 = 1$ (double number) corresponds to the pseudoscalar of $Cl_{0,3}$, and $i, j, k$ are the bivectors of Cl(0, 3).

Felsberg and Sommer [97] proposed a 2D generalization of the analytic signal. This new method is formulated in terms of the Riesz transform used instead of the Hilbert transform. An essential contribution is the geometric interpretation of the phase concept which establishes a relation between the 1-D analytic signal and the 2-D monogenic signal which is surprisingly formulated by the Radon (1986) transform.

Bernstein [112] proposed the monogenic signal theory to express the monogenic curvature in scale space and proposes the diffusive wavelets for the demodulation of 2D AM-FM signals.

Pollock and Mann [199] formulated a method for detecting vortices in sampled vector fields using the geometric algebra framework. Their vortex-detection algorithm recursively searches for vector samples and by applying the 2D version of the Gaussian Theorem the algorithm extracts vortex cores.

Mann and Rockwood [172] computed singularities of 3D vector fields and an octree-based solution for finding critical points and their indexes in the 3D vector field.

W. Reich and G. Scheuermann [87], [88] proposed the Clifford convolution extending the classical convolution on scalar fields which were used for pattern matching on vector fields. The Clifford Fourier transform [54] is utilized for the analysis of discretization, sampling, measurement, and interpolation errors. This method is utilized to analyze the behavior of smoothing and differential operators, which play an important role in feature detection on flow fields.

E. Bayro [31] extended the real and complex wavelet transforms to the quaternion wavelet transform and formulates a quaternionic wavelet pyramid for multi-resolution analysis utilizing the quaternionic phase concept. The author showed an application of the discrete QWT for optical flow estimation. The estimation of motion is carried out through different resolution levels. The method uses a similarity distance which is evaluated using a confidence mask and the quaternionic phase concept.

E. Hitzer [132], [133] proposed the generalization of the orthogonal 2D plane split (OPS) of quaternions. The OPS split uses the choice of one or two general pure unit quaternions $\boldsymbol{p}^2 = \boldsymbol{q}^2 = -1$, which correspond to two observers planes in the $\mathbb{R}^4$ space of quaternions. In this approach, the QFTs can be split into pairs of complex FFTs, QFTs which can be seen geometrically as plane observers. This novel QFT is formulated considering desired geometric and phase characteristics, for special applications, and allows their generalizations to higher dimensional Clifford algebras as shown in [134] a space-time FT in $Cl_{3,1}$. This method resembles the space-time multivector wave packet analysis.

T. Batard and M. Berthier in article No. 167 of [112] suggested a spinor representation of arbitrary surfaces for images. Spinor algebras are sub-algebras of Clifford algebras as complex numbers, quaternions, rotor, and motor algebras. The authors showed image processing applications such as segmentation, diffusion and a Fourier Transform for multi-channel images to *Spin*(3).

E. Moya-Sanchez and E. Bayro-Corrochano in [36] and ctrb. No. 187 of [112] proposed the quaternion atomic function kernels for application in image processing. The approach also allows the use of analytic functions for the generalization of monogenic functions and the development of a steerable quaternionic multiresolution wavelet scheme for image structure and contour detection. In addition, the authors generalized the Radon transform which permits the detection of color image shape contours.

In article No. 172 of [112], S. Ebert, S. Bernstein, and F. Sommen proposed a harmonic analysis for the Clifford valued functions on the spin group as the rotor or motor groups. The authors analyze the Clifford valued diffusive wavelets on the sphere.

### 2) CLIFFORD AND QUATERNION FOURIER TRANSFORM FOR COLOR IMAGE PROCESSING

In the past, researchers in color image processing separated the processing of red, green, and blue color channels. Conceptually the color space should be analyzed in a holistic 3D manner. New works suggest holistic multidimensional processing using quaternion algebra and geometric algebra. These approaches overcome the limitation of channel-wise processing and guarantee a sound color space image processing either using Euclidean metric of the $\mathbb{R}^3$ or space-time (Minkowski) metric in $G_{2,1}$ [43].

For 2D color image processing, T. Batard et al [12] developed a special color formulation of the Clifford Fourier transform (CFT) [54] in $Cl_{4,0}$.

Menneson *et al.* [174] proposed techniques for color object recognition using the Clifford Fourier transform introduced in [13], however, this color CFT does not have a general formulation for multivector signals and an inverse. The CFT color embeds a 3D color vector signal in $\mathbb{R}^4$ using a bivector $\boldsymbol{B} \in G_4$ and its dual $I_4$ $B$. This CFT is different from the standard CFT proposed by Ebling and Scheuermann [88] which can be utilized to general multivector-valued signals and it has a general inverse CFT.

R. Soulard and P. Carre [112] proposed a color extension of the monogenic signal using wavelets by embedding the color vector signals in the geometric algebra $G_5$. This approach seems to be better than the old-fashioned channel-wise color image processing. The authors formulated a novel multiresolution geometric color analysis utilizing non-separable wavelets, which guarantee a good orientation analysis well separated from the color information. The method includes a statistical coefficient modeling for thresholding and denoising.

Pedone *et al.* [195] proposed a global method for the registration of color images w.r.t. translation. Their method represents translations as convolutions with unknown shifted delta functions and performs the Wiener deconvolution for recovering the shift between two color images. Interestingly enough, they derived a quaternionic version of the Wiener deconvolution filter for the registration. The Wiener filter explicitly takes into account the variations of noise.

S. Lazendinć *et al.* [158] proposed a generalization of the quaternion dictionary learning method using the octonion algebra framework. The classical dictionary learning methods concatenate spectral bands into a large monochrome image; in contrast, the authors processed all the spectral bands simultaneously. Their method preserves better the color fidelity of the reconstructed multispectral image. Image reconstruction and denoising were carried out for color images and Landsat 7 images.

#### 3) EDGE DETECTION AND CLUSTERING OF COLOR IMAGES AND MULTISPECTRAL IMAGES

Traditional approaches of color edge image detection relied on processing the separated red, green, and blue color channels. Koschan and Abidi [148] provided an overview of edge detection techniques for color images. E. Bayro-Corrochano and S. Flores in Chap. 29 in [84] proposed a color edge detector using rotors in $G_3$ and a Multi-Layer Perceptron. Schlemmer *et al.* [208] proposed a novel approach based on Sobel operator and used vector value filter masks to detect color edges.

Franchini *et al.* [98] proposed a FPGA implementation to speed up their edge detection method which uses the geometric product of k-vectors, to extend the convolution operator and to apply the Fourier transform to vector fields. Their approach is applied successfully for edge detection of multispectral magnetic-resonant images.

Edge detection in multispectral images has been a challenging task over the past few years. Multispectral images as in the cases of medical images or remote sensing images, usually involve hundreds of spectral channels of the same scene; therefore it is required of computational efficient edge detectors.

Cao *et al.* [139], [241] developed a new vector-based approach for edge detection in multichannel remotely sensed images. The authors detected the discontinuity between homogeneous image regions using the image density value estimated at the mean vector of the sliding window.

Wang R., Shen M. and Cao W. [233] proposed a new multivector sparse representation model for multispectral images using the geometric algebra framework. Their model represents a multispectral image as a GA multivector considering fully the spatial and spectral information. Moreover, a GA dictionary learning algorithm is formulated using the K-GA-singular value decomposition (GASVD)- As a result, due to the consideration of the relationship between spectral channels in multispectral images, the method successfully manages to reject the artifacts and blurring effects.

### G. COMPUTER GRAPHICS AND OTHERS

Gómez *et al.* [105] explored the use of Clifford algebra to represent n-dimensional lattices. This approach allows to describe the geometrical crystallography in a sound language valid in any dimension. This work shows applications to the problems in quasi-crystals when handling the faceting and phason degrees of freedom.

Wareham et al [236] introduced conformal geometric algebra (CGA) as a language for computer graphics and computer vision. They compute position interpolation based on CGA. Their method can be extended to higher-dimension spaces and all conformal transforms (including dilations). The authors discussed a method of dealing with conics in CGA and the intersection and reflections of rays with such conic surfaces.

Hitzer and Perwass [131] developed an intuitive software tool that visualizes 3D space group symmetries. The interactive package computes various aspects of crystals using the Clifford (geometric) algebra framework. Their package corresponds to an algebraic implementation of groups generated by reflections. A combination of reflections relates the geometric product of vectors which in turn describe the orientation of the reflection planes. The interactive graphics package coupled with CLUCalc [196] can be used to understand how reflections are combined to generate all the 230 3-D space groups.

Hildenbrand *et al.* [126] explained that at the beginning in computer graphics projective geometry was adequate to represent points and formulate their mappings using a linear transformation. In contrast, geometric algebra offers a new computational framework for handling transformations in terms of bivectors such as rigid motion, similitude, affine, projective, and conformal transformations. The authors clarified that it is also possible to handle the transformations of not only points but of lines, planes, circles, spheres, and hyperplanes. They showed that in geometric algebra, one can formulate complex numbers, quaternions, dual-quaternions, Grassmann algebra, and Grassmann-Cayley algebra. The authors wrote a tutorial emphasizing that geometric algebra is a unified language involving many mathematical systems which can be used advantageously in Computer Graphics.

Papagiannakis [189] showed that computer implementations of geometric algebra (GA) can perform at a faster level compared to standard (dual) quaternion geometry implementations for real-time character animation blending. By this, we mean that if some piece of geometry (e.g. Quaternions) is implemented through geometric algebra, the result is as efficient in terms of visual quality and even faster (in terms of computation time and memory usage) as the traditional quaternion and dual quaternion algebra implementation. His work describes two implementation approaches for quaternion interpolation using Euclidean GA rotors for skinned character animation blending. It also lays the foundation so that GA can be employed for further calculations (skinning, rendering) under a unified geometry computation framework.

Papaefthymiou et al [190] presented an all-inclusive algorithm for real-time animation interpolation and GPU-based geometric skinning of animated, deformable virtual characters using the Conformal model of Geometric Algebra (CGA). They compared their method with standard quaternions, linear algebra matrices, and dual-quaternions blending and skinning algorithms and illustrated how their CGA-GPU inclusive skinning algorithm can provide a smooth and more efficient results as state-of-the-art previous methods.

Papaefthymiou et al [191] proposed an efficient method for robust authoring (rotation) of Augmented reality scenes using Euclidean geometric algebra (EGA) rotors and they present two fast animation blending methods using GA and CGA. The authors compared the efficiency of different GA code generators: (a) Gaigen library, (b) libvsr and (c) Gaalop using their animation blending methods and compare them with other alternative animation blending techniques: (a) quaternions and (b) dual-quaternions, so that a future user of GA libraries can choose the most appropriate one that will give the most optimal and faster results.

Hadfield et al [113] explained that modern Geometric Algebra software systems tend to fall into one of two categories, either fast, difficult to use, statically typed, and syntactically different from the mathematics or slow, easy to use, dynamically typed, and syntactically close to the mathematical conventions. Gajit is a system that aims to get the best of both worlds. It allows us to prototype and debug algorithms with the Python library Clifford which is designed to be easy to read and write and then to optimize our code both symbolically with GAALOP and via the LLVM pipeline with Numba [113] resulting in highly performant code for very little additional effort. |

Keninck [146] introduced a novel visualization method for elements of arbitrary Geometric Algebras. The algorithm removes the need for a parametric representation, requires no precomputation, and produces high-quality images in real-time. It visualizes the outer product null space (OPNS) of 2-dimensional manifolds directly and uses an isosurface approach to display 1- and 0-dimensional manifolds. A multi-platform browser-based implementation is publicly available.

Kamarianakis and [145] presented an integrated rigged character simulation framework in Conformal Geometric Algebra (CGA) that supports, for the first time, real-time cuts and tears, before and/or after the animation, while maintaining deformation topology. Previous implementations originally required weighted matrices to perform deformations, whereas, in the current state-of-the-art, dual-quaternions handle both rotations and translations, but cannot handle dilations. Their CGA algorithm also provides easy interpolation and application of all deformations in each intermediate step, all within the same geometric framework. These interactive, real-time cut and tear operations can enable a new suite of applications, especially for medical surgical simulation.

### H. GEOSPATIAL DATA, REMOTE SENSING, GIS, AGRICULTURE

Bhatti *et al.* [52] provided a detailed review of GA in different fields of AI and computer vision regarding its applications and the current developments in geospatial research. The authors showed applications of the Clifford–Fourier transform and quaternion algebra in remote sensing image processing. They focus on how GA helps AI and solves classification problems, as well as improving these methods using geometric algebra processing in terms of Clifford Support Vector Machines. The authors discuss the issues, challenges, and future perspectives of GA with regard to possible research directions.

Yu, Yuan, and Luo [244] presented a unified multi-dimensional GIS data model, constructed by linking data objects of different dimensions within the multivector structure of Clifford algebra. The authors proposed algorithms for geographical network analysis (such as shortest path, minimum, and maximum flow analysis), and a high-dimensional Voronoi diagram was constructed. The authors showed that traditional GIS analysis algorithms can be extended not only to accommodate various dimensions but also to become beneficial regarding performance.

Yuan et al [245] explained that traditional Euclidean geometry-based Geographical Information System (GIS) is a multidimensional unification with a weak ability to object expression and analysis of high dimensions. In this regard, geometric algebra can connect different geometric and algebra systems, and in addition, it provides a rigorous and elegant foundation for expression, modeling, and analysis in GIS. The authors showed that the use of a multi-dimension-unified and coordinate-free GA framework allows us to handle data models, data indexes, and data analysis algorithms for multi-dimensional vector data, raster, and vector field data. Moreover, GA allows undoubtedly the convenient representation of multidimensional spatio-temporal changes.

Prince *et al.* [144] used Clifford geometric algebra to enhance the segmented images acquired from the UAVs of different agricultural fields. According to the authors, previous image segmentation approaches depend upon the intensity of red, green, and blue colors; but the complete perspective could not be obtained from these approaches. Geometric algebra overcomes this limitation and leads to genuine color space image processing. The image segmentation of foreground and background is enhanced using Clifford geometric algebra; hence, the results obtained are fine-tuned segmented images. The authors believe that their research would have a positive impact on the amelioration of the condition of the farmers and their livelihood.

### I. QUANTUM COMPUTING

Nowadays, quantum computers for consumers are not yet available, however, the use of neural networks to carry out quantum computing appears a promising approach to carry out quantum computing. This requires a flexible way to formulate quantum processors integrating and relating the principles of the fields of neurocomputing, quantum computing, and quantum mechanics. On the other hand, it is necessary to respect certain fundamentals principles of those fields. In this regard, we believe that neurocomputing helps to relate effectively those fields.

In quantum mechanics, the complex space notions and the imaginary unit $i_{\mathbb{C}}$ are essential.

Many works like Space Time Algebra (STA) [118], the 4D GA with Minkowski metric, $G_{3,1}$ [77], [214], [215] showed the role of $i_{\mathbb{C}}$ in the Dirac, Pauli, and Schrödinger's equations and its geometric interpretation in terms of rotations in real space time [119].

Other authors proposed the Multi Particle Space and Time Algebra (MSTA) [77], [79], [151], [215]. The MSTA formulates a time dimension and three spatial dimensions for each particle. The MSTA constructs a sound conceptual framework for a multi-time methodology to quantum theory.

In an enlightening article Cafaro and Mancini [56] presented an explicit GA characterization of 1 and 2-qubit quantum states and a GA characterization of a universal set of quantum gates for quantum computation. Moreover, the authors show the universality of quantum gates in terms of geometric algebra.

Conte [72] demonstrated quantum interference and indeterminism in Clifford algebra by using only logic. Quantum mechanics involves two basic foundations; indeterminism and quantum interference. The roots of indeterminism and of quantum interference are not necessarily in physics but it is clearly represented in a logic formulated in Clifford algebra. The author explains that cognition not only coexists with matter in quantum mechanics but it seems that the first supervises the latter.

Bayro [37] formulates quantum gates for the Lie Group SO(3) using neural networks in the sub-algebra $G_3^+$. They integrate the fields of neurocomputing, quantum computing, and quantum mechanics in a unifying mathematical framework. The authors claim that neural network models are used as the hardware for quantum processing. Along these lines of thought, the Autonomous Perceptron Neuron was generalized to the Quaternion Quantum Neural Network (QQNN) [45]. The article shows the quaternionic quantum neural networks for pattern recognition using surprisingly just one quaternion neuron. The experiments demonstrated the excellent performance of the QQNN.

### J. NEUROCOMPUTING

Initial attempts at applying geometric algebra to neural geometry have already been described in earlier papers [24], [121], [122]. Bayro [28] proved that standard feedforward networks are generalizable in the geometric algebra framework. The author formulated the Geometric Multi-Layer Perceptron and the Geometric Radial Basis Function. Furthermore, he used quaternion wavelets for the Quaternion Wavelet Function Network [20], [24] replacing the Radial Basis Functions in the hidden layer with quaternion wavelets.

Kusakabe *et al.* presented a study of neural networks using quaternion algebra [150]. The authors formulated the backpropagation training rule in terms of quaternions. Later on, Nobuyuki *et al.* [180] formulated similarly a quaternion neural network using a quaternion version of the back-propagation training rule. Their experiments present precise geometrical transformations in 3D space, as well as in color space image compression. The authors showed that the quaternion neural networks have a better performance than the real-valued neural network regarding convergence speed and the handling of the 3-bit parity check problem.

Kusamichi H, Isokawa T. and Matsui N. [149] presented a quaternion neural network for extracting color information from gloomy images. The network is trained by imposing a gloomy image as input and a good quality one as a target. A quaternion-based backpropagation training rule is adopted. The authors showed that their method is very useful for color night vision.

Bayro *et al.* [40] presented the design of Radial Basis Function geometric bioinspired networks and their applications. The question is how biological neural networks handle complex geometric representations involving Lie group operations like rotations. The authors used Atomic Functions and geometric neural networks to detect 2D geometric features of real images and motor algebra with RBF networks to act as an observer for detecting 3-D screw-line axis data.

Thiruvengadam *et al.* [224] developed a novel paradigm to design hyper-algebraic networks in the so-called hyper-field cognition framework expanding upon the mathematical foundations of neural networks in the 5D conformal geometric algebraic space.

The revolution of the Convolutional Neural Networks is basically due to the increase in the amount of hidden filter-convolutional layers, the inclusion of the pooling, and ReLU layers. However, much of the existing work has been focused on real-valued numbers. Convolutional neural networks (CNN) have recently achieved state-of-the-art results in various applications, see T. Parcollet, M. Morchid, and G. Linares [193]. Bayro [41] developed the Geometric Algebra CNN as a special case of the Quaternion CNN. The author represented the convolution and its products in the quaternion algebra framework. Gaudet and Maida [69] analyzed the benefits of generalizing the real-valued CNNs into the hypercomplex numbers, quaternions specifically, and provided the architecture components needed to build deep quaternion networks.

The brain can store information and adapt its neuro-connections for a variety of complex tasks necessary; this is known as synaptic plasticity [141], [143]. The third generation of neural networks, or Spiking Neural Networks (SNN) can mimic biological behavior. For the last ten years, we eye-witnessed impressive progress in the development of spiking neural networks with interesting applications [3], [102], [103], [137]. Bayro et al [45] proposed the Quaternion Spiking Neural Network (QSNN). This work shows that the quaternion spiking neural networks can adapt the robot online to reach the desired position.

#### 1) CLIFFORD SUPPORT VECTOR MACHINES

It is well known that the classical works on SVM algorithms are not able to carry out the classification of multiple classes which is solved by MIMO architectures. Bayro-Corrochano *et al.* [35], [38] generalized the real-valued SVM algorithms using geometric algebra. The authors proposed the CSVM which generalizes the real-valued SVM over the complex, quaternion, and hyper-complex values. The authors utilize kernels for nonlinear functions SVM which are formulated in Clifford algebra. The proposed kernels involving the Clifford or geometric product can be applied for nonlinear classification. Bayro-Corrochano and Arana-Daniel [35] presented the CSVM which can be used for the tasks of classification, regression, and recurrence. Similar to a Multiple-Input Multiple-Output (MIMO) architecture, the CSVM is fed with multiple multivector inputs and the outputs are multivectors; thus the CSVM can be utilized in multiple classes. Lopez et al [165] proposed a parallelization approach for the CSVM, based on two characteristics of the Gaussian Kernel. The authors showed that the pure real-valued result and its commutativity permit to separate the multivector data in its related subspaces. Since the subspaces are independent of each other, the classification problem can be straightforwardly solved using parallelism.

In recent work, Wang *et al.* [230] formulated the Clifford fuzzy SVM (CFSVM), which uses a fuzzy membership to each multiple input point for multiple classes; as a result, different input points have different contributions to the learning of decision surface. The CFSVM approach improves the CSVM in reducing the effect of outliers and noise in data points. CFSVM is suitable for applications where data points have unmodeled statistical characteristics.

### K. ROBOTICS: KINEMATICS, DYNAMICS, TRACKING, CONTROL

For kinematics and dynamics of robot mechanisms, many authors have used different mathematical systems such as vector calculus, quaternion algebra, or linear algebra which is most often utilized. Note that in these frameworks for handling the kinematics and dynamics only the geometric entities as points and lines are used, which induce complicated computations.

Ickes [140] presented in 1970 a method for performing digital control system attitude computations using quaternions. In 1970, Grubin [108] showed the derivation of the quaternion scheme via the Euler axis and angle. In the eighties, authors discussed quaternion algebra for orientation and for manipulation of finite rotations and animating quaternion curves [161], [211], [220]. In the nineties, authors showed the role of quaternions for rotation and attitude representations. [159], [212]. Dooley and McCarthy [76] showed that the motion of cooperating robot systems is formulated by relating the equations of motion for each arm and the workpiece utilizing the constraint equations of the closed chain. Dual quaternions seem to provide a sound algebraic representation for these constraints.

In geometric algebra, in 1995 begun the use of rotors and motors for kinematics [14]–[17], [19].

Berman, Liberman and Flash [50] used motor algebra, a 4D degenerate geometric algebra, which permits a rigorous yet simple formulation of the 3D rigid-body velocity. Using this approach, the authors analyzed the 3D extended arm pointing and reaching movements.

Seybold [210] defined and applied a nonlinear conjugate gradient method in a vector space. This approach solves multilinear functions formulated in the motor algebra, which address inverse kinematic problems. The approach is illustrated using a Stanford-type robot arm.

Thiruvengadam and Miller [225] represented a robotic system or manipulator in terms of a network whose motion-generating kinematic pairs are formulated by means of the network's inter-connected nodes. For this network theoretic framework, a formulation in Clifford Algebra was proposed utilizing higher dimensional multivectors that approximate the computational outcomes of complicated systems of equations. This approach is applied to solve the inverse and forward problems.

Bayro and Falcon [33] used conformal geometric algebra (CGA) for a variety of applications in computer vision and robotics. According to these authors CGA aids to increase our intuition and insight into the geometry in question. In addition, it helps us to diminish the computational burden of the task.

Hildenbrand, Pitt and Koch [128] developed the Geometric algebra algorithms optimizer (Gaalop). This is a package for high-performance computing using the conformal geometric algebra framework. Using Gaalop, Hildebrand et al [127] solved the inverse kinematics used in computer graphics and robotics.

Aristidu and Lasenby [7] presented a new forward and backward reaching inverse kinematics solver called FABRIK in CGA. The solver can find joint positions by locating points on lines. The approach can handle most joint types and support a variety of biomechanical constraints on chains with single and multiple end-effectors.

Yao *et al.* [243] presented a novel method based on GA for the singularity analysis of 3 DOF overconstrained planar parallel manipulators 3-RPR. GA offers a compact and geometrically intuitive formulation of the singularity polynomial of the parallel manipulators 3-RPR.

Lian [162] presented in conformal geometric algebra a method for geometric error modeling of Parallel Manipulators (PMs) in terms the visual representation and direct calculation in GA. Using linearization of the finite motion, he analyzes the error propagation of the open-loop chain. The method separates the error sources in terms of joint perturbations and geometric errors.

Bayro *et al.* [27] presented in the motor algebra framework the mathematical treatment of kinematics which becomes much easier using points, lines, and planes. John Selig [ [207], Chap. 9] analyzed the Newton mechanics and the Newton-Euler recursive algorithm in terms of screws as 6D vectors and the transformations using matrices. The author presents the use of additional geometric entities which further helps to reduce the representation and computational difficulties. As shown by Bayro [ [42], Chap. 14], the robot kinematics formulation can be handled advantageously using points, lines, planes, circles, and spheres in the CGA [127].

Bayro [39] introduced the Euler-Lagrange dynamics using CGA. Later on, he [ [42], Chap. 15] formulated the Newton-Euler recursive algorithm for dynamics using motor algebra. The Hamiltonian mechanics on phase space can be formulated in the GA as well [42]. Robot Newton-Euler Modeling and Control using Hamiltonians in GA were presented in [46]. In the modeling, control, and tracking for robot arms were formulated using a geometric algebra framework.

H. Hafield and J. Lasenby [115] tackled the problem of constrained rigid body dynamics in the Conformal and

Projective Geometric Algebras (CGA, PGA). First we construct a screw-theory based formulation of dynamics in CGA and note the equivalence between this and the PGA dynamics presented by Gunn in [110], [111]. After verifying the formulation via simulation, they add constraints applying the concept of virtual power.

Hafield and Lasenby [114] analyzed the forward and inverse kinematics of the Delta robot from a geometric perspective using Conformal Geometric Algebra. They calculated explicit formulae for all joints in both the forward and inverse kinematic problems as well as explicit forward and inverse Jacobians to allow for velocity and force control. They verify the kinematics in Python and simulate a physical model in the Unity3D game engine to act as a test-bed for future development of control algorithms.

### L. CONTROL ENGINEERING

In the last decades, researchers have mostly worked in robot modeling, kinematics and dynamics; fortunately, some researchers also worked to develop methods for control engineering using quaternions, dual quaternions, and geometric algebra. Thus, there is an urgent need to relate the geometric algebra framework with advanced techniques of non-linear control, envisage a new concept of geometric control using multivectors and Lyapunov functions of multivectors, derive cost functions for optimal control, and be able to prove the stability of the controllers in the k-vectors subspaces.

Xian *et al.* [237] studied the problem of task-space tracking control of redundant robot manipulators. Using a quaternion representation of the end-effector orientation, the authors design a class of task-space controllers that ensure asymptotic end-effector position and orientation tracking. Moreover, they proposed model-based and adaptive full-state feedback controllers which can eliminate the link velocity measurements via a model-based observer.

Price *et al.* [200] suggested a self-reconfigurable control for dual-quaternion systems with unknown control direction. The approach formulated the creation of multiple equilibrium surfaces for the system in the extended state space.

Maeda, Fujiwara, and Ito [169] formulated a position control scheme for an actual robot system using high dimensional neural networks. The authors proposed a complex-valued neural network and quaternion neural network which learn inverse kinematics of the robot systems.

Oviedo-Barriga *et al.* [183], [184] presented the controlling walking biped robots, which is a challenging problem due to their complex and uncertain dynamics. The authors proposed a sliding mode controller based on a dynamic model which was obtained using the CGA. They obtained the first and second derivatives of the reference signal via an exact robust differentiator which is based on high order sliding modes.

Carbajal-Espinoza et al [57] described in CGA a new 3D pose estimation of objects using reflectional symmetry. The authors present a real-time implementation for the pose estimation of objects tracked by a stereo vision system.

González-Jim énez *et al.* [106] proposed a controller, based on sliding mode control, for the n-link robotic manipulator pose tracking problem. The point pair is a geometric entity used to represent the position and orientation of the end-effector. A sliding mode controller of easy implementation is proposed which has the following properties: robustness against perturbations and parameter variations, as well as finite-time convergence.

Özgü and Mezouar [187] used screw theory expressed via unit dual quaternion representation to formulate efficiently both the forward position and velocity kinematics and the pose control of an n-DOF robot arm. The authors show that the efficiency is because of the reduced computer memory usage and the fast computation of the equations. The representation of task space is singularity-free and there is robustness to numerical errors. Finally, the approach yields compactness of the representations.

Takahashi *et al.* [222] studied the control performance of an adaptive controller using multilayer hypercomplex-valued neural networks, namely complex, hyperbolic, complex, and quaternion neural networks. The direct controller is synthesized online using the multilayer hypercomplex-valued neural network which adapts the desired plant output w.r.t a reference model.

Medrano and Bayro [173] discussed the design of controllers for robotics systems using Hamilton's equations. Unlike other works, they proposed to rebuild Hamilton's equations to an iterative form (for robots with any degrees of freedom) using the screw theory. Hence, the equations in the phase space are computed using screws and co-screws. For the controllers, they proposed two laws of control that ensure the convergence of the error to zero. The controllers are designed in terms of sliding mode control theory and on the laws of control using linear gains. The first theory gives robustness for the systems with matching perturbations, and the second one supplies speed for trajectory tracking. Then, to prove the stability of the proposed laws of control, the authors designed diverse Lyapunov functions with screws and co-screws to ensure that the robotic systems are globally asymptotically stable. The article shows a numeric example to illustrate the properties of the two designed controllers.

Bayro et al [45] formulated the Quaternion Spiking Neural Network (QSNN) extending the real-valued spiking neural networks using the quaternion algebra framework. The QSNN has remarkably the capacity to measure the parameters of the plant in question and accordingly adjust the parameters of the controller. The authors showed that the QSNN carries out both the plant estimation parameters and the adaptation of the controller parameter similar to an adaptive PID controller. Thus the QSNN makes it possible online that the robot reaches the desired position.

Arellano [8] proposed the dynamic model and applied nonlinear control for a quadrotor, This development was done using the motor algebra framework $G^+_{3,0,1}$. The kinematics for the quadrotor model and the dynamics based on Newton-Euler formalism are described. The authors applied block-control to the quadrotor model using super twisting control and also an estimator of the internal dynamics parameters to handle maneuvers detached from the origin.

Carbajal-Espinoza *et al.* [58] described a synthesis of the kinematic model of the pose of a 7-DOF robot manipulator using CGA. The authors proposed the error feedback and Lyapunov functions in terms of CGA. The authors believe that they are starting a new venue of research in control of robot manipulators and robot legs which can better be modeled in terms of geometric primitives like lines, circles, planes, spheres.

Gunn [109] in his Ph.D. thesis presented a modern formulation of rigid body mechanics in spaces of constant curvature. He developed the necessary theory – from projective geometry, exterior algebra, and quadratic forms – required to describe a class of Cayley-Klein spaces including the three classical spaces of constant curvature: Euclidean, elliptic, and hyperbolic. These n-dimensional Cayley-Klein geometries are then realized as real Clifford algebras constructed on the dual projective Grassmann algebras, of which only the euclidean case is degenerate. Poincare duality provides non-metric access to the standard Grassmann algebra. These Clifford algebras for n=2 and n=3 are described in detail. The role of non-simple bivectors and their connection to classical line geometry for n=3 receives particular attention.

Bayro *et al.* [46] proposed the robot dynamics in terms of Hamiltonians, which is different from the Lagrangian formulation of electromechanical or robotics systems. Using the iterative Newton-Euler, the local Hamiltonians are computed as well as the derivative of the moments at each robot joint. Decentralized controllers are applied at each joint.

Bayro and Osuna formulated the dynamic model and the trajectories were generated using quadratic programming with geometric constraints designed in CGA. The dynamic is represented in motor algebra using screw theory supported by an iterative Newton-Euler algorithm. The author implemented nonlinear controllers using integral sliding modes for collision avoidance. Experiments show the performance of modeling and the nonlinear control for tracking robot manipulators with perturbations.

### *M. ESTIMATION, POSE, HAND-EYE, RELOCALIZATION, KALMAN FILTER*

Stanway and Kinsey [221] presented the formulation of a stable adaptive identifier to estimate rigid body rotations using rotors in GA. The authors showed an experimental evaluation of this technique, which reduces dead reckoning navigation errors on these platforms and provides comparable performance to previously reported SO(3) constrained Linear Algebra (LA) approaches.

Rosenhahn, Perwass and Sommer [204] discussed the 2D–3D pose estimation for the case of 3D free-form contours. The authors derived cycloidal curves as orbits of coupled twist transformations. Furthermore, they represent the 3D contours in the spectral domain as an extension of cycloidal curves.

Vázquez and Bayro [228] presented the application of a new hypercomplex-valued Radial Basis Network (RBF) to estimate unknown geometric transformations such as in the case of the Hand-Eye Calibration problem.

Marin and Bayro [51] showed a novel approach for building 3D geometric maps using two sensors, a laser range finder, and a stereo camera system. The formulation is derived using CGA. The use of known visual landmarks in the map permits a good robot localization. In a previously captured environment, the approach uses landmarks for the robot relocalization.

Bashi and Kaminsky [32] evaluated the performance of an extended Kalman filter and a real-valued artificial neural network. These methods were compared for the processing accelerometer data collected during impact acceleration tests.

Bayro and Zhang [25] used the motor algebra for the linearization of the 3D Euclidean motion of lines. This motion model was used for the development of the Motor Extended Kalman Filter (MEKF). The MEKF estimates with high accuracy the relative position of the robot end-effector w.r.t. a 3D reference line. The authors claim that future vision systems can be reliably calibrated using the MEKF algorithm.

Xiaodong *et al.* [239], developed the quaternion-valued feedforward neural network (QFNN) to process 3D and 4D signals using the quaternion algebra. The authors used the unscented Kalman filter (WLQUKF) algorithm to train the QFNN. This method is formulated using recent studies in the augmented quaternion statistics and HR calculus. Due to the augmented quaternion statistics, the WLQUKF manages to process general quaternion-valued noncircular, nonlinear, and nonstationary signals, effectively.

### *N. BIOMEDICAL ENGINEERING AND BIOTECHNOLOGY*

Rivera-Rovelo *et al.* [203] developed a method using self-organizing neural networks for detecting shapes of 2D or 3D objects using a set of rigid transformations represented as versors in CGA. These transformations were applied to any geometric entity of GA which defines the object shape. This approach used haptic interfaces which provide the surgeon useful sensing information about the patient's body tissues.

Castillo-Muñiz [67] utilized for medical applications a haptic interface to get some sample points of an object surface or organ tissue. The authors use lines, circles, and spheres of CGA to represent organ shapes.

Bayro-Corrochano and Rivera-Rovelo [47] developed a novel approach for the modeling of 2D surfaces and 3D volumetric data. The modeling is based on the marching cubes algorithm using instead spheres and their representation in CGA. The authors present also a method for non-rigid registration of models based on spheres. For the registration, an annealing scheme is used, similar to the Thin-Plate Spline Robust Point Matching (TPS-RPM) algorithm. The authors present an interesting application in GA, namely the tracking of objects needed in minimally invasive surgical procedures.

Sepulveda-Cervantes and Portilla-Flores [209] introduced a novel method for haptic rendering contact force and surface properties for virtual objects using orthogonal decomposition in CGA. If we compare it with vector calculus, the CGA provides an easier and more intuitive language to deal with the problem of haptic rendering. The CGA is a sound geometric language due to its algebraic properties and a kind of framework which allows a simpler representation of geometric objects and their linear transformation.

Garza, Sanchez, and Bayro [99] reported geometric computing methods useful for medical robot vision. Through reformulating screw theory, seen as a generalization of quaternions, in CGA, the authors for neurosurgery solve the hand-eye calibration, the problem of the 3D registration using RGB-d cameras, efficient interpolation, and tracking.

Bayro-Corrochano, Lechuga-Gutierréz, and Garza-Burgos [44] proposed methods for the interpolation, virtual reality, graphics engineering, and haptics by reformulating screw theory in CGA. The authors formulated intuitive geometric equations to carry out surface operations as in kidney surgery. The interpolation is used for the interpolation and dilation in 3D of points, lines, planes, circles, and spheres. The authors interpolate trajectories of the surgical instrument.

Grafton and Lasenby [107] presented a method for representing surfaces using a set of dual quaternion control points, to fit to point clouds. Each control point is defined by a position and radius, which specify the area of the surface it affects, and a dual quaternion defining the transformation it applies. They fit surfaces to point clouds using a modified iterative gradient descent algorithm, adding control points to regions of the surface. These methods are applied to the problem of representing human breathing by fitting surfaces to a subject's chest as recorded by an RGB-D (image plus depth) camera and parameterizing the breathing using each control point's parameters. Variations in the breathing pattern are shown before and after exercise.

### *O. HARDWARE AND SOFTWARE FOR GEOMETRIC ALGEBRA*

In recent years, Franchini *et al.* developed efficient hardware to accelerate GA algorithms : a sliced co-processor for GA operations [90], an embedded FPGA-based computer graphics co-processor with GA support [91], fixed-size quadruples for a novel hardware-oriented representation of the 4D GA [92], design space exploration of parallel embedded architecture for GA operations [93], designed and implementation of an embedded co-processor with native support for 5D, quadruple-based GA [94], Conformal ALU: a CGA Co-processor for medical image processing [95] and a variety of embedded co-processors with GA support [96]. Recently, a coprocessor called GAPPCO easy to configure geometric algebra was developed by Hildenbrand *et al.* [129].

Soria-García *et al.* [219] reported an implementation of the conformal voting scheme utilizing reconfigurable hardware in $G_{3,1}$. This algorithm extracts from edge images geometric entities, like circles and lines. The authors derived the conformal voting scheme into two main stages: a local stage computed using neighborhoods in the image, and a global stage using the results of the local voting stage. The authors focused on the stage which requires the most computational demand using FPGA, while the global voting stage is executed on a PC.

Keninck and Dorst [147] introduced a novel and matrix-free implementation of the widely used Levenberg-Marquardt algorithm, in the language of Geometric Algebra. The resulting algorithm is shown to be compact, geometrically intuitive, numerically stable, and well-suited for efficient GPU implementation.

## IV. CONCLUSION AND PROSPECTS

This section summarizes the reviews and explains the current research challenges in the field of GA applications in engineering and computer science. The author makes use of scientometrics to analyze the quantitative features and characteristics in scientific publications. The study focuses on research works in which the development and mechanism of applied GA are studied by statistical mathematical methods.

This review presents a comprehensive study of works on applications of Quaternion Algebra and Geometric Algebra in computer science and engineering from 1995 to 2020. After an outline of geometric algebra, the application of GA has been analyzed across many fields and the advantages and shortcomings of the use of the geometric algebra framework has been shown. Furthermore, the challenges and prospects of various applications have been reviewed. Eyewitnessing the continuous developments using GA in image processing, computer vision, neurocomputing, quantum computing, robot modeling, control and tracking, improvement of computer hardware performance, we are convinced that the GA has proven to be the best geometric language available to tackle existing problems; therefore, one should further continue developing step by step GA-based algorithms. We believe that this review will help to orient and encourage researchers to continue in the progress of geometric computing for intelligent machines.

### *A. MODELS FOR EUCLIDEAN AND PSEUDO-EUCLIDEAN GEOMETRY*

While solving problems in engineering and computer science, an important issue which metric space we should use to represent models and compute algorithms. In this review, we focus on three well-understood space models:

i. Models for 2D and 3D spaces with a Euclidean metric: 2D and 3D are very useful to handle the algebra of directions in the plane and 3D physical space. For 3D rotations, rotors (isomorph to quaternions) are utilized. One can use $G_3$ to model the kinematics of points, lines, and planes. Rotors are well suited for the interpolation in graphics and the estimation of rigid body rotations.

ii. Models for 4D spaces with non-Euclidean metric: the linearization of a rigid motion transformation requires a homogeneous representation which can be formulated in the geometric algebra for the 4D space as the motor algebra $G^+_{3,0,1}$ This is the algebra of Plücker lines, useful to model the kinematics of points, lines, and planes, which is better than the 3D $G_3$. Note that lines belong to a manifold, the nonsingular study 6D quadric, and the motors belong to the manifold 8D Klein quadric. Thus in $G^+_{3,0,1}$, the motion for constant velocity can be modeled. The exponent of the model or twist is expressed in terms of bivector basis. Motors can be used to interpolate 3D rigid motion and to estimate trajectories using EKF techniques. For projective geometry problems like in computer vision, you require a homogeneous coordinate representation, thus the image plane is represented in $\mathbb{P}^2$ and the visual space in $\mathbb{P}^3$. The $n$-view geometry [116] is treated using tensor calculus and invariant theory, so you model the visual space in the $G_{3,1}$ (Minkowski metric) framework and $G_3$ for the image plane. The intrinsic camera parameters are modeled with an affine transformation within geometric algebra as part of the projective mapping. This mapping is formulated as the projective split between the projective space and the image plane. Furthermore, GA offers the Incidence Algebra, an algebra of oriented subspaces, which can be utilized in $G_{3,1}$ and $G_3$ frameworks to handle geometric constraints and invariant theory.

iii. In vector calculus, quaternion algebra, or linear algebra, the formulation of kinematics and dynamics involving only points and lines turns to be very complicated due to the following two reasons: by a formulation utilizing vectors, matrices and tensors, the practitioner has a poor insight of the geometry of the problem losing the intuition, as a result, this leads complex algebraic representations; second in the classical mathematical systems the computing involves redundant coefficients which unnecessarily slows down real-time computations. Classical approaches for kinematics and dynamics use vectors; and for linear transformations, matrices or tensors involve redundant coefficients. For example, a quaternion or a rotor has just four coefficients, in contrast, a 3D rotation matrix has nine coefficients. In the practice, if you compute reflections, inversions, rotations, and translations of entities like points, lines, planes, and spheres with matrices, the computation time grows more rapidly than when one utilizes rotors (quaternions) or motors (dual quaternions) instead. In geometric algebra, one utilizes 6-tuples to represent lines, twists, or wrenches. Circles, spheres and planes can be spanned in terms of k-vectors based on nonlinear representations. Recall that motor algebra was utilized to formulate the kinematics of robot manipulators using the points, lines and planes. Thereafter for the same goal, conformal geometric algebra was utilized also for the representation of circles and spheres. These additional geometric entities aid, even more, to eliminate redundant coefficients and diminish the computational complexity. Authors have proved that the mathematical treatment of dynamics is indeed much easier utilizing geometric primitives and spinors of the motor algebra framework. Classical approaches for the Newton mechanics and the Newton-Euler recursive algorithm use vectors for the screws and the linear transformation matrices. In contrast, the computation of the dynamics of robots can be carried out in the motor algebra framework using an iterative Newton-Euler algorithm in terms of screw theory. Moreover, the authors reformulated the Euler-Lagrange equations using the conformal geometric algebra framework. To generate robot navigation trajectories, authors resort to optimization using quadratic programming s.t geometric constraints. These constraints can be efficiently formulated in terms of screw theory in the conformal geometric algebra framework.

iv. Conformal models: for conformal transformations (angle preserving), authors use a non-Euclidean geometric algebra $G_{n,1}$ that extends its multivector basis with null vectors such as the origin and the point at infinity. Furthermore, it uses the computational framework called *horosphere*. This manifold is computed as the meet between a hyperplane and the null cone. Since $G_{n,1}$ uses a nonlinear representation for the geometric entities, one can compute the Euclidean metric via the inner product of null vectors. Be aware that the basic geometric entity of $G_{n,1}$ is the sphere. The geometric entities points, planes, lines, planes, circles, and spheres are represented in terms of vectors or their dual forms, the latter help to reduce the complexity of algebraic expressions. Interestingly enough, authors use $G_{n,1}$ either for kinematics in robotics or for projective geometry in computer vision. These approaches are promising, thus they have to be recognized by the community. If the digital camera is calibrated, one uses easily the homogeneous models from conformal geometric algebra to tackle simultaneously problems of robotics and computer vision without abandoning the mathematical framework. In addition, incidence algebra of points, lines, planes, circles, and spheres can be used in the conformal geometric framework as well.

v. Integral transforms: Clifford (geometric) algebra is a promising framework for fields such as image and signal processing. Researchers split often the correlation between the spatial domain and the temporal domain, unfortunately, they ignore the essential structural correlation. Using the wavelet transform, researchers can handle the information processing simultaneously in space and frequency. Important progress was achieved by the applications of the Quaternion Fourier Transform and Quaternion Wavelet Transforms and their generalization to Clifford Fourier Transformation. Many authors presented interesting applications of the QFFT for the case of color image processing. Recently some authors combine quaternion algebra with dictionary learning methods. Other researchers generalized the quaternion dictionary learning method using the octonion algebra framework. The octonion algebra combined with dictionary learning methods can be used for the representation of multispectral images with up to 7 color channels. Recently, authors proposed the Space-Time Split QFT for color image processing with pseudo-Euclidean metric using the computational framework light cone.

### B. TENDENCIES OF GA APPLICATIONS IN ENGINEERING AND COMPUTER SCIENCE

We carried out a scientometric analysis using Gephy https://gephi.org/. We selected 2,354 documents according to the following word filtering:
Year:-Since 1995-To 2020
Subject area:-“Mathematics”-“Engineering”-“Computer Science”
Document type:-“Article”-“Conference paper”-“Book chapter”-“Book”-“Editorial”
The results are shown in figure 2.a-c. Then taking into account 4,509 documents, we extended the word filtering to:
Article title, Abstract, Keywords: -“Geometric algebra”-“Clifford algebra”-“Conformal Geometric Algebra” -“Motor algebra”- “Quaternion Algebra”-“Quaternion”-“Dual quaternions”, for Figures 2.a-c
Article title, Abstract, Keywords: -“Geometric algebra”-“Clifford algebra”-“Conformal Geometric Algebra” -“Motor algebra”, for Figure 2.b

Figure 2.a shows from 1995 up to 2021 the increase in the publications in computer science and engineering using geometric algebra, quaternion algebra, and quaternions. The increase in 2008 and 2017 in Figure 2.a might be two increments due to more involvement of researchers using geometric algebra which was encouraged by the spread of geometric algebra due to conferences like each tree years International Conference in Applications of Clifford Algebras in Mathematics and Physics and the each three or four years Int. Cong. Applications of Geometric Algebra in Computer Science and Engineering AGACSE; also from 1995 until approximated 2007, researchers managed via publications in conferences and journals to make public to the community of computer science and engineering the new language of geometric algebra. Nowadays, there are plenty of reviewers in journals and conferences for articles in applications of geometric algebra, This process to show GA in the community took around 15 years. We believe that the growth in publications using quaternion algebra and quaternions shown in Figure 2.a is steady, in contrast, the growth in publications using geometric algebra shows an abrupt increase possible for the reasons given above.

In the last decades, pioneering groups have taken GA to handle challenging problems using a modern advance geometric language as GA. They are responsible for spreading worldwide the benefits of the use of GA. They organize the conferences ICCA and AGACSE and many summer schools, tutorials, and workshops as Sigraph, ICPR, and CIARP. Also, they published special issues in top journals and hold invited lectures at conferences. The Journal of Advances of Applications in Clifford Algebra started in the seventies and it publishes works in mathematics, physics, computer science, AI, and engineering.

The Figures 2.b shows the distribution around the world of the use of different geometric algebras for rotors, motors,

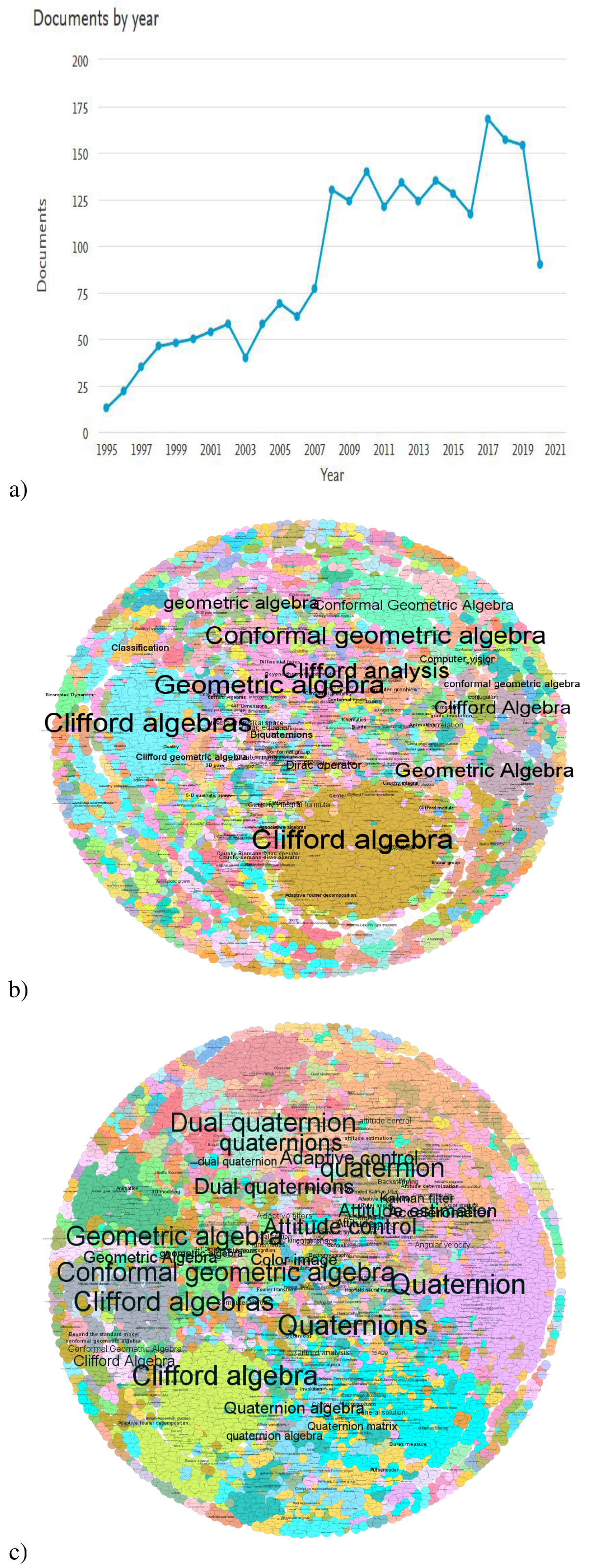



**FIGURE 2.** Documents: a) per year; b) geometric algebra, Clifford algebra, conformal geometric algebra, motor algebra; c) geometric algebra, Clifford algebra, conformal geometric algebra, motor algebra, quaternion algebra, quaternion, dual quaternions.

and conformal geometric algebra. The circles represent the relative amount of publications. Figures 2.c show that quaternion algebra is still the preferred framework to solve problems in computer science and engineering and it remains popular.

Nonetheless, after 1995, the use of geometric algebra first started and gradually expanded, as shown in Figure 2.a. It is important to clarify that geometric algebra allows the modeling of geometric entities such as points, lines, planes, circles, spheres, hyperplanes, and hyperspheres; it also allows the modeling of Lie groups and Lie algebras using bivector algebras of the general linear algebra in higher dimensions. Often authors use a matrix representation for quaternion computing, in contrast for twenty years, users compute in terms of multivectors, incidence algebra, and spinors like rotors and motors with bivectors avoiding entirely the use of vectors and matrices. Many researchers working in graphic engineering, computer vision, image processing, neural-computing, mechanics, robotics, control engineering, and mechatronics used quaternions for SO(3) and few works dual quaternions for SE(3). To deal with kinematics, differential kinematics, Euler-Lagrange, or Newton-Euler dynamics most of the researchers use Denavit-Hartenberg representation by homogeneous matrices, and the screw theory for lines, twists, and wrenches use 6D vectors, matrices, and tensors. In contrast, the screw theory can be elegant and computationally efficient using bivector algebra, even the tensors like inertia and Coriolis can be represented as spinors in $G_{9,3}$.

In control engineering, we have the Kalman filter which can be reformulated in geometric algebra as the Motor Extended Kalman filter. For closed-loop control, we require observers and controllers which should deal with nonlinear plants. So some pioneer works are trying to relate geometric algebra for the modeling with advanced methods for the design of observers and controllers for regulation and tracking, as well as the derivation of functionals for the optimal control of linear and nonlinear systems. To derive controllers and observers, adequate Lyapunov functions are needed to ensure stability. Here is an excellent opportunity for geometric algebra, namely to formulate the dynamics of the geometric entities like lines, planes, circles, spheres, hyperplanes, and hyperspheres which can be expressed as inner products and used as arguments of the Liapunov functions: e.g. the projection center of mass to screw lines, planes or circles at the joints of a robot manipulator.

On the other hand, in computer vision, graphic engineering, and GIS, researchers use matrix algebra and vector calculus and tensor; however, the algebra of incidence of geometric algebra allows the modeling the conformal or projective mapping as in the n-view geometry using points, lines, planes, circles and spheres utilizing the duality operator or Pseudoscalar in the Outer Product Null Space. Regarding image processing, since the nineties, researchers have formulated the Quaternion Fourier transform, where the kernel is in terms of the Lie group SO(3) quaternion algebra. To extend these results, the Clifford Fourier Transforms can use different kernels in terms of bivectors, and very importantly, one can change the metric from Euclidean to pseudo-Euclidean. One new result is the Quaternion Split Fourier transform [43]. In neurocomputing, also since the nineties, the researchers have used complex numbers and quaternions. Again, geometric algebra allows generalizing of the real-valued neural networks and even deep learning with geometric algebra neural networks as Clifford SVMs or conformal neural networks. Recent works are using quaternion to formulate the quaternion quantum neural network and the quaternion quantum Fourier transform. The venue is open to developing the quaternion Radon transform for 3D reconstruction.

### C. DIFFICULTIES IN THE USE OF GA FOR APPLICATIONS

Since the influence of geometric algebra in applications in computer science and engineering is still not as expected, thus it is important to explain that the community due to tradition is focused on the use of quaternion algebra, in this regard when is needed the community should leave the use of quaternion algebra and learn to work with the geometric algebra framework for a geometric way to compute geometric entities, incidence algebra and formulate the linear transformations, Lie groups in terms of bivectors.

In general, GA doesn't use vector calculus and matrices, this avoids processing redundant parameters of the matrices, as a result even working in higher dimensions, for linearizing functions and separate clusters for linear hyperplane boundaries, GA offers more efficient and fast algorithms. As long as the community adopts GA, which substitutes quaternion algebra and even better generalizes it over hypercomplex numbers with a more rich geometric interpretation, the user will utilize a geometric language more powerful for fairly challenging applications. In addition, the use of FUGAL and CUDA-Nvidia and parallel processors can boost the performance of GA algorithms.

We have to admit that GA is not the solution for everything, that depends upon the problem better said use GA when is needed. GA helps to have a better geometric insight of the problem in question and therefore to develop efficient algorithms. If the problem is not too complex you may still use matrix algebra or quaternion algebra, however generally speaking in the GA framework the practitioner discovers other geometric aspects which remain hidden or obscured by the use of vector calculus, matrix algebra, and even quaternion algebra. As an example Rotor algebra isomorph to quaternion algebra has a geometric interpretation, i.e. rotors for rotations are seen as geometric objects which can be also used for screw theory in higher dimensions. When you are solving complex problems, it is favorable to find geometric constraints, so that you can reduce the dimension of the solution space. GA is very useful for this purpose, e.g. in optimization, you can define the quadratic programming subject to geometric constraints formulated in conformal geometric algebra.

It is worth to mention, that the interest of the community in GA grows slowly. This is because GA is being spread by pioneers and it has not yet reached the whole community. Furthermore, it is not easy to become acquainted with GA and start rapidly to write GA algorithms. So the practitioners looking for easy solutions become scared to embark themselves in a GA project which demands study GA, learn to think using the GA framework, and most important program GA algorithms which are completely different from the algorithms using traditional methods. This is one of the most noticeable reasons, why the progress in GA applications is not fast as expected. We are hoping that this review will encourage researchers to start research using the GA framework.

### D. PROSPECTS

The progress and tendencies in applications of GA in engineering and computer science are shown in previous figures and one can recognize an important aspect that takes place: namely, that there is an interaction bridge between geometric algebra for the treatment and processing of geometric objects and the signal and image processing using Clifford algebra. We believe that both mathematical frameworks are complementary and in their interaction the methods profit and become more robust.

In the beginning, for computer vision, $G_{3,1}$ was used for representing the visual space, and $G_3$ for the image plane. The intrinsic camera parameters are modeled by an affine transformation within geometric algebra as part of the projective mapping. This mapping is a result of the projective split between the projective space and the image plane. With the new RGB-D, one gets directly the 300,000 3D points per frame 25-30 frames per second of the visual space; thus, one is freed to compute the affine transformation of the intrinsic cameras for each image capture as when we handle the n-view geometry [116]. This opens the avenue for the efficient use of conformal geometric algebra $G_{4,1}$ as we only need rotors, translators, and dilators that correspond to components of the similitude Lie group. There are already works for computer vision which is using the $G_{4,1}$ for the visual space and the $G_{3,1}$ of the image plane.

The fields of computer vision, graphic engineering, GIS, and real-time computing for deep-learning, big data, and data analytics can benefit from the progress in low-energy consumption and extremely fast hardware which uses CUDA Nvidia. This progress in hardware opens new opportunities for geometric algebra, which needs powerful geometric computing accelerators connected with the CPU of computers. Here, one must highlight the role of high dimensional geometric algebras with Euclidean and pseudo-Euclidean metrics for future advanced computing. One limiting aspect ten years ago was that practitioners were restricted to the use of lower dimension algebras for solving problems in real-time. Nowadays, we should exploit the progress in hardware and on the Internet like G5 for cloud computing and ubiquitous computing using the geometric algebras in higher dimensions like $G_{6,0,2}$, $G_{6,3}$ $G^+_{9,3}$ $G^+_{6,0,6}$ and in general other geometric algebras $G_{n,m}$ for $n, m \geq 5$.

In robotics, the modeling and control, and tracking have been formulated using the motor algebra $G^+_{3,0,1}$ and conformal geometric algebra $G_{4,1}$. The Newton-Euler dynamics was computed using screw theory in $G^+_{3,0,1}$. The Euler-Lagrange dynamic equations have been formulated in $G_{4,1}$ and used for modeling the kinematics and dynamics of humanoid robots. There have been interesting works to compute the direct and inverse kinematics of reconfigurable robots using screw theory in terms of quaternions (rotors).

The Newton-Euler dynamic of quadrotors was modeled using motor algebra. The modeling of robots using quaternion, dual quaternion, motor algebra, or conformal geometric in terms of points, lines, planes, circles, and spheres and screw theory is a trend that started in the eighties. The results are remarkable; however, the design of controllers and observers still need to be carried out in the geometric algebra framework, so that the modeling, control, and tracking can be done in the same framework. It is needed to relate the robot modeling using geometric algebra with classical control engineering, such as adaptive control, nonlinear control, sliding modes control, fuzzy nonlinear control, geometric Lyapunov functions for stability analysis, tracking, optimization using quadratic programming with geometric constraints derived from using conformal geometric algebra.

In neurocomputing, since the nineties, the models such as RBF, MLP, SOM, and SVM have been formulated using quaternions or Clifford (geometric) algebra for classification, regression, and recurrence. However, CNNs showed the community the spectacular power of deep learning due to the use of many filters as kernels along with the layers; thus, modules need to be incorporated into these architectures to handle the geometry in signal processing.[c] Since the layers care for feature extraction, regressors and estimators of 3D motions need to be connected at the output by using rotors (quaternions) or motors (dual quaternions) or at the output using sphere neurons of conformal geometric algebra. Generally speaking, the feature extraction should be done using real-valued deep learning and only at the output with few layers using geometric neurons in Euclidean or pseudo-Euclidean geometric algebras for the different tasks as estimation, classification, regression, and recurrence. At these modules, feedback for space-time patterns can also be used.

Many works show that Integral transforms formulated in Clifford (geometric) algebra are very useful for signal and image processing. The wavelet transform can handle the information processing simultaneously in space and frequency. One great progress was the introduction of Quaternion Fourier Transform and Quaternion Wavelet Transforms and their generalization to Clifford Fourier Transformation. A recent tendency in color image processing is to combine quaternion algebra with dictionary learning methods. Researchers generalized the quaternion dictionary learning method using octonion algebra. The octonion algebra together with dictionary learning methods is very efficient for the representation of multispectral images with up to seven color channels. Recent works show the Space-Time Split QFT for handling color image processing using pseudo-Euclidean metric using the light cone. Research has to continue formulating the quaternion fractional Fourier transform, quantum quaternion Fourier transform and quaternion Radon transform for 3D processing and solving problems in the processing of microscope, astronomy, medical images, as well as the processing and filtering of color images.

Regarding the development of geometric computing accelerators in a geometric algebra with certain metric, we have to split and specialize both the representation and the operation computations. Since multivectors are too big; the chosen metric can indeed constrain even more the world representation but it is unfortunately not enough for efficient and fast computation. Note that the operations on multivectors are in general simple and universal, however too slow if they are not specialized, optimized, and formulated in such a manner, that one can take advantage of cost-effective hardware to speed up the computations.

We should utilize modern hardware and the Internet like G5 for cloud computing and ubiquitous computing using the geometric algebras in higher dimensions, $G_{p,q,r}$, $n = p + q + r$ n>5, for geometric computing to solve difficult problems. It is well known if you cast your problem in higher dimensions you linearize functions and separate data clusters. The curve of hardware progress also requires progress in algorithm efficiency, we believe that geometric algebra can indeed be helpful for the development of more efficient algorithms.

## APPENDIX I: GLOSSARY

Vector spaces and multivector geometric algebras.

| | |
|---|---|
| $\mathbb{R}$ | The real numbers |
| $\mathbb{S}^n$ | The unit sphere in $\mathbb{R}^{n+1}$ |
| $\mathbb{C}$ | The complex numbers |
| $\mathbb{H}$ | The quaternion algebra |
| $\mathbb{R}^n$ | A vector space of dimension $n$ over the field $\mathbb{R}$ with Euclidean signature |
| $\mathbb{R}^{p,q}$ | A vector space of dimension $n = p + q$ over the field $\mathbb{R}$ with signature $(p, q)$ |
| $\mathbb{R}^{m \times n}$ | The direct product $\mathbb{R}^m \otimes \mathbb{R}^n$ |
| $\mathcal{G}_n$ | The geometric algebra over $\mathbb{R}^n$ |
| $\mathcal{G}_{p,q}$ | The geometric algebra over $\mathbb{R}^{p,q}$ |
| $\mathcal{G}_{p,q,r}$ | The special or degenerated geometric algebra over $\mathbb{R}^{p,q}$ |
| $\mathcal{G}^k_{p,q}$ | The $k$-vector space of $\mathcal{G}_{p,q}$ |
| $x$ | Scalar element |
| $\mathbf{x}$ | vector of $\mathbb{R}^n$ |
| $\boldsymbol{x}$, $\boldsymbol{X}$ | Vector and multivector of geometric algebra |
| $\boldsymbol{X}_k$ | A blade of grade $k$ |
| $\mathcal{F}$ | Flag, a set of geometric entities of an object frame |

Mulivector operator symbols.

| | |
|---|---|
| $\boldsymbol{X}\boldsymbol{Y}$ | Geometric product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ |
| $\boldsymbol{X} * \boldsymbol{Y}$ | Scalar product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ |
| $\boldsymbol{X} \cdot \boldsymbol{Y}$ | Inner product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ |
| $\boldsymbol{X} \wedge \boldsymbol{Y}$ | Wedge product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ |
| $\boldsymbol{X} \bigvee \boldsymbol{Y}$ | Meet operation of $\boldsymbol{X}$ and $\boldsymbol{Y}$ |
| $\boldsymbol{X} \bigwedge \boldsymbol{Y}$ | Join product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ |
| $\boldsymbol{X}^{-1}$ | Inverse of $\boldsymbol{X}$ |
| $< \boldsymbol{X} >_k$ | Projection of $\boldsymbol{X}$ onto grade $k$ |
| $\boldsymbol{X}^*$ | Dual of $\boldsymbol{X}$ |
| $\widetilde{\boldsymbol{X}}$ | Reverse of $\boldsymbol{X}$ |
| $\boldsymbol{X}^\dagger$ | Conjugate of $\boldsymbol{X}$ |
| $\Vert \boldsymbol{X} \Vert$ | Norm of $\boldsymbol{X}$ |
| $P^{\Vert}_{<\boldsymbol{Y}>_l}(< \boldsymbol{X} >_k)$ | Projection of $< \boldsymbol{X} >_k$ onto $< \boldsymbol{Y} >_l$ |
| $P^{\perp}_{<\boldsymbol{Y}>_l}(< \boldsymbol{X} >_k)$ | Projection of $< \boldsymbol{X} >_k$ onto $< \boldsymbol{Y} >_l$ |

## APPENDIX II: PACKAGES FOR APPLICATIONS OF GEOMETRIC ALGEBRA IN MATHEMATICS, PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Lounesto as a pioneer developed in 1987 the software package CLICAL for Clifford (Geometric) algebra computing. CLICAL is Fortran based and it is useful for fast computation and theorem proving, see

http://users.tkk.fi/ppuska/mirror/Lounesto/CLICAL.htm

Thereafter many researchers have been developing software packets for multivector programming, for example

The Matlab-based geometric algebra tutorial GABLE supports $N \leq 3$, see

http://staff.science.uva.nl/ leo/GABLE/index.html

The Maple-based CLIFFORD supports $N \leq 9$. It can be used for symbolic programming and theorem proving, see

http://math.tntech.edu/rafal

The $C$++-based CLUCal is handy for computer scientists and engineers and for all who want to learn geometric algebra computations in 2D and 3D, especially for visualization, computer vision, and crystallography: former link

http://www.perwass.de/cbup/clu.html

GAIGEN2 generates fast $C$++ or JAVA sources for geometric algebras $G_2$, $G_3$, $G_4$. It is a user-friendly package for learning geometric algebra computing and for handling a variety of problems in computer science and graphics, see

http://www.science.uva.nl/ga/gaigen/

Recently the ganja tool was developed by Steven de Kenninck. It seems very suitable for learning GA, see

https://observablehq.com/@enkimute/ganja-js-introduction

$C$++ MV 1.3.0 sources supporting $N \leq 63$. Ian Bell developed it up to 1.6 with significant functionality extensions and bug fixes. This program is a powerful multivector software for applications in physics, computer science and engineering. It is a good source of inspiration for writing one's code, see

http://www.iancgbell.clara.net/maths/index.htm

The $C$++ GEOMA v1.2 developed by Patrick Stein contains $C$++ libraries for Clifford algebra with an orthonormal basis, see

http://nklein.com/software/geoma

The reader can also download our $C++$ programs, which are being routinely updated for applications in robotics, image processing, wavelets transforms, computer vision, neural computing, and medical robotics, see

http://www.gdl.cinvestav.mx/edb/GAprogramming

GAALOP is an optimizer for different programming languages such as C/C++, Python, Mathematica, and Matlab. It was developed by Hildenbrand, Pitt, and Koch [128], see

http://www.gaalop.de

Recent packages for conformal geometric algebra:
Chaïm Zonnenberg,
http://www.cs.uu.nl/groups/MG/gallery/CGAP/index.html
Jose L. Aragon, A Mathematica package for conformal geometric algebra: https://arxiv.org/abs/1711.02513
Paolo Colapinto, Versor (libvsr) A (fast) Generic C++ library for Geometric Algebras, including Euclidean, Projective, Conformal, Spacetime (etc). http://versor.mat.ucsb.edu/
Following sites offer information and tools for geometric algebra:
bivector.net (Geometric Algebra various tools and info)
enkimute.github.io/ganja.js (especially the Coffee Shop: https://enkimute.github.io/ganja.js/examples/coffeeshop.html)
clifford.readthedocs.io

Hosny Eid [136] provided a high-level introduction to the abstract concepts and algebraic representations behind the elegant GA mathematical structure. His article focuses on the conceptual and representational abstraction levels behind GA mathematics with sufficient references for more details. In addition, his article strongly recommends applying the methods of Computational Thinking in both introducing GA to software engineers, and in using GA as a mathematical language for developing Geometric Computing software systems.

We suggest that the readers who want to develop their own program for Clifford or geometric algebra applications, learn from the cited multivector software packets and integrate these new developments into their programs. CLUcal and GAIGEN are highly recommended for learning geometric algebra. To write a $C++$ geometric algebra program, one should start by looking at GEOMA, the MV 1.3.0, the code generator of GAIGEN or visit our homepage. For symbolic computing and theorem proving, CLICAL and Maple-based CLIFFORD are the best software packages.

Concluding geometric algebra is a language for modeling geometric primitives and their transformations as SO(3), SE(3), and Conformal transformations and the computing of intersections using Incidence Algebra. However, this doesn't mean that the whole problem should be formulated and computed in GA, e.g. solution of the equation has to be done using standard numerical algorithms in C++ or Python. The user has to use GA where is needed. We can say that GA is a metalanguage to tackle key geometric issues of the problems. GA is a convenient tool at some phases of solving a problem. In the review of works, you can see how the authors proceed in solving problems, thus you learn much from those works. Thanks to the progress in GA software packages and hardware to speed up the computation, the user can resort to libraries for different geometric algebras $G_{p,q,r}$ and accelerate their computing. There is no need to reformulate the equations in terms of vectors or matrices. The equations given in section II can directly implement with these software packages and speed up them using FPGA, CUDA with Nvidia.

## REFERENCES


[1] R. Ablamowicks. eCLIFFORD Software Packet Using Maple for Clifford Algebra. Computations. Accessed: Sep. 10, 2019. [Online]. Available: http://math.tntech.edu/rafal

[2] E. T. Abou El Dahab, "A formulation of Hamiltonian mechanics using geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 10, no. 2, pp. 217–223, Sep. 2000.

[3] H. Adeli and A. Panakkat, "A probabilistic neural network for Earthquake magnitude prediction," *Neural Netw.*, vol. 22, no. 7, pp. 1018–1024, Sep. 2009.

[4] A. Al-Nuaimi, E. Steinbach, . WB. Lopes, and C. G. Lopes, "6DOF point cloud alignment using geometric algebra-based adaptive filtering," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[5] G. Altamirano-Gomez and E. Bayro-Corrochano, "Conformal geometric method for voting," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (Lecture Notes in Computer Science), vol. 8827, E. Bayro-Corrochano and E. Hancock, Eds. Cham, Switzerland: Springer, 2014, pp. 802–809.

[6] P. Arena, R. Caponetto, L. Fortuna, G. Muscato, and M. G. Xibilia, "Quaternionic multilayer perceptrons for chaotic time series prediction," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 79, no. 10, pp. 1682–1688, 1996.

[7] A. Aristidou and J. Lasenby, "Inverse kinematics solutions using conformal geometric algebra," in *Guide to Geometric Algebra in Practice*, L. Dorst and J. Lasenby, Eds. Berlin, Germany: Springer, 2011, ch. 3, pp. 47–62.

[8] C. Arellano-Muro, G. Osuna-Gonzalez, B. Castillo-Toledo, and E. Bayro-Corrochano, "Newton–Euler modeling and control of a multicopter using motor algebra $G^+_{3,0,1}$," *Adv. Appl. Clifford Algebras*, vol. 30, p. 19, Feb. 2020.

[9] A. Arsenovic, "Applications of conformal geometric algebra to transmission line theory," *IEEE Access*, vol. 5, pp. 19920–19941, 2017.

[10] M. A. J. Ashdown. (1998). *Maple code for Geometric Algebra*. [Online]. Available: http://www.mrao.cam.ac.uk/emaja

[11] W.E. Baylis, *Clifford (Geometric) Algebras: With Applications to Physics, Mathematics, and Engineering*. Basel, Switzerland: Birkhäuser, 1996.

[12] T. Batard, M. Berthier, and C. Saint-Jean, "Clifford–Fourier transform for color image processing," in *Geometric Algebra Computing in Engineering and Computer Science*, E. J. Bayro-Corrochano and G. Scheuermann, Eds. Berlin, Germany: Springer, 2010, ch. 8, pp. 135–161.

[13] T. Batard and M. Berthier, "The spinor representation of images," in *Quaternion and Clifford Fourier Transforms and Wavelets, Trends in Mathematics (TIM)*, S. J. Sangwine, Eds. Basel, Switzerland: Birkhäuser, 2013.

[14] E. Bayro-Corrochano and G. Sommer, "Object modelling and collision avoidance using Clifford algebra," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, Sep. 1995, pp. 699–704.

[15] E. Bayro-Corrochano and J. Lasenby 1997, "A unified language for computer vision and robotics," in *Proc. Int. Workshop Algebr. Frames Perception-Action Cycle* (Lecture Notes in Computer Science), vol. 1315. Kiel, Germany: Springer, Sep. 1997, pp. 219–234.

[16] E. Bayro-Corrochano and J. Lasenby, "Object modelling and motion analysis using Clifford algebra," in *Proc. Eur.-China Workshop Geometric Modeling Invariants Comput. Vis.*, Xi'an, China, Apr. 1995, pp. 143–149.

[17] E. Bayro-Corrochano and J. Lasenby, "What can Hamilton, Grassman and Clifford tell us about computer vision and robotics?" in *Proc. 19th DAGM Symp. Pattern Recognit.*, Brunswick, Germany, Sep. 1997, pp. 164–171.

[18] E. Bayro-Corrochano and J. Lasenby, "Geometric techniques for the computation of projective invariants using n uncalibrated cameras," in *1st Indian Conf. Comput. Vis. Image Process.*, New Delhi, India, Dec. 1998, pp. 95–100.
[19] E. Bayro-Corrochano and D. Kähler, "Motor algebra approach for computing the kinematics of robot manipulators," *J. Robotic Syst.*, vol. 17, no. 9, pp. 495–516, 2000.
[20] E. Bayro Corrochano, S. Buchholz, and G. Sommer, "Selforganizing clifford neural network," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Washington, DC, USA, vol. 1, 1996, pp. 120–125.
[21] E. Bayro-Corrochano, S. Buchholz, and G. Sommer, "A new self-organizing neural network using geometric algebra," in *Proc. 13th Int. Conf. Pattern Recognit.*, Vienna, Austria, vol. 4, 1996, pp. 120–125.
[22] E. Bayro-Corrochano, S. Buchholz, and G. Sommer, "A new self-organizing neural network using geometric algebra," in *Proc. Int. Neural Netw. Soc. Annu. Meeting: World Congr. Neural Netw. (WCNN)*, San Diego, CA, USA, Aug. 1996, pp. 245–249.
[23] E. Bayro-Corrochano and S. Buchholz, "Geometric neural networks," in *Proc. Int. Workshop Algebraic Frames Perception-Action Cycle* (Lecture Notes in Computer Science), vol. 1315. Kiel, Germany: Springer, Sep. 1997, pp. 379–394.
[24] E. Bayro-Corrochano, "Clifford self-organizing neural network, Clifford wavelet network," in *Proc. 14th IASTED Int. Conf. Appl. Inf.*, Innsbruck, Austria, Feb. 1996, pp. 271–274.
[25] E. Bayro-Corrochano and Y. Zhang, "The motor extended Kalman filter: A geometric approach for rigid motion estimation," *J. Math. Imag. Vis.*, vol. 13, pp. 205–228, Dec. 2000.
[26] E. J. Bayro-Corrochano, "Geometric neural computing," *IEEE Trans. Neural Netw.*, vol. 12, no. 5, pp. 968–986, Sep. 2001.
[27] E. Bayro-Corrochano, K. Daniilidis, and G. Sommer, "Motor algebra for 3D kinematics: The case of the hand-eye calibration," *J. Math. Imag. Vis.*, vol. 13, no. 2, pp. 79–99, 2000.
[28] E. Bayro-Corrochano, *Geometric Computing For Perception Action Systems*. New York, NY, USA: Springer-Verlag, 2001.
[29] E. Bayro-Corrochano and V. Banarer, "A geometric approach for the theory and applications of 3D projective invariants," *J. Math. Imag. Vis.*, vol. 16, pp. 131–154, Mar. 2002.
[30] E. Bayro-Corrochano, L. Reyes-Lozano, and J. Zamora-Esquivel, "Conformal geometric algebra for robotic vision," *J. Math. Imag. Vis.*, vol. 24, no. 1, pp. 55–81, Jan. 2006.
[31] E. Bayro-Corrochano, "The theory and use of the quaternion wavelet transform," *J. Math. Imag. Vis.*, vol. 24, no. 1, pp. 19–35, Jan. 2006.
[32] A. S. Bashi and E. J. Kaminsky, "Comparison of neural network and extended Kalman filter determination of kinematics from impact acceleration tests," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Comput. Cybern. Simulation*, Orlando, FL, USA, vol. 4, Oct. 1997, pp. 3501–3506.
[33] E. Bayro-Corrochano and L. E. Falcón, "Geometric algebra of points, lines, planes and spheres for computer vision and robotics," *Robotica*, vol. 23, no. 6, pp. 755–770, Nov. 2005.
[34] E. Bayro-Corrochano, J. R. Vallejo-Gutierrez, and N. Arana-Daniel, "Recurrent clifford support machines," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 3613–3618.
[35] E. Bayro-Corrochano and A. Arana-Daniel, "Clifford support vector machines for classification, regression, and recurrence," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1731–1746, Nov. 2010.
[36] E. Bayro-Corrochano and E. U. Moya-Sánchez, "Quaternion atomic function for image processing," in *Guide to Geometric Algebra in Practice*, L. Dorst and J. Lasenby, Eds. Berlin, Germany: Springer, 2011, ch. 8, pp. 151–173.
[37] E. Bayro-Corrochano and S. Gamboa, "Quaternion quantum neural computing," *Int. J. Wavelets, Multiresolution Inf. Process.*, 2020.
[38] E. Bayro-Corrochano and G. Scheuermann, Eds., *Geometric Algebra Computing in Engineering and Computer Science*, Berlin, Germany: Springer, 2010.
[39] E. Bayro-Corrochano, *Geometric Computing: For Wavelet Transforms, Robot Vision, Learning, Control and Action*. London, U.K.: Springer-Verlag, 2010.
[40] E. Bayro-Corrochano, E. Vazquez-Santacruz, E. Moya-Sanchez, and E. Castillo-Munis, "Geometric bioinspired networks for recognition of 2-D and 3-D low-level structures and transformations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 2020–2034, Oct. 2016.
[41] E. Bayro-Corrochano, *Geometric Algebra Applications: Computer Vision, Graphics and Neurocomputing*, vol. 1. Berlin, Germany: Springer-Verlag, 2019.
[42] E. Bayro-Corrochano, *Geometric Algebra Applications: Robot Modeling and Control*. Berlin, Germany: Springer-Verlag, Heidelberg, 2020.
[43] E. Bayro-Corrochano, U. Uriostegui-Legorreta, and Z. Vazquez, "The Lorentz group in conformal geometric algebra and split quaternions for color image processing: Theory and practice," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.
[44] E. Bayro-Corrochano, L. Lechuga-Gutiérrez, and M. Garza-Burgos, "Geometric techniques for robotics and HMI: Interpolation and haptics in conformal geometric algebra and control using quaternion spike neural networks," *Robot. Auton. Syst.*, vol. 104, pp. 72–84, Jun. 2018.
[45] E. Bayro-Corrochano, S. Solis-Gamboa, G. Altamirano-Escobedo, L. Lechuga-Gutierres, and J. Lisarraga-Rodriguez, "Quaternion-spike and quaternion-quantum neural networks: Theory and applications," *Int. J. Neural Syst.*, vol. 31, no. 2, 2021, Art. no. 2050059.
[46] E. Bayro-Corrochano, J. Medrano-Hermosillo, G. Osuna-González, and U. Uriostegui-Legorreta, "Newton-Euler modeling and Hamiltonians for robot control in the geometric algebra," *J. Robot.*, 2021.
[47] E. Bayro-Corrochano and J. Rivera-Rovelo, "The use of geometric algebra for 3D modeling and registration of medical data," *J. Math. Imag. Vis.*, vol. 34, no. 1, pp. 48–60, May 2009.
[48] M. Bernal-Marin and E. Bayro-Corrochano, "Integration of Hough transform of lines and planes in the framework of conformal geometric algebra for 2D and 3D robot vision," *Pattern Recognit. Lett.*, vol. 32, no. 16, pp. 2213–2223, Dec. 2011.
[49] I. Bell. *C ++ MV 1.3.0 to 1.6 Sources Supporting N $\leq$ 63*. Accessed: Aug. 16, 2008. [Online]. Available: http://www.iancgbell.clara.net/maths/index.htm
[50] S. Berman, D. G. Liebermann, and T. Flash, "Application of motor algebra to the analysis of human arm movements," *Robotica*, vol. 26, no. 4, pp. 435–451, Jul. 2008.
[51] M. Bernal-Marin and E. Bayro-Corrochano, "Visual and laser guided robot relocalization using lines, Hough transformation and machine learning techniques," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 4142–4147.
[52] U. A. Bhatti, Z. Yu, L. Yuan, Z. Zeeshan, S. A. Nawaz, M. Bhatti, A. Q. U. Mehmood Ain, and L. Wen, "Geometric algebra applications in geospatial artificial intelligence and remote sensing image processing," *IEEE Access*, vol. 8, pp. 155783–155796, 2020.
[53] F. Brackx, F. R. Delanghe, and F. Sommen, *Clifford Analysis, Research Notes in Mathematics*, vol. 76. Bath, U.K.: Pitman Books Limited, 1982.
[54] F. Brackx, N. De Schepper, and F. Sommen, "The Clifford-Fourier transform," *J. Fourier Anal. Appl.*, vol. 11, no. 6, pp. 669–681, 2005.
[55] T. Bülow, "Hypercomplex Fourier transforms," Ph.D. dissertation, Comput. Sci. Inst., Christian Albrechts Universität, Kiel, Germany, 1999.
[56] C. Cafaro and S. Mancini, "A geometric algebra perspective on quantum computational gates and universality in quantum computing," 2010, *arXiv:1006.2071*. [Online]. Available: http://arxiv.org/abs/1006.2071
[57] O. Carbajal-Espinosa, G. Osuna-González, L. Gonález-Jiménez, A. Loukianov, and E. Bayro-Corrochano, "Visual servoing and robust object manipulation using symmetries and conformai geometric algebra," in *Proc. 14th IEEE-RAS Int. Conf. Humanoid Robots*, Madrid, Spain, Nov. 2014, pp. 1051–1054.
[58] O. Carbajal-Espinosa, L. González-Jiménez, J. Obviedo-Barriga, B. Castillo-Toledo, A. Loukianov, and E. Bayro-Corrochano, "Modeling and pose control of robotic manipulators and legs using conformal geometric algebra," *Computación y Sistemas*, vol. 19, no. 3, pp. 475–486, Oct. 2015.
[59] M. Castilla, J. C. Bravo, M. Ordonez, and J. C. Montano, "Clifford theory: A geometrical interpretation of multivectorial apparent power," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 10, pp. 3358–3367, Nov. 2008.
[60] M. Castilla, J. C. Bravo, and M. Ordonez, "Geometric algebra: A multivectorial proof of Tellegen's theorem in multiterminal networks," *IET Circuits, Devices Syst.*, vol. 2, no. 4, pp. 383–390, Aug. 2008.

[61] M. V. Castilla, J. C. Bravo, and F. I. Martin, "Multivectorial strategy to interpret a resistive behaviour of loads in smart buildings," in *Proc. IEEE 12th Int. Conf. Compat., Power Electron. Power Eng. (CPE-POWERENG)*, Apr. 2018, pp. 1–5.

[62] M. Castro-Núñez and R. Castro-Puche, "The IEEE Standard 1459, the CPC power theory, and geometric algebra in circuits with nonsinusoidal sources and linear loads," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, p. 12, pp. 2980–2990, Dec. 2012.

[63] M. Castro-Núñez and R. Castro-Puche, "Advantages of geometric algebra over complex numbers in the analysis of networks with nonsinusoidal sources and linear loads," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, p. 9, pp. 2056–2064, Sep. 2012.

[64] M. Castro-Núñez, D. Londoño-Monsalve, and R. Castro-Puche, "$M$, the conservative power quantity based on the flow of energy," *J. Eng.*, vol. 2016, no. 7, pp. 269–276, 2016.

[65] M. Castro-Núñez, D. Londoño-Monsalve, and R. Castro-Puche, "Theorems of compensation and Tellegen in non-sinusoidal circuits via geometric algebra," *J. Eng.*, vol. 2019, no. 4, pp. 3409–3417, 2019.

[66] M. Castro-Núñez, R. Castro-Puche, and E. Nowicki, "The use of geometric algebra in circuit analysis and its impact on the definition of power," in *Proc. Int. School Nonsinusoidal Currents Compensation*, Jun. 2010, pp. 89–95.

[67] E. Castillo-Muñiz, J. Rivera-Rovelo, and E. Bayro-Corrochano, "Using haptic and neural networks for surface and mechanical properties 3D reconstruction," in *Ubiquitous Computing and Ambient Intelligence (UCAmI)*, (Lecture Notes in Computer Science), vol. 7656, J. Bravo, D. López-de-Ipiña, and F. Moya, Eds. Springer, 2012, pp. 422–429.

[68] W. K. Clifford, "Preliminary sketch of bi-quaternions," *Proc. London Math. Soc.*, vol. 4, pp. 381–395, Mar. 1873.

[69] C. Gaudet and A. Maida, "Deep quaternion networks," 2017, *arXiv:1712.04604*. [Online]. Available: http://arxiv.org/abs/1712.04604

[70] J. M. Chappell, S. P. Drake, C. L. Seidel, L. J. Gunn, A. Iqbal, A. Allison, and D. Abbott, "Geometric algebra for electrical and electronic engineers," *Proc. IEEE*, vol. 102, no. 9, pp. 1340–1363, Sep. 2014.

[71] V. M. Chernov, "Discrete orthogonal transforms with data representation in composition algebras," in *Proc. Scandin. Conf. Image Anal.*, Uppsala, Sweden, 1995, pp. 357–364.

[72] E. Conte, "What is the reason to use Clifford algebra in quantum cognition? Part I:'It from qubit' on the possibility that the amino acids can discern between two quantum spin states," *NeuroQuantology*, vol. 10, no. 3, pp. 561–565, Sep. 2012.

[73] T. Chen, B. Chen, W. Ma, and L. Sun, "Quaternion least mean kurtosis algorithm for adaptive filtering of 3D and 4D signal processes," in *Proc. 20th Int. Conf. Inf. Fusion (Fusion)*, Jul. 2017, pp. 1–5.

[74] K. Daniilidis and E. Bayro-Corrochano, "The dual quaternion approach to hand-eye calibration," in *Proc. 13th Int. Conf. Pattern Recognit.*, vol. 1, 1996, pp. 318–322.

[75] T. Debaecker, R. Benosman, and S.-H. Ieng, "Cone of view camera model using conformal geometric algebra for classical and panoramic image sensors," in *Proc. 8th Workshop Omnidirectional Vis., Camera Netw. Non-Classical Cameras (OMNIVIS)*, R. Swaminathan, V. Caglioti, and A. Argyros, Eds. Marseille, France, Oct. 2008, pp. 1–13.

[76] J. R. Dooley and J. M. McCarthy, "Spatial rigid body dynamics using dual quaternion components," in *Proc. IEEE Int. Conf. Robot. Automat.*, Sacramento, CA, USA, vol. 1, Apr. 1991, pp. 90–95.

[77] C. Doran, A. Lasenby, and S. Gull, "States and operators in the spacetime algebra," *Found. Phys.*, vol. 23, no. 9, pp. 1239–1264, 1993.

[78] C. Doran, "Geometric algebra and its applications to mathematical physics," Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 1994.

[79] C. Doran, A. Lasenby, S. Gull, S. Somaroo, and A. Challinor, "Spacetime algebra and electron physics," *Adv. Imag. Electron Phys.*, vol. 95, pp. 271–386, May 1996.

[80] L. Dorst, D. Fontjine, and T. Mann. *GAIGEN 2: Generates Fast C + + or JAVA Sources for Low Dimensional Geometric Algebra*. [Online]. Available: http://www.science.uva.nl/ga/gaigen/

[81] L. Dorst, D. Fontjine, and S. Mann, "Geometric algebra for computer science," in *An Object-Oriented Approach to Geometry*. Cambridge, MA, US: Morgan Kaufmann, 2007.

[82] L. Dorst, D. Fontjine, and T. Mann. *GAIGEN 2: Generates Fast C ++ or JAVA Sources for Low Dimensional Geometric Algebra*. Accessed: Feb. 4, 2007. [Online]. Available: http://www.science.uva.nl/ga/gaigen/

[83] L. Dorst, "3D oriented projective geometry through versors of $R^{3,3}$," *Adv. Appl. Clifford Algebras*, vol. 26, pp. 1137–1172, Dec. 2016.

[84] L. Dorst, C. Doran, and J. Lasenby, Eds. *Applications of Geometric Algebra in Computer Science and Engineering*. Basel, Switzerland: Birkhäuser, 2002.

[85] T. A. Ell, "Hypercomplex spectral transformations," Ph.D. dissertation, Univ. Minnesota, Minneapolis, MN, USA, 1992.

[86] G. El Mir, C. Saint-Jean, and M. Berthier, "Conformal geometry for viewpoint change representation," *Adv. Appl. Clifford Algebras*, vol. 24, pp. 443–463, Oct. 2014.

[87] J. Ebling and G. Scheuermann, "Clifford convolution and pattern matching on vector fields," in *Proc. IEEE Vis.*, Los Alamitos, CA, USA, Oct. 2003, pp. 193–200.

[88] J. Ebling and G. Scheuermann, "Clifford Fourier transform on vector fields," *IEEE Trans. Vis. Comput. Graphics*, vol. 11, no. 4, pp. 469–479, Jul. 2005.

[89] S. Franchini, A. Gentile, F. Sorbello, G. Vassallo, and S. Vitabile, "Clifford algebra based edge detector for color images," in *Proc. 6th Int. Conf. Complex, Intell., Softw. Intensive Syst.*, Jul. 2012, pp. 84–91.

[90] S. Franchini, A. Gentile, M. Grimaudo, C. A. Hung, S. Impastato, F. Sorbello, G. Vassallo, and S. Vitabile, "A sliced coprocessor for native clifford algebra operations," in *Proc. 10th Euromicro Conf. Digit. Syst. Design Architectures, Methods Tools (DSD )*, Lübeck, Germany, Aug. 2007, pp. 436–439.

[91] S. Franchini, A. Gentile, F. Sorbello, G. Vassallo, and S. Vitabile, "An embedded, FPGA-based computer graphics coprocessor with native geometric algebra support," *Integration*, vol. 42, no. 3, pp. 346–355, 2009.

[92] S. Franchini, A. Gentile, F. Sorbello, G. Vassallo, and S. Vitabile, "Fixed-size quadruples for a new, hardware-oriented representation of the 4D Clifford algebra," *Adv. Appl. Clifford Algebras*, vol. 21, no. 2, pp. 315–340, 2011.

[93] S. Franchini, A. Gentile, F. Sorbello, G. Vassallo, and S. Vitabile, "Design space exploration of parallel embedded architectures for native clifford algebra operations," *IEEE Design Test Comput.*, vol. 29, no. 3, pp. 60–69, Jun. 2012.

[94] S. Franchini, A. Gentile, F. Sorbello, and S. Vitabile, "Design and implementation of an embedded coprocessor with native support for 5D, quadruple-based clifford algebra," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2366–2381, Dec. 2013.

[95] S. Franchini, A. Gentile, F. Sorbello, G. Vassallo, and S. Vitabile, "ConformalALU: A conformal geometric algebra coprocessor for medical image processing," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 955–970, Apr. 2015.

[96] S. Franchini, A. Gentile, F. Sorbello, G. Vassallo, and S. Vitabile, "A family of embedded coprocessors with native geometric algebra support," in *Proc. 6th Int. Conf. Appl. Geometric Algebra Comput. Sci. Eng.*, 2017, pp. 955–970.

[97] M. Felsberg and G. Sommer, "The monogenic signal," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3136–3144, Dec. 2001.

[98] S. Franchini, A. Gentile, G. Vassallo, F. Sorbello, and S. Vitabile, "A specialized architecture for color image edge detection based on clifford algebra," in *Proc. 7th Int. Conf. Complex, Intell., Softw. Intensive Syst.*, Jul. 2013, pp. 128–135.

[99] M. Garza-Burgos, E. Sanchez-O, and E. Bayro-Corrochano, "Medical robot vision using the conformai geometric algebra framework," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2016, pp. 1087–1093.

[100] A. Gentile, S. Segreto, F. Sorbello, G. Vassallo, S. Vitabile, and V. Vullo, "CliffoSor: An innovative FPGA-based architecture for geometric algebra," in *Proc. 45th Congr. Eur. Regional Sci. Assoc. (ERSA)*, Vrije, Amsterdam, Aug. 2005, pp. 211–217.

[101] P. Girard, "Analytic video (2DCt) signals by Clifford Fourier transforms in multiquaternion Grassmann-Hamilton-Clifford algebras," in *Proc. Electron. Conf. Top. Session Quaternion Clifford FTs Wavelets (ICCA)*, K. Gurlebeck, Ed. Weimar, Germany, Jul. 2011, pp. 1–15.

[102] S. Ghosh-Dastidar and H. Adeli, "Improved spiking neural networks for EEG classification and epilepsy and seizure detection," *Integr. Computer-Aided Eng.*, vol. 14, no. 3, pp. 187–212, May 2007.

[103] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *Int. J. Neural Syst.*, vol. 19, no. 4, pp. 295–308, 2009.

[104] R. Goldman, S. Mann, and X. Jia, "Computing perspective projections in 3-dimensions using rotors in the homogeneous and conformal models of Clifford algebra," *Adv. Appl. Clifford Algebras*, vol. 24, no. 2, pp. 465–491, 2014.

[105] A. Gómez, J. L. Aragón, O. Caballero, and F. Dávila, "The applications of clifford algebras to crystallography using *Mathematica*," in *Clifford Algebras with Numeric and Symbolic Computations*, R. Abłamowicz, J. M. Parra, and P. Lounesto, Eds. Boston, MA, USA: Birkhäuser, 1996, pp. 251–266.
[106] L. E. González-Jim énez, O. Carbajal-Espinosa, A. Loukianov, and E. Bayro-Corrochano, "Robust pose control of robot manipulators using conformal geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 24, pp. 533–552, Feb. 2014.
[107] A. Grafton and J. Lasenby, "Surface fitting using dual quaternion control points with applications in human respiratory modelling," in *Advances in Computer Graphics*. 2020, pp. 421–433.
[108] C. Grubin, "Derivation of the quaternion scheme via the Euler axis and angle," *J. Spacecraft Rockets*, vol. 7, no. 10, pp. 1261–1263, 1970.
[109] C. Gunn, "Geometry, kinematics, and rigid body mechanics in Cayley-Klein geometries," Ph.D. dissertation, Fakultät II Mathematik und Naturwissenschaften der Technischen Universität Berlin, Berlin, Germany, 2011.
[110] C. Gunn, "Geometric algebras for Euclidean geometry," *Adv. Appl. Clifford Algebras*, vol. 27, no. 1, pp. 185–208, Mar. 2017.
[111] C. G. Gunn, "Projective geometric algebra: A new framework for doing Euclidean geometry," 2019, *arXiv:1901.05873*. [Online]. Available: http://arxiv.org/abs/1901.05873
[112] K. Gürlebeck, Ed., in *Proc. Electron. Conf. (ICCA)*, Weimar, Germany, Jul. 2011.
[113] K. Gürlebeck, Ed., *Quaternion and Clifford FTs and Wavelets*. Weimar, Germany: Univ. of Weimar, Jul. 2011.
[114] H. Hadfield, L. Wei, and J. Lasenby, "The forward and inverse kinematics of a delta robot," in *Advances in Computer Graphics*. 2020, pp. 447–458.
[115] H. Hadfield and J. Lasenby, "Constrained dynamics in conformal and projective geometric algebra," in *Advances in Computer Graphics*. 2020, pp. 459–471.
[116] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[117] W. R. Hamilton, *Lectures on Quaternions*, H. Smith, Ed. Dublin, Ireland, 1853.
[118] D. Hestenes, *Space-Time Algebra*. London, U.K.: Gordon and Breach, 1996.
[119] D. Hestenes, "Observables, operators, and complex numbers in the Dirac theory," *J. Math. Phys.*, vol. 16, no. 3, p. 556, 1975.
[120] D. Hestenes, *New Foundations for Classical Mechanics*, Ed. Dordrecht, The Netherlands: D. Reidel, 1986.
[121] D. Hestenes, "Invariant body kinematics: I. Saccadic and compensatory eye movements," *Neural Netw.*, vol. 7, no. 1, pp. 65–77, 1993.
[122] D. Hestenes, "Invariant body kinematics: II. Reaching and neurogeometry," *Neural Netw.*, vol. 7, no. 1, pp. 79–88, 1993.
[123] D. Hestenes, "Hamiltonian mechanics with geometric calculus," in *Spinors, Twistors, Clifford Algebras and Quantum Deformations* (Fundamental Theories of Physics), vol. 52, Z. Oziewicz, B. Jancewicz, and A. Borowiec, Eds. Dordrecht, The Netherlands: Springer, 1993, pp. 203–214.
[124] D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*, Ed. Dordrecht, The Netherlands: D. Reidel, 1984.
[125] D. Hestenes and R. Ziegler, "Projective geometry with Clifford algebra," *Acta Applicandae Math.*, vol. 23, pp. 25–63, Apr. 1991.
[126] D. Hildenbrand, D. Fontijne, C. Perwass, and L. Dorst, "Geometric algebra and its application to computer graphics," *Proc. Tutorial Eurographics, 25th Annu. Conf. Eur. Assoc. Comput. Graph.* Grenoble, Aug. 2004, pp. 1–49.
[127] D. Hildenbrand, J. Zamora, and E. Bayro-Corrochano, "Inverse kinematics computation in computer graphics and robotics using conformal geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 18, nos. 3–4, pp. 699–713, 2008.
[128] D. Hildenbrand, J. Pitt, and A. Koch, "Gaalop—High performance parallel computing based on conformal geometric algebra," in *Geometric Algebra Computing in Engineering and Computer Science*, E. Bayro-Corrochano and G. Scheuermann, Eds. Berlin, Germany: Springer, 2010, pp. 477–494.
[129] D. Hildenbrand, S. Franchini, A. Gentile, and G. S. Vassallo Vitabile, "GAPPCO: An easy to configure geometric algebra coprocessor based on gapp programs," *Adv. Appl. Clifford Algebras*, vol. 27, pp. 2115–2132, Jan. 2017.
[130] A. Hirose, *Complex-Valued Neural Networks* (Studies in Computational Intelligence), vol. 400, 2nd ed. Berlin, Germany: Springer, 2012
[131] E. Hitzer and C. Perwass, "Interactive 3D space group visualization with CLUCalc and the clifford geometric algebra description of space groups," *Adv. Appl. Clifford Algebras*, vol. 20, nos. 3–4, pp. 631–658, Oct. 2010.
[132] E. Hitzer and S. Sangwine, "The orthogonal planes split of quaternions," in *Proc. Electron. Conf. Top. Quaternion Clifford FTs Wavelets (ICCA)*, K. Gürlebeck, Ed. Weimar, Germany, Jul. 2011, pp. 20–35.
[133] E. Hitzer, "OPS-QFTs: A new type of quaternion Fourier transforms based on the orthogonal planes split with one or two general pure quaternions," in *Proc. Numer. Anal. Appl. Math. (ICNAAM)*, vol. 1389, 2011, pp. 280–283.
[134] E. M. S. Hitzer, "Quaternion Fourier transform on quaternion fields and generalizations," *Adv. Appl. Clifford Algebras*, vol. 17, no. 3, pp. 497–517, Aug. 2007.
[135] E. Hitzer, T. Nitta, and Y. Kuroe, "Applications of Clifford's geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 23, no. 2, pp. 377–404, Jun. 2013.
[136] E. A. Hosny, "Introducing geometric algebra to geometric computing software developers: A computational thinking approach," May 2017, *arXiv:1705.06668*. [Online]. Available: https://arxiv.org/abs/1705.06668
[137] R. Hu, Q. Huang, H. Wang, J. He, and S. Chang, "Monitor-based spiking recurrent network for the representation of complex dynamic patterns," *Int. J. Neural Syst.*, vol. 29, no. 8, Oct. 2019, Art. no. 1950006.
[138] J. Hrdina and A. Návrat, "Binocular computer vision based on conformal geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 27, no. 3, pp. 1945–1959, Sep. 2017.
[139] L. Hui, X. Chen, and W. Cao, "The multispectral image edge detection based on Clifford gradient," in *Proc. 7th Int. Conf. Comput. Intell. Secur.*, Dec. 2011, pp. 1238–1242.
[140] B. P. Ickes, "A new method for performing digital control system attitude computations using quaternions," *AIAA J.*, vol. 8, no. 1, pp. 13–21, 1970.
[141] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.
[142] *The Integrating Vision Toolkit*. Accessed: Sep. 11, 2019. [Online]. Available: http://ivt.sourceforge.net/
[143] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.
[144] P. W. Khan, G. Xu, M. A. Latif, K. Abbas, and A. Yasin, "UAV's agricultural image segmentation predicated by clifford geometric algebra," *IEEE Access*, vol. 7, pp. 38442–38450, 2019.
[145] M. Kamarianakis and G. Papagiannakis, "Deform, cut and tear a skinned model using conformal geometric algebra," in *Advances in Computer Graphics*. 2020, pp. 434–446.
[146] S. De Keninck, "Non-parametric realtime rendering of subspace objects in arbitrary geometric algebras," in *Advances in Computer Graphics* (Lecture Notes in Computer Science), vol. 11542, M. Gavrilova, J. Chang, N. Thalmann, E. Hitzer, H. Ishikawa, Eds. Berlin, Germany: Springer, 2019, pp. 549–555.
[147] S. De Keninck and L. Dorst, "Geometric algebra Levenberg-Marquardt," in *Advances in Computer Graphics*. 2020, pp. 511–522.
[148] A. Koschan and M. Abidi, "Detection and classification of edges in color images," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 64–73, Jan. 2005.
[149] H. Kusamichi, T. Isokawa, N. Matsui, and Y. Ogawa, "A new scheme for color night vision by quaternion neural network," in *Proc. 2nd Int. Conf. Auton. Robots Agents*, Palmerston North, New Zeland, Dec. 2004, pp. 101–106.
[150] T. Kusakabe, T. Isokawa, N. Kouda, and N. Matsui, "A study of neural network based on quaternion," in *Proc. 41st SICE Annu. Conf. (SICE)*, Osaka, Japan, vol. 2, Aug. 2002, pp. 736–739.
[151] A. Lasenby, C. Doran, and S. Gull, "2-spinors, twistors and supersymmetry in the spacetime algebras," in *Spinors, Twistors, Clifford Algebras and Quantum Deformations*, I. Z. Oziewicz, Eds. Dordrecht, The Netherlands: Kluwer, 1993, pp. 233–245.
[152] J. Lasenby, W. J. Fitzgerald, A. N. Lasenby, and C. J. L. Doran, "New geometric methods for computer vision: An application to structure and motion estimation," *Int. J. Comput. Vis.*, vol. 26, pp. 191–213 , Feb. 1998.
[153] J. Lasenby and E. Bayro-Corrochano, "Analysis and computation of projective invariants from multiple views in the geometric algebra frameworks," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 13, no. 8, pp. 1105–1121, 1999.

[154] A. Lasenby, J. Lasenby, and R. Wareham, "A convariant approach to geometry using geometric algebra," Univ. Cambridge Department of Engineering, Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR-483, 2005.
[155] J. Lasenby, E. Bayro-Corrochano, and G. Sommer 1996, "A new framework to the formation of invariants and multiple-view," in *Proc. ICIP*, Lausanne, Switzerland, 1996, pp. 313–316.
[156] J. Lasenby, E. Bayro-Corrochano, and G. Sommer 1996, "A new methodology for computing invariants in computer vision," in *Proc. ICPR*, Vienna, Austria, vol. 1, 1996, pp. 393–397.
[157] J. Lasenby and E. Bayro-Corrochano, "Computing 3D projective invariants from points and lines," in *7th Int. Conf. Comput. Anal. Images Patterns*, G. Sommer, K. Daniilidis, and J. Pauli, Eds. Kiel, Germany: Springer-Verlag, Sep. 1997, pp. 82–89.
[158] S. Lazendic, H. De Bie, and A. Pizurica, "Octonion sparse representation for color and multispectral image processing," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Rome, Italy, Sep. 2018, pp. 608–612.
[159] B. U. Lee, "Unit quaternion representation of rotation," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 1991.
[160] H. Li, D. Hestenes, and A. Rockwood, "Generalized homogeneous coordinates for computational geometry," in *Geometric Computing with Clifford Algebra*, G. Sommer, Ed. New York, NY, USA: Springer-Verlag, 2001.
[161] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 4, no. 4, pp. 629–642, 1987.
[162] B. Lian, "Geometric error modeling of parallel manipulators based on conformal geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 28, no. 1, pp. 1–24, Mar. 2018.
[163] G. López-González, N. Aran-Daniel, and E. Bayro-Corrochano, *In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, E. Bayro-Corrochano and E. Hancock, Eds. Puerto Vallarta, Mexico: Springer, Nov. 2014, pp. 722–729.
[164] G. López-Gonzólez, G. Altamirano-Gómez, and E. Bayro-Corrochano, "Geometric entities voting schemes in the conformal geometric algebra framework," *Adv. Appl. Clifford Algebras*, vol. 26, pp. 1945–1959, Aug. 2016.
[165] G. López-González, N. Arana-Daniel, and E. Bayro-Corrochano, "Parallel clifford support vector machines using the Gaussian kernel," *Adv. Appl. Clifford Algebras*, vol. 27, no. 1, pp. 647–660, Mar. 2017.
[166] W. B. Lopes, A. Al-Nuaimi, and C. G. Lopes, "Geometric-algebra lms adaptive filter and its application to rotation estimation," *IEEE Signal Process. Lett.*, vol. 23, no. 6, pp. 858–862, Jun. 2016.
[167] W. B. Lopes and C. G. Lopes, "Geometric-algebra adaptive filters," *IEEE Trans. Signal Process.*, vol. 67, no. 14, pp. 3649–3662, Jul. 2019.
[168] P. Lounesto, *Clifford Algebras and Spinors*. Cambridge, U.K.: Cambridge Univ. Press, 1997.
[169] Y. Maeda, T. Fujiwara, and H. Ito, "Robot control using high dimensional neural networks," in *Proc. SICE Annu. Conf. (SICE)*, Sep. 2014, pp. 738–743.
[170] S. A. Matos, J. R. Canto, C. R. Paiva, and A. M. Barbosa, "A new framework based on geometric algebra for the analysis of materials and metamaterials with electric and magnetic anisotropy," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, San Diego, CA, USA, Jul. 2008, pp. 1–4.
[171] D. P. Mandic, C. Jahanchahi, and C. C. Took, "A quaternion gradient operator and its applications," *IEEE Signal Process. Lett.*, vol. 18, no. 1, pp. 47–50, Jan. 2011.
[172] S. Mann and A. Rockwood, "Computing singularities of 3D vector fields with geometric algebra," in *Proc. IEEE Vis. (VIS)*, Oct./Nov. 2002, pp. 283–290.
[173] J. Medrano-Hermosillo and E. Bayro-Corrochano, "Localized Hamiltonian control of serial robots by screw theory in the geometric algebra framework," *J. Robot.*, 2021.
[174] J. Menneson, C. Saint-Jean, and L. Mascarilla, "Color object recognition based on a Clifford Fourier transform," in *Guide to Geometric Algebra in Practice*, L. Dorst, J. Lasenby, Eds. Berlin, Germany: Springer, 2011, ch. 9 pp. 175–191.
[175] B. Mishra and P. Wilson, "Hardware implementation of a geometric algebra processor core," in *Proc. Int. Conf. Appl. Comput. Algebra (IMACS)*, Nara, Japan, Jul./Aug. 2005. [Online]. Available: http://eprints.ecs.soton.ac.uk/10957/
[176] A. Menti, T. Zacharias, and J. Milias-Argitis, "Geometric algebra: A powerful tool for representing power under nonsinusoidal conditions," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 3, pp. 601–609, Mar. 2007.
[177] F. Montoya, A. Alcayde, F. Arrabal-Campos, and R. Baños, "Quadrature current compensation in non-sinusoidal circuits using geometric algebra and evolutionary algorithms," *Energies*, vol. 12, no. 4, p. 692, Feb. 2019.
[178] F. G. Montoya, A. Alcayde, and F. Arrabal-Campos, "Geometric algebra power theory (GAPOT)," *IEEE Trans. Circuits Syst.*, 2021.
[179] E. U. Moya-Sánchez and E. Bayro-Corrochano, "Symmetry feature extraction based on quaternionic local phase," *Adv. Appl. Clifford Algebras*, vol. 24, pp. 333–334, Feb. 2014.
[180] M. Nobuyuki, T. Isokawa, H. Kusamichi, F. Peper, and H. Nishimura, "Quaternion neural network with geometrical operators," *J. Intell. Fuzzy Syst.*, vol. 15, no. 3, pp. 149–164, 2004.
[181] H. Lev-Ari and A. M. Stankovic, "A geometric algebra approach to decomposition of apparent power in general polyphase networks," in *Proc. 41st North Amer. Power Symp.*, Oct. 2009, pp. 1–6.
[182] Y. Li, W. Liu, X. Li, Q. Huang, and X. Li, "GA-SIFT: A new scale invariant feature transform for multispectral image using geometric algebra," *Inf. Sci.*, vol. 281, pp. 559–572, Oct. 2014.
[183] J. Oviedo-Barriga, O. Carbajal-Espinosa, L. González-Jiménez, B. Castillo-Toledo, and E. Bayro-Corrochano, "Robust tracking of bio-inspired references for a biped robot using geometric algebra and sliding mode control," *Robotica*, vol. 33, no. 1, pp. 209–244, 2015.
[184] J. Oviedo-Barriga, O. Carbajal-Espinosa, L. Gonzalez-Jimenez, B. Castillo-Toledo, and E. Bayro-Corrochano, "Robust tracking of bio-inspired references for a biped robot using geometric algebra and sliding modes," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 5317–5322.
[185] R. Ortega, A. van der Schaft, B. Maschke, and G. Escobar, "Energy-shaping of port-controlled Hamiltonian systems by interconnection," in *Proc. 38th IEEE Conf. Decis. Control*, vol. 2, Dec. 1999, pp. 1646–1651.
[186] J. Ortegón-Aguilar and E. Bayro-Corrochano, "Lie algebra and system identification techniques for 3D rigid motion estimation and monocular tracking," *J. Math. Imag. Vis.*, vol. 25, pp. 173–185, Sep. 2006.
[187] E. Özgür and Y. Mezouar, "Kinematic modeling and control of a robot arm using unit dual quaternions," *Robot. Auton. Syst.*, vol. 77, pp. 66–73, Mar. 2016.
[188] C. R. Paiva and S. A. Matos, "Minkowskian isotropic media and the perfect electromagnetic conductor," *IEEE Trans. Antennas Propag.*, vol. 60, no. 7, pp. 3231–3245, Jul. 2012.
[189] G. Papagiannakis, "Geometric algebra rotors for skinned character animation blending," in *Proc. SIGGRAPH Asia Tech. Briefs (SA)*, 2013, pp. 1–6.
[190] M. Papaefthymiou, D. Hildenbrand, and G. Papagiannakis, "An inclusive conformal geometric algebra GPU animation interpolation and deformation algorithm," *Vis. Comput.*, vol. 32, nos. 6–8, pp. 751–759, Jun. 2016.
[191] M. Papaefthymiou, D. Hildenbrand, and G. Papagiannakis, "A conformal geometric algebra code generator comparison for virtual character simulation in mixed reality," *Adv. Appl. Clifford Algebras*, vol. 27, no. 3, pp. 2051–2066, Sep. 2017.
[192] R. Pappas, "A formulation of Hamiltonian mechanics using geometric calculus," in *Clifford Algebras and Their Applications in Mathematical Physics*, F. Brackx, Ed. Norwell, MA, USA: Kluwer, 1993, pp. 251–258.
[193] T. Parcollet, M. Morchid, and G. Linares, "Quaternion convolutional neural networks for heterogeneous image processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 8514–8518.
[194] T. Parcollet, M. Morchid, and G. Linarès, "A survey of quaternion neural networks," *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2957–2982, Apr. 2020.
[195] M. Pedone, E. Bayro-Corrochano, J. Flusser, and J. Heikkilä, "Quaternion Wiener deconvolution for noise robust color image registration," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1278–1282, Sep. 2015.
[196] C. B. U. Perwass. (2006). *CLUCal*. [Online]. Available: http://www.clucal.info/
[197] C. Perwass, C. Gebken, and G. Sommer, "Implementation of a clifford algebra co-processor design on a field programmable gate array," in *Clifford Algebras: Appl. to Math., Phys., Eng. 6th Int. Conf. Clifford Algebras Appl.* (Progress in Mathematical Physics), R. Ablamowicz, Ed. Boston, MA, USA: Birkhäuser, 2003, pp. 561–575.
[198] A. I. Petroianu, "A geometric algebra reformulation and interpretation of Steinmetz's symbolic method and his power expression in alternating current electrical circuits," *Elect. Eng.*, vol. 97, no. 3, pp. 175–180, 2015.

[199] S. Pollock and S. Mann, "Vortex detection in vector fields using geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 24, no. 2, pp. 423–442, Jun. 2014.
[200] W. D. Price, C. Ton, W. MacKunis, and S. V. Drakunov, "Self-reconfigurable control for dual-quaternion/dual-vector systems," in *Proc. Eur. Control Conf. (ECC)*, Zürich, Switzerland, Jul. 2013, pp. 860–865.
[201] Q. Chen and J. Dai, "Canonical solution of two-point boundary-value problems," *J. Guid., Control, Dyn.*, vol. 34, no. 1, pp. 287–293, Jan. 2011.
[202] R.-R. Jorge and B.-C. Eduardo, "Medical image segmentation, volume representation and registration using spheres in the geometric algebra framework," *Pattern Recognit.*, vol. 40, no. 1, pp. 171–188, 2007.
[203] J. Rivera-Rovelo, H. Silena, and E. Bayro-Corrochano, "Object segmentation using growing neural gas and generalized gradient vector flow in the geometric algebra framework," in *Progress in Pattern Recognition, Image Analysis and Applications (CIARP)*, vol. 4225, J. F. Martínez-Trinidad J. A. Carrasco Ochoa, J. Kittler Eds. Berlin, Germany: Springer, 2006, pp. 306–315.
[204] B. Rosenhahn, C. Perwass, and G. Sommer, "Free-form pose estimation by using twist representations," *Algorithmica*, vol. 38, no. 1, pp. 91–113, Jan. 2004.
[205] S. J. Sangwine, "Fourier transforms of colour images using quaternion or hypercomplex, numbers," *Electron. Lett.*, vol. 32, no. 21, pp. 1979–1980, 1996.
[206] S. J. Sangwine, "Fourier transform of color images using quaternion, or hypercomplex, numbers," *Electron. Lett.*, vol. 32, no. 21, 1979–1980, 1996.
[207] J. Selig, *Introductory Robotics* Hertfordshire, U.K.: Prentis Hall, 1992.
[208] M. Schlemmer, H. Hagen, and I. Holtz, "Clifford pattern matching for color image edge detection," in *Visualization of Large and Unstructured Data Sets* (Lecture Notes in Informatics), vol. 4, no. 1. 2006, pp. 47–58.
[209] G. Sepulveda-Cervantes and E. Portilla-Flores, "Ortogonal approach for haptic rendering algorithm based in conformal geometric algebra," *Appl. Math. Inf. Sci.*, vol. 9, no. 1, pp. 113–124, 2015.
[210] F. Seybold, "Solving motor algebra equation via conjugate gradients," *J. Clifford Anal., Clifford Algebras Appl.*, vol. 1, no. 2, pp. 101–108, 2012.
[211] K. Shoemake, "Animating rotation with quaternion curves," *ACM SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 245–254, Jul. 1985.
[212] M. D. Shuster, "A survey of attitude representations," *J. Astron. Sci.*, 41 (4), pp. 439–517, 1993.
[213] G. Sommer, E. Bayro-Corrochano, and T. Bülow, "Geometric algebra as a framework for the perception—Action cycle," in *Proc. Workshop Theor. Found. Comput. Vis.* Wien, Austria: Springer-Verlag, Mar. 1996, pp. 251–259.
[214] S. S. Somaroo, D. G. Cory, and T. F. Havel, "Expressing the operations of quantum computing in multiparticle geometric algebra," *Phys. Lett. A*, vol. 240, nos. 1–2, pp. 1–7, Mar. 1998.
[215] S. Somaroo, A. Lasenby, and C. Doran, "Geometric algebra and the causal approach to multiparticle quantum mechanics," *J. Math. Phys.*, vol. 40, no. 7, p. 3327, 1999.
[216] G. Sommer and Y. Y. Zeevi, Eds., "Algebraic frames for the perception-action cycle," in *Proc. Workshop AFPAC*, (Lecture Notes in Computer Science), vol. 1888. Berlin, Germany: Springer, 2000.
[217] G. Sommer, Ed., *Geometric computing with Clifford Algebras: Theoretical Foundations and Applications in Computer Vision and Robotics*. Berlin, Germany: Springer, 2001.
[218] G. Soria-García, G. Altamirano-Gómez, S. Ortega-Cisneros, and E. Bayro-Corrochano, "FPGA implementation of a geometric voting scheme for the extraction of geometric entities from images," *Adv. Appl. Clifford Algebras*, vol. 27, no. 1, pp. 685–705, Mar. 2017.
[219] G. Soria-Garcia, G. Altamirano-Gomez, S. Ortega-Cisneros, and E. Bayro-Corrochano, "Conformal geometric algebra voting scheme implemented in reconfigurable devices for geometric entities extraction," *IEEE Trans. Ind. Electron.*, vol. 64, no. 11, pp. 8747–8755, Nov. 2017.
[220] K. W. Spring, "Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: A review," *Mechanism Mach. Theory*, vol. 21, no. 5, pp. 365–373, 1986.
[221] M. J. Stanway and J. C. Kinsey, "Rotation identification in geometric algebra: Theory and application to the navigation of underwater robots in the field," *J. Field Robot.*, vol. 32, no. 5, pp. 632–654, Aug. 2015.
[222] K. Takahashi, Y. Wakaume, T. Hirayama, and M. Hashimoto, "Remarks on hypercomplex-valued neural network-based direct adaptive controller," in *Proc. Int. Conf. Multimedia Syst. Signal Process. (ICMSSP)*, Taipei, Taiwan, Sep. 2016, pp. 85–89.
[223] O. Tanrikulu and A. G. Constantinides, "Least-mean kurtosis: A novel higher-order statistics based adaptive filtering algorithm," *Electron. Lett*, vol. 30, no. 3, pp. 189–190, Feb. 1994.
[224] S. Thiruvengadam, J. S. Tan, and K. Miller, "Artificial intelligence using hyper-algebraic networks," *Neurocomputing*, vol. 399, pp. 414–448, Jul. 2020.
[225] S. Thiruvengadam and K. Miller, "A geometric algebra based higher dimensional approximation method for statics and kinematics of robotic manipulators," *Adv. Appl. Clifford Algebras*, vol. 30, no. 1, pp. 1–23, Feb. 2020.
[226] C. C. Took and D. P. Mandic, "The quaternion LMS algorithm for adaptive filtering of hypercomplex processes," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1316–1327, Apr. 2009.
[227] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.
[228] E. Vazquez-Santacruz and E. Bayro-Corrochano, "An improved geometric radial basis function network for hand-eye calibration," in *Proc. Int. Joint Conf. Neural Netw.*, San Jose, CA, USA, Jul. 2011, pp. 1308–1315.
[229] Z. Wanxie and W. T. Zhigan, and S. Jun, *Theory and Computation of State Space Control*. Beijing, China: Science Publishing House, 2006
[230] R. Wang, X. Zhang, and W. Cao, "Clifford fuzzy support vector machines for classification," *Adv. Appl. Clifford Algebras*, vol. 26, no. 2, pp. 825–846, Jun. 2016.
[231] R. Wang, Z. Cao, X. Wang, W. Xue, and W. Cao, "GA-STIP: Action recognition in multi-channel videos with geometric algebra based spatio-temporal interest points," *IEEE Access*, vol. 6, pp. 56575–56586, 2018.
[232] R. Wang, Y. Shi, and W. Cao, "GA-SURF: A new speeded-up robust feature extraction algorithm for multispectral images based on geometric algebra," *Pattern Recognit. Lett.*, vol. 127, pp. 11–17, Nov. 2019.
[233] R. Wang, K. Wang, W. Cao, and X. Wang, "Geometric algebra in signal and image processing: A survey," *IEEE Access*, vol. 7, pp. 156315–156325, 2019.
[234] R. Wang, Y. He, C. Huang, X. Wang, and W. Cao, "A novel least-mean kurtosis adaptive filtering algorithm based on geometric algebra," *IEEE Access*, vol. 7, pp. 78298–78310, 2019.
[235] R. Wang, W. Zhang, Y. Shi, X. Wang, and W. Cao, "GA-ORB: A new efficient feature extraction algorithm for multispectral images based on geometric algebra," *IEEE Access*, vol. 7, pp. 71235–71244, 2019.
[236] R. Wareham, J. Cameron, and L. Lasenby, "Applications of conformal geometric algebra in computer vision and graphics," in *Proc. 6th Int. Conf. Comput. Algebra Geometric Algebra Appl.*, May 2004, pp. 329–349.
[237] B. Xian, M. S. De Queiroz, D. Dawson, and I. Walker, "Task-space tracking control of robot manipulators via quaternion feedback," *IEEE Trans. Robot. Autom.*, vol. 20, no. 1, pp. 160–167, Feb. 2004.
[238] W. Xie, J. Li, and J. Pei, "An analysis of THz-TDS signals using geometric algebra," *Proc. SPIE*, vol. 7277, Feb. 2009, Art. no. 72770E.
[239] L. Xiaodong, L. Aijun, Y. Changjun, and S. Fulin, "Widely linear quaternion unscented Kalman filter for quaternion-valued feedforward neural network," *IEEE Signal Process. Lett.*, vol. 24, no. 9, pp. 1418–1422, Sep. 2017.
[240] D. Xu, Y. Xia, and D. P. Mandic, "Optimization in quaternion dynamic systems: Gradient, hessian, and learning algorithms," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 249–261, Feb. 2016.
[241] C. Xu, H. Liu, W. Cao, and J. Feng, "Multispectral image edge detection via Clifford gradient," *Sci. China Inf. Sci.*, vol. 55, no. 2, pp. 260–269, 2012.
[242] V. Záda, K. Belda, "Robot control in terms of Hamiltonian mechanics," in *Proc. 22nd Int. Conf. Eng. Mech.*, Svratka, Czech Republic, May 2016, pp. 627–630.
[243] H. Yao, Q. Chen, X. Chai, and Q. Li, "Singularity analysis of 3-RPR parallel manipulators using geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 27, no. 3, pp. 2097–2113, Sep. 2017.
[244] Z. Yu, L. Yuan, and W. Luo, "Clifford algebra and GIS spatial analysis algorithms—The case study of geographical network and Voronoi analysis," in *Proc. 2nd Int. Workshop Comput. Graph., Comput. Vis. Math.*, 2010, pp. 155–158.

[245] L. Yuan, Z. Yu, W. Luo, L. Yi, and G. Lü, "Geometric algebra for multidimension-unified geographical information system," *Adv. Appl. Clifford Algebras*, vol. 23, no. 2, pp. 497–518, 2013.

[246] D. Hestenes, *Space-Time Algebra*. London, U.K.: Gordon and Breach, 1966.

[247] D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Dordrecht, The Netherlands: D. Reidel, 1984.

[248] D. Hestenes, *New Foundations for Classical Mechanics*. Dordrecht, The Netherlands: D. Reidel, 1986, Books on Mathematics and Physics Useful for Engineering and Computer Science.

[249] C. Doran and A. Lasenby, *Geometric Algebra for Physicists*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[250] H. Li, *Invariant Algebras and Geometric Reasoning*. Singapore: World Scientific, 2018.

[251] C. Perwass, *Geometric Algebra With Applications in Engineering*. Berlin, Germany: Springer-Verlag, 2009.

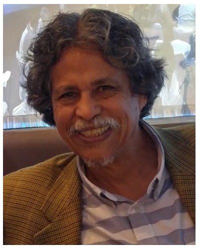
[252] D. Hildenbrand, *Foundations of Geometric Algebra Computing*. Berlin, Germany: Springer-Verlag, 2013.

[253] E. Hitzer, S. J. Sangwine, and J. Stephen, Eds., *Quaternion and Clifford Fourier Transforms and Wavelets*. 2013, Computer Science and Engineering.

[254] E. Bayro-Corrochano, *Geometric Algebra Applications: Computer Vision, Graphics and Neurocomputing*, vol. 1. Heidelberg, Germany: Springer-Verlag, 2019.

[255] E. Bayro-Corrochano, *Geometric Algebra Applications: Robot Modeling and Control*, vol. 2. Cham, Switzerland: Springer, 2020.

[256] E. Bayro-Corrochano, *Geometric Algebra Applications: Integral Transforms for Science and Engineering*, vol. 3. Cham, Switzerland: Springer, to be published.

**EDUARDO BAYRO-CORROCHANO** (Senior Member, IEEE) received the Ph.D. degree in systems engineering from the University of Wales, Cardiff, U.K., in 1994. From 1995 to 1999, he was a Researcher and a Lecturer with the Institute for Computer Science, Christian–Albrechts University, Kiel, Germany, working on applications of geometric Clifford algebra to cognitive systems. From 2007 to 2008, he was the DFG Merkator Gast Professor with the KIT, Technische Hochschule, Karlsruhe, Germany, and from 2014 to 2015, he was a Visiting Full Professor with the Media Laboratory, Massachusetts Institute of Technology (MIT). He is currently a Full Professor in geometric cybernetics with the Department of Electrical Engineering and Computer Science, CINVESTAV Guadalajara, Guadalajara Campus, Zapopan, Jalisco, Mexico. He is the author of seven Springer-Verlag books. He has published over 230 refereed journal articles, book chapters, and conference papers. He is a fellow of the International Association of Pattern Recognition Society and the Asian Artificial Intelligence Association. He was an Associate Editor of the IEEE Transaction on Neural Networks and Learning Systems, *Journal of Mathematical Imaging and Vision*, and *Journal on Pattern Recognition*. He is an Associate Editor of the *Robotica* journal and an Editor of ICRA Conference. He was the General Chair of ICPR 2016 and IEEE/RAS Humanoids 2016, Cancun, Mexico.