VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MATEMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

GEOMETRICKÉ ALGEBRY A JEJICH APLIKACE

GEOMETRIC ALGEBRAS AND THEIR APPLICATIONS

HABILITAČNÍ PRÁCE
HABILITATION THESIS

AUTOR ALEŠ NÁVRAT
AUTHOR

BRNO 2023

Contents

1 Introduction and motivation

In the most general meaning, by a geometric algebra we mean an algebraic representation of a geometric concept. The basic examples are the Grassmann algebra and the Hamilton's algebra of quaternions. The former algebra can be used to describe vector subspaces by so called Plücker embedding while the latter can be used to represent rotations in the 3D space. The two algebras were unified later by W. H. Clifford into one geometric algebra, and this is what we usually mean by a geometric algebra nowadays. Hence, in the algebraic sense, a geometric algebra is the same as a Clifford algebra - a well established and well studied object in mathematics. The adjective 'geometric' means that we take into account its Grassmannian structure and we specify its relation to the geometry. Although one usually means the Euclidean geometry in the engineering literature, it turns out that it is convenient to see the geometry in a more abstract sense here. Namely, we will use the concept of Klein geometry, where the geometry of the space is given by its symmetries. By viewing these symmetries as orthogonal transformations of a suitable vector space, we can represent them by invertible elements in the corresponding Clifford algebra.

For this point of view it is crucial to understand the representations of orthogonal transformations in Clifford algebras. The best way to demonstrate the principles of such representations is shown in the well known examples from the lowest dimensions. Concretely, these are the fields of complex numbers $\mathbb{C}$ and quaternions $\mathbb{H}$, which both can be considered as subalgebras of a Clifford algebra. The complex numbers parameterize 2D rotations while the quaternions parameterize 3D rotations. Let us recall that rotations in the plane are given by multiplications by a unit complex number if we identify $\mathbb{R}^2$ with the complex plane $\mathbb{C}$. Namely, given $(v^1, v^2) \in \mathbb{R}^2$ which we see as a complex number $v^1 + \mathbf{i}v^2$, the coordinates of a vector rotated by an angle φ are given by real and imaginary parts of the complex number

$$(\cos\varphi + \mathbf{i}\sin\varphi)(v^1 + v^2\mathbf{i}). \tag{1.1}$$

Moreover, the imaginary part of a product of two complex numbers encodes the scalar product of the corresponding vectors in the plane while the real part is the area spanned by these two vectors. The spatial rotations in 3D space can be represented by unit quaternions in a similar way however it turns out that the left multiplication from (1.1) must be replaced by another operation called the conjugation or the sandwich product. Concretely, we first identify each vector $v = (v^1, v^2, v^3) \in \mathbb{R}^3$ with a pure imaginary quaternion

$$\mathsf{v} = v^1\mathbf{i} + v^2\mathbf{j} + v^3\mathbf{k}, \tag{1.2}$$

where $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$ are three complex units which satisfy relations $\mathbf{k} = \mathbf{ij} = -\mathbf{ji}$. A rotation in 3D space by an arbitrary angle φ along an axis determined by the unit vector n, which is represented by the quaternion $\mathbf{n}$ in the sense of (1.2), is then given by formula

$$\mathsf{v} \mapsto \mathsf{R}\mathsf{v}\mathsf{R}^{-1}, \tag{1.3}$$

where R is a unit quaternion given by

$$\mathsf{R} = \cos\frac{\varphi}{2} + \mathbf{n}\sin\frac{\varphi}{2}. \tag{1.4}$$

The multiplicative inverse of a nonzero quaternion is given by the conjugate-to-norm ratio which for the unit quaternion has a simple form $\mathsf{R}^{-1} = \cos(\varphi/2) - \mathbf{n}\sin(\varphi/2)$. The complex units actually represent rotations by the right angles around the coordinate axes and the relations express their non-commutativity. Indeed, for example the rotation of v by 90° around the x-axis is

$$\mathbf{i}\mathsf{v}\mathbf{i} = -v^1\mathbf{i} + v^2\mathbf{j} + v^3\mathbf{k}.$$

Let us note that the half-angle in the formula (1.4) can be explained by the fact that the unit quaternion R appears twice in (1.3). Consequently, the same rotation is represented by $\pm\mathsf{R}$ which reflects the fact that the representation of orthogonal transformations in the Clifford algebra has a spinorial nature. The form of rotations in the complex plane given by (1.1) is degenerate due to commutativity of complex numbers (and rotations in the plane).

The benefits of representing rotations by quaternions are well known and this approach is used in applications a lot. The implementation of algorithms based on quaternions is easier, the algorithms are less time consuming compared to the usual representation by matrices and Euler angles. Another significant benefit is the elimination of the effect known as the gimbal lock. What is less known and used is that we can also easily express scalar and vector products and thus the projections and reflections of vectors via quaternions. Indeed, the real part of the quaternionic product uv of two pure imaginary quaternions gives the scalar product (up to a sign) of the vectors represented by quaternions u, v while the imaginary part of uv encodes the vector product of these vectors. The reflection of a vector v in a plane with the normal unit vector u is given by a formula similar to the one for rotations, namely

$$\mathsf{v} \mapsto \mathsf{uvu}. \tag{1.5}$$

By Cartan-Dieudonné theorem, each rotation can be expressed as a composition of two consecutive reflections in the planes. If $\mathsf{u}_1, \mathsf{u}_2$ denote the corresponding unit normal vectors, then by (1.5) the rotation is given by $\mathsf{v} \mapsto \mathsf{u}_2\mathsf{u}_1\mathsf{v}\mathsf{u}_1\mathsf{u}_2$. Combining this result with (1.3), we get another expression for a quaternion that realizes a rotation, namely $\mathsf{R} = \mathsf{u}_2\mathsf{u}_1$. Obviously, the formula for reflection can be also used to derive a formula for the orthogonal projection, namely $\mathsf{v} \mapsto 1/2(\mathsf{v} + \mathsf{uvu})$ gives the projection of a vector v to the plane with normal vector u in terms of quaternionic multiplication.

The theory of geometric algebras is a framework that generalizes and unifies this concept of using complex numbers and quaternions for representing rotations etc. A necessary algebraic and geometric background for the formulation of such theory is given in the preliminary Chapter 2. Here we refer mainly to [1] and also to [33, 12, 34]. For a more detailed description of Klein geometries and their 'curved' analogues see [2, 3]. A panorama of basic examples of geometric algebras is introduced in Chapter

3. In particular, sections 3.1-3.3 describe geometric algebras suitable for representing Euclidean geometry in a general dimension, which is the most appealing towards applications. A geometric algebra for conics is described in section 3.4 and the last section 3.5 is devoted to the complex geometric algebras and their applications in quantum computing.

2 Preliminaries from algebra and geometry

For good understanding of geometric algebras it is necessary to know certain concepts and constructions from algebra and geometry. This chapter aims to summarize the related mathematical background. A basic knowledge of mathematics of the reader is assumed.

2.1 Grassmann algebra

Let us recall that Grassmann algebra on a vector space V is an algebra defined by the *wedge product* $\wedge$ (also called the outer product) which is linear with respect to multiplication by scalars, distributive with respect to vector addition, associative and for each vector $v \in V$ satisfies

$$v \wedge v = 0. \tag{2.1}$$

Substituting $u+v$ for v in this equation we get $u \wedge v + v \wedge u = 0$ which shows that the wedge product is anti-commutative. The property (2.1) can be easily generalized to an outer product of k vectors by associativity. Namely, $v_1 \wedge \cdots \wedge v_k = 0$ holds if and only if the vectors $v_1, \dots, v_k$ are linearly dependent. For linearly independent vectors the elements $v_1 \wedge \cdots \wedge v_k$ are called *k-blades*. Their linear combinations form the vector space $\Lambda^k V$ of Grassmann elements of *grade k*, so called *k-vectors*. Identifying 0-blades with the scalars the linear combinations of k-blades for $k \in \mathbb{Z}$ form an associative algebra with unity which is called the *Grassmann algebra* (also called the outer algebra). It is denoted by ΛV and its general elements are called *multivectors* and will be denoted by bold letters.

For practical purposes it is convenient to describe the Grassmann algebra via basis. If we fix a basis $(e_1, \dots, e_n)$ of V, then a basis blade of grade r is $e_A = e_{i_1} \wedge \cdots \wedge e_{i_k}$, where the multi-index A is a set of indices ordered in the natural way $1 \leq i_1 \leq \cdots \leq i_k \leq n$, and we put $e_\emptyset = 1$. For the outer product we have

$$e_j \wedge e_A = \begin{cases} e_j \wedge e_{i_1} \wedge \cdots \wedge e_{i_k} & \text{if } j \notin A \\ 0 & \text{if } j \in A \end{cases}. \tag{2.2}$$

Blades of grades $0 \leq k \leq n$ form the basis of the graded Grassmann algebra $\Lambda \mathbb{R}^n$. Hence the dimension of $\Lambda^k \mathbb{R}^n$ is equal to $\binom{n}{k}$ and the dimension of the whole Grassmann algebra is 2^n. The element of the highest grade, i.e. the element of the one-dimensional vector space $\Lambda^n V$, is called the *pseudoscalar*.

Geometrically, the Grassmann algebra can be viewed as an algebra of subspaces as follows. For a basis $w_1, \dots, w_k$ of a vector subspace $W \subset V$ we form the outer product $\mathsf{w} = w_1 \wedge \cdots \wedge w_k \subset \Lambda^k V$. If we choose a different basis, then we get the same k-blade up to a scalar multiple which is equal to the determinant of the matrix of the corresponding basis change. Thus the projective class $[\mathsf{w}] = [w_1 \wedge \cdots \wedge w_k]$ is independent of the choice of basis and the prescription $W \mapsto [\mathsf{w}]$ defines an injective map $\iota : W \to \mathbb{P}(\Lambda^k V)$, called the *Plücker embedding*. Hence the subspace W can be represented uniquely by the projective class of k-blades $\mathsf{w} \in \Lambda^k V$ in this sense.

For further constructions and applications it is useful to consider the dual vector space V^*, i.e. the vector space of linear forms on V, and its induced dual Grassmann algebra. With the help of V^* we can define a product on Grassmann algebra ΛV that is "dual" to the wedge product as follows. For $\alpha \in V^*$ the *left contraction* is a linear mapping defined on basis blades by

$$\alpha \cdot e_A = \alpha \cdot (e_{i_1} \wedge \cdots \wedge e_{i_k}) = \sum_{k=1}^{r} (-1)^k \alpha(e_{i_k}) e_{A \setminus \{i_k\}}, \tag{2.3}$$

where $e_{A \setminus \{i_k\}}$ is the blade of grade $k-1$ created by deleting e_{i_k} from e_A. The left contraction by a general k-form, i.e. an element of the dual Grassmann algebra $\boldsymbol{\alpha} \in \Lambda(V^*) \cong (\Lambda V)^*$, is then given by (2.3) together with the recursive formula $(\alpha \wedge \beta) \cdot e_A = \alpha \cdot (\beta \cdot e_A)$ for each $\alpha, \beta \in V^*$. One can define also the right contraction in the same way but by the recursive formula in the reversed order. Note that the contractions lower the grades of k-vectors in contrast to the wedge product. In particular, the left contraction of a chosen pseudoscalar $\mathsf{I} \in \Lambda^n V$ defines the duality

$$\Lambda^k V^* \to \Lambda^{n-k} V, \ \boldsymbol{\alpha} \mapsto \boldsymbol{\alpha}^* = \boldsymbol{\alpha} \cdot \mathsf{I} \tag{2.4}$$

between Grassmann algebras generated by vector space V and its dual vector space V^*. The compatibility of the wedge product and the left contraction with duality (2.4) can be expressed by the following formula for multivectors $\mathsf{u}, \mathsf{v} \in \Lambda V$ which says that $\mathsf{u}\cdot$ is left adjoint to $\mathsf{u}\wedge$

$$(\mathsf{u} \wedge \mathsf{v})^* = \mathsf{u} \cdot \mathsf{v}^*. \tag{2.5}$$

Note that on the right-hand side we consider the left contraction of a form v^* by a multivector u instead of the contraction of a multivector by a form. This is possible since we can switch the role of V and V^* in the above considerations. In that case we obtain a duality $\Lambda^k V \to \Lambda^{n-k} V^*$ defined by $\mathsf{v}^* = \mathsf{v} \cdot \mathsf{I}^*$, where I^* is a pseudoscalar in the dual Grassmann algebra ΛV^*. It is easy to show by definitions that the composition of the two duality mappings v^{**} is equal to v up to a nonzero scalar multiple which depends on the choice of pseudoscalars I a I^*.

2.2 Quadratic space

Quadratic space is a finite dimensional vector space V equipped with a quadratic form, or equivalently a symmetric bilinear form. Recall that *symmetric bilinear form* on V is a map $B : V \times V \to \mathbb{R}$ which is linear in both arguments and which is symmetric, i.e.

$B(u,v) = B(v,u)$ for all $u,v \in V$. In the case of a complex bilinear form the values are in the field of scalars $\mathbb{C}$ instead of $\mathbb{R}$. Symmetric bilinear form B is determined uniquely by quadratic form $Q_B = B(v,v)$ by the so called polarization identity

$$B(u,v) = \frac{1}{2}\left(Q_B(u+v) - Q_B(u) - Q_B(v)\right), \tag{2.6}$$

so we may equivalently alter when speaking about quadratic and bilinear form. The subspace formed by vectors $v \in V$ for which $B(v,w) = 0$ holds for all $w \in V$ is called the *kernel* of B. The bilinear form B is called *nondegenerate* if its kernel is trivial. In this case, and only in this case, the bilinear form defines the isomorphism

$$V \to V^*, \; v \mapsto B(v,\cdot), \tag{2.7}$$

between vector space V and its dual V^*. According to the law of inertia each symmetric bilinear form is diagonalizable. Concretely, in each real vector space of dimension n endowed with a non-degenerate symmetric bilinear form B of signature (p,q) there exists an associated orthonormal basis $(e_1,\ldots,e_n)$ defined by

$$B(e_i,e_j) = \begin{cases} 1 & \text{if } i = j = 1,\ldots,p \\ -1 & \text{if } i = j = p+1,\ldots,n \\ 0 & \text{if } i \neq j \end{cases} \quad \text{where } 1 \leq i,j \leq n = p+q.$$

The couple (p,q) is called *signature* of the bilinear form B. The quadratic space defined by a bilinear form of signature (p,q) will be denoted $\mathbb{R}^{p,q}$. If one considers also degenerate bilinear forms, then an orthonormal basis is extended by a basis of the kernel of dimension r and by signature we mean the triple (p,q,r). Clearly, in the complex case no signature is involved, since each basis vector e_j may be multiplied by the imaginary unit i to change the sign of its square.

Recall further that a vector $v \in V$ is called *isotropic* if $B(v,v) = 0$. *Anisotropic subspace* is the maximal subspace of V which does not contain any isotropic vector. On the contrary, the subspace containing only isotropic vectors is called the *isotropic subspace*. The dimension of the maximal isotropic subspace of quadratic space $\mathbb{R}^{p,q}$ is equal to $d = \min(p,q)$. Moreover, there exist a distinguished basis of this space, so called *Witt basis* formed by vectors $e_1,\ldots,e_d,f_1,\ldots,f_d$ such that $B(e_i,f_j) = \delta_{ij}$ and $B(e_i,e_j) = B(f_i,f_j) = 0$. If we extend this basis of the maximal isotropic subspace by the orthonormal basis of the complementary anisotropic subspace we get a basis of the whole $\mathbb{R}^{p,q}$ such that in a suitable ordering the bilinear form B is given by matrix

$$B = \begin{pmatrix} 0 & 0 & 1_d \\ 0 & 1_{|p-q|} & 0 \\ 1_d & 0 & 0 \end{pmatrix}, \tag{2.8}$$

where 1_d stands for the unit $d \times d$ matrix.

2.3 Clifford algebra

Let us consider a real quadratic space (V, B). In contrast to Grassmann algebra defined by vector space V only, we take the bilinear form B into account from beginning and we construct a new product on ΛV, so called *geometric product*, also called the Clifford product, denoted simply by concatenation of multivectors. Similarly to the outer product we demand the distributive law and associativity but for vectors $v \in V$ we demand

$$v^2 = B(v, v) \tag{2.9}$$

instead of (2.1). In contrast to the outer product, the geometric product is not anti-commutative in general. Indeed, substituting $u + v$ for v in (2.9) gives $uv + vu = 2B(u, v)$. *Clifford algebra* is the maximally free algebra together with the geometric product, i.e. the set of linear combinations of geometric products of k vectors, where $k \in \mathbb{N}$ is arbitrary (but $k \leq n$ is sufficient as we shell see).

Since this definition is rather abstract for computations in applications we write also the definition via basis. Namely, let us fix an orthonormal basis $(e_1, \ldots, e_n)$ associated to a nondegenerate bilinear form B of signature (p, q). The geometric product on vectors can be written in terms of the wedge product and inner product $e_i \cdot e_j = B(e_i, e_j)$ as follows

$$e_i e_j = e_i \cdot e_j + e_i \wedge e_j, \quad 1 \leq i, j \leq m,$$

For the geometric product between blades of a general grade we obtain a similar formula. Namely, the wedge product is extended according to (2.2) while the inner product extends to the left contraction which can be viewed via isomorphism $V^* \cong V$ given by (2.7) as a new product on Grassmann algebra. Namely, instead of (2.3) the formula for the left contraction by a vector is given by

$$e_j \cdot e_A = e_j \cdot (e_{i_1} \wedge \cdots \wedge e_{i_r}) = \sum_{k=1}^{r} (-1)^k B(j, i_k) e_{A \setminus \{i_k\}}.$$

and for the geometric product we get

$$e_j e_A = e_j \cdot e_A + e_j \wedge e_A. \tag{2.10}$$

Finally, these definitions are linearly extended to the whole of the vector space $\Lambda \mathbb{R}^n$. Thus we get the real Clifford algebra $\mathbb{G}_{p,q} = \mathrm{Cl}(\mathbb{R}^{p,q})$. Note that this algebra is naturally graded; the grade zero and grade one elements are identified with $\mathbb{R}$ and $\mathbb{R}^n$ respectively. Note also that the definition of the Clifford algebra by formula (2.10) allows the bilinear form to be degenerate. The Clifford algebra generated by a bilinear form with a kernel of dimension r and whose restriction to the nondegenerate part has signature (p, q) will be denoted by $\mathbb{G}_{p,q,r}$.

The $\mathbb{Z}$-grading by grades define a $\mathbb{Z}_2$-grading of the Clifford algebra according to the parity of grades. Namely, the linear map $v \to -v$ on $\mathbb{R}^n$ extends to an automorphism α called the grade involution and decomposes $\mathbb{G}_{p,q}$ into positive and negative eigenspaces.

The former is called the even subalgebra $\mathbb{G}^0_{p,q}$ and the latter is called the odd part $\mathbb{G}^1_{p,q}$. In addition to α, there are two important antiautomorphisms of real Clifford algebras. The first one is $\tilde{x}$ called the reverse or transpose operation and it is defined by extension of identity on $\mathbb{R}^n$ and by the anti automorphism property $\widetilde{xy} = \tilde{y}\tilde{x}$. The second anti automorphism is called the Clifford conjugation $\bar{x}$ and the operation is defined by composing α and the reverse

$$\bar{x} = \alpha(\tilde{x}) = \widetilde{\alpha(x)}. \tag{2.11}$$

By a recursive application of equation (2.10), the geometric product of k vectors can be expressed as a linear combination of blades of grades $\leq k$, more precisely there appear blades of grades $k, k-2, \ldots$. So we get a map from Clifford algebra to Grassmann algebra which is an isomorphism of filtered vector spaces in fact, [1]. Its inverse is called the *quantum map* and it allows to see the Grassmann structure inside the Clifford algebra. In particular, the outer product and the left contraction can be seen as the highest respectively the lowest grade part of the geometric product, i.e. for blades $\mathsf{u}, \mathsf{v} \in \Lambda\mathbb{R}^n$ of grades k, ℓ we have

$$\mathsf{u} \wedge \mathsf{v} = [\mathsf{uv}]_{k+\ell}, \ \mathsf{u} \,\lrcorner\, \mathsf{v} = [\mathsf{uv}]_{\ell-k}, \tag{2.12}$$

where $[\ ]_k$ denotes the projection operator $\mathbb{G}_{p,q} \to \Lambda^k\mathbb{R}^n$. The existence of the quantum map also shows that the Clifford algebra has dimension 2^n and that the basis blades of the Grassmann algebra form a basis of the Clifford algebra.

2.4 Orthogonal Lie group and algebra

Let us recall that orthogonal transformations of a quadratic space (V, B) are those invertible transformations that keep the bilinear symmetric form B invariant. These transformations form the so called *orthogonal group*

$$\mathrm{O}(V) = \{A \in \mathrm{GL}(V) |\ B(Au, Av) = B(u, v) \text{ for all } u, v \in V\}. \tag{2.13}$$

The group structure is revealed by the well-known Cartan-Dieudonné theorem which says that each element of $\mathrm{O}(V)$ is given by a composition of simple reflections, i.e. a composition of maps

$$R_v(u) = u - 2\frac{B(u, v)}{B(v, v)}v \tag{2.14}$$

where $u, v \in V$ and v may be assumed of the unit length, i.e. $B(v, v) = 1$, since this this formula is invariant with respect to rescaling. The group of orthogonal transformations is a Lie group, i.e. it is a smooth manifold at the same time and the group multiplication is a smooth map. Hence the local structure of $\mathrm{O}(V)$ is determined by the structure of the corresponding Lie algebra, i.e. the structure of the tangent space of $\mathrm{O}(V)$ in the identity. By derivation of (2.13) we get the *orthogonal Lie algebra*

$$\mathfrak{o}(V) = \{A \in \mathrm{End}(V) |\ B(Au, v) + B(u, Av) = 0 \text{ for all } u, v \in V\}, \tag{2.15}$$

where the Lie bracket is given by the commutator of linear maps. Recall that the Lie algebra is locally diffeomorphic to the Lie group and this diffeomorphism is given by the *exponential map*. This map can be defined in various ways, for example by a curve in $g(t) \in \mathrm{O}(V)$ which is the unique solution of the Cauchy problem

$$\dot{g} = A \circ g, \;\; g(0) = \mathrm{id}, \tag{2.16}$$

Then the exponential map $\exp : \mathfrak{o}(V) \to \mathrm{O}(V)$ is defined as the solution in time $t = 1$, i.e. $\exp(A) = g(1)$. If we take into account also the time slot, then we write $g(t) = \exp(At)$. By the consecutive integration of equation (2.16) we get the well-known series for the exponential map

$$\exp(At) = \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k. \tag{2.17}$$

Let us also recall that the orthogonal group of transformations of the quadratic space $\mathbb{R}^{p,q}$ is denoted by $\mathrm{O}(p,q)$ and its subgroup given by matrices with the unit determinant, i.e. the linear maps preserving the orientation, is denoted by $\mathrm{SO}(p,q)$. The corresponding Lie algebra for both these groups is the same and is denoted by $\mathfrak{so}(p,q)$.

Now we show how the orthogonal group and algebra are represented in the Clifford algebra setting. For the simple reflection of a vector u in vector v given by (2.14) we get an nice formula in the Clifford algebra. Namely, the basic equation (2.9) implies $B(u,v) = \frac{1}{2}(uv + vu)$ and $v^{-1} = v/B(v,v)$ and so we get

$$R_v(u) = u - \frac{uv^2 + vuv}{v^2} = -\frac{vuv}{v^2} = -vuv^{-1}. \tag{2.18}$$

since the geometric product between a vector and a scalar coincides with the usual scalar multiplication. The crucial observation now is that the composition of such two reflections is given by geometric product of corresponding vectors, namely

$$R_{v_1} \circ R_{v_2} = (v_1 v_2) u (v_1 v_2)^{-1}.$$

Hence by Cartan-Dieudonné theorem each orthogonal map is represented in Clifford algebra in this sense by a geometric product of vectors of unit norm. Concretely, each element of $\mathrm{O}(V)$ is represented as

$$u \mapsto (-1)^k \mathsf{g} u \mathsf{g}^{-1}, \tag{2.19}$$

where $\mathsf{g} \in \mathrm{Cl}(V)$ is an invertible element, called *versor*, which is crated by the geometric product of k unit vectors and satisfies unitary condition

$$\mathsf{g}\widetilde{\mathsf{g}} = 1, \tag{2.20}$$

where $\widetilde{\mathsf{g}}$ is the reverse of element g, see the last paragraph of 2.3. Note that this representation is not unique since the versors $\pm\mathsf{g}$ represent the same orthogonal map. In fact, the versors (2.20) form the group $\mathrm{Pin}(V)$ which is a double cover of $\mathrm{O}(V)$, for more details see [].

This representation of group $\mathrm{O}(V)$ defines a representation of the corresponding Lie algebra $\mathfrak{so}(V)$ in Clifford algebra. Namely, by differentiation of (2.19) for a curve $\mathsf{g}(t)$ of versors representing a curve $g(t) \in SO(V)$ and by linearity of the geometric product we get that in the Clifford algebra an anti-symmetric map $A = \dot{g}(0) \in \mathfrak{so}(V)$ is represented by

$$v \mapsto \mathsf{A}v - v\mathsf{A}, \tag{2.21}$$

where $\mathsf{A} = \dot{\mathsf{g}}(0) \in \mathrm{Cl}(V)$. It is easy to show by basic properties of the geometric product that it always has grade two, i.e. it is a so called *bivector* $\mathsf{A} \in \Lambda^2 V$, see [1]. Indeed, bivectors have the exceptional property that they are closed under the commutator with respect to the geometric product. Hence they form a Lie algebra $\Lambda^2 V$ and the formula (2.21) defines an surjective homomorphism of Lie algebras $\Lambda^2 V \to \mathfrak{so}(V)$. In the case of a nondegenerate form B, the dimensions are the same and this homomorphism is an isomorphism of Lie algebras

$$\Lambda^2 V \cong \mathfrak{so}(V). \tag{2.22}$$

Each bivector A defines a Cauchy problem in the Clifford algebra $\dot{\mathsf{g}} = \mathsf{A}\mathsf{g}$ with $\mathsf{g}(0) = 1$ and so it defines a curve of versors $\mathsf{g}(t) \in \mathrm{Cl}(V)$ as the unique solution of this problem. Thus we get the exponential mapping in the Clifford algebra setting which maps the Lie algebra of bivectors to the Lie group of versors. By the recursive integration we get a formula for the exponential map as in (2.17) but with powers computed with respect to the geometric product, i.e.

$$\exp(\mathsf{A}t) := \mathsf{g}(t) = \sum_{k=0}^{\infty} \frac{t^k}{k!}\mathsf{A}^k. \tag{2.23}$$

2.5 Klein geometry

A geometry in the sense of Klein is a smooth manifold M together with a transitive and effective action of a Lie group G, i.e. for each two points $x, y \in M$ there exists a $g \in G$ such that $y = gx$ and the subgroup of G such that $\forall x \in M : gx = x$ is trivial. For a point $x \in M$ the subgroup $H_x = \{x \mid gx = x\}$ of the group G is called the *stabilizer* of point x. It follows from the definition of Klein geometry that the stabilizer is a closed and $H_x = \pi^{-1}(x)$, where $\pi : G \to M$ is defined by $x \mapsto gx$. It is easy to see that π induces a bijection $G/H_x \to M$. Hence the Klein geometry can be defined without the use of manifold M as follows. *Klein geometry* is a couple (G, H), where G is a Lie group and H is its closed subgroup such that the left coset space G/H is connected. The group G is called the main group of the geometry and G/H is called the homogeneous space of the geometry or simply the Klein geometry. Manifold $M \cong G/H$ is called a *model* of the Klein geometry.

The most well known examples of Klein geometries and the most important examples with respect to applications at the same time are the Euclidean geometry on an Euclidean space and the conformal geometry on a sphere. Let us mention some next examples before a more detailed description of these geometries: the spherical geometry $\mathrm{O}(n+1)/\mathrm{O}(n)$, the hyperbolic geometry $\mathrm{O}(1,n)/(\mathrm{O}(1) \times \mathrm{O}(n))$, the projective geometry $\mathrm{SL}(n+1)/P$, where P is the stabilizer od a line in $\mathbb{R}^{n+1}$ going through the origin, for details see [2, 3].

In this perspective, the Euclidean space E^n can be viewed as Klein geometry $\mathrm{E}(n)/\mathrm{O}(n)$ instead of the usual definition by the five Euclidean axioms. The main group $G = \mathrm{E}(n)$ is the group of Euclidean transformations in $\mathbb{R}^n$, defined as the semi-direct product of the orthogonal group $\mathrm{O}(n)$ and the vector group $\mathbb{R}^n$, i.e each element of the group is given by a composition of a rotation and a translation. The Euclidean space can be viewed also as $\mathrm{SE}(n)/\mathrm{SO}(n)$ if one considers only the transformations preserving orientation. In the standard presentation of Euclidean group, the Euclidean space E^n is identified with the affine hyperplane $M = \{x_1 = 1\} \subset \mathbb{R}^{n+1}$ and $\mathrm{E}(n)$ is identified with the group of all linear automorphims of $\mathbb{R}^{n+1}$ preserving the hyperplane and thus induce isometries in it. A matrix representation is given by

$$\mathrm{E}(n) = \Big\{ \begin{pmatrix} 1 & 0 \\ x & A \end{pmatrix} : A \in \mathrm{O}(n), x \in \mathbb{R}^n \Big\}. \tag{2.24}$$

The stabilizer $H = \mathrm{O}(n)$ of the first standard basis vector of $\mathbb{R}^{n+1}$ is formed by matrices of the same form but with $x = 0$. Thus a point $x \in E^n$ corresponding to vector $(1, x) \in \mathbb{R}^{n+1}$ is represented by the class of matrices (2.24) with a fixed $x \in \mathbb{R}^n$ and arbitrary $A \in \mathrm{O}(n)$, hence by a translation by vector $x \in \mathbb{R}^n$. The local structure of Euclidean geometry is given by the corresponding Lie algebra

$$\mathfrak{se}(n) = \Big\{ \begin{pmatrix} 0 & 0 \\ x & A \end{pmatrix} : A \in \mathfrak{so}(n), x \in \mathbb{R}^n \Big\} = \mathfrak{so}(n) \oplus \mathbb{R}^n. \tag{2.25}$$

The *conformal geometry* is the Klein geometry $\mathrm{SO}(n+1,1)/P$, where P is a so called parabolic subgroup and can be described as a semi-direct product of the conformal group $\mathrm{CO}(n)$ and the vector group $\mathbb{R}^n$. The basic model M of the conformal geometry is the sphere S^n which we view as the unit sphere embedded in the quadratic space $\mathbb{R}^{n+1,1}$ with a bilinear form B of signature $(n+1,1)$ by mapping $x \mapsto (x, 1)$. The points on the sphere are then in one to one correspondence with the cone of isotropic lines in $\mathbb{R}^{n+1,1}$ going through the origin, hence the subgroup P is the stabilizer of an isotropic line. It can be also described in terms of its Lie algebra $\mathfrak{p}$ as follows. In Witt basis (2.8) of the quadratic space $\mathbb{R}^{n+1,1}$, the matrix representation of the main group $\mathrm{SO}(n+1,1)$ is given by

$$\mathfrak{so}(n+1,1) = \Bigg\{ \begin{pmatrix} a & z & 0 \\ x & A & -z^T \\ 0 & -x^T & -a \end{pmatrix} : A \in \mathfrak{so}(n), z \in \mathbb{R}^{n*}, x \in \mathbb{R}^n, a \in \mathbb{R} \Bigg\} \tag{2.26}$$

and the stabilizer of the isotropic line e_1 is given by those matrices with $x = 0$, i.e. the Lie algebra of the parabolic subgroup P is $\mathfrak{p} = \mathfrak{so}(n) \oplus \mathbb{R} \oplus \mathbb{R}^{n*}$.

The Euclidean geometry can be viewed as a reduction of conformal geometry as follows. Comparing formulas (2.25) and (2.26) we see that there exists a homomorphism of Lie algebras $\mathfrak{se}(n) \to \mathfrak{so}(n+1,1)$ such that the subalgebra $\mathfrak{so}(n)$ maps to $\mathfrak{p}$. The composition of this homomorphism with the exponential map defines an injective homomorphism between connected components of identity of the corresponding Lie groups which factors to an injective map

$$E^n = \mathrm{SE}(n)/\,\mathrm{SO}(n) \to \mathrm{SO}(n+1,1)/P = S^n, \tag{2.27}$$

which transfers the Euclidean structure from E^n to the conformal sphere $S^n \subset \mathbb{R}^{n+1,1}$. This map is well known - it is the inverse to stereographic projection. According to the above description (2.26) on points, which we identify with translations, the map given by

$$\exp \begin{pmatrix} 0 & 0 & 0 \\ x & 0 & 0 \\ 0 & -x^T & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ x & 1_n & 0 \\ \frac{1}{2}x^T x & -x^T & 1 \end{pmatrix}. \tag{2.28}$$

Hence in the vector model in the Witt basis an Euclidean point given by translation $x \in \mathbb{R}^n$ is represented by the isotropic line $(1, x, \frac{1}{2}x^T x)$.

2.6 Geometric Algebra

By a general geometric algebra (GA) we mean a Clifford algebra where we consider also its Grassmann structure, i.e. we use both the wedge product and the geometric product, see sections 2.1 and 2.3. The understanding of a geometry in the sense of Klein, see section 2.5, allows to relate such an algebra to a concrete geometry. Namely, we say that $\mathrm{Cl}(V)$ is the Clifford algebra for Klein geometry (G, H) if there exists a model manifold $M \subseteq V$ such that the main group G acts on M transitively by orthogonal transformations from $O(V)$ with a stabilizer isomorphic to H. In particular it means that $G \subset \mathrm{SO}(V)$ is represented by versors of $\mathrm{Cl}(V)$, see section 2.4. In literature, the relation of the geometry to GA is usually given by specifying the natural inclusion map $J : M \hookrightarrow V$ which we refer to as the embedding. A list of examples of geometric algebras and related geometries is given below in section 3.

By geometric objects in algebra $\mathrm{Cl}(V)$ associated to Klein geometry $G/H \cong M$ we mean submanifolds $M \cap W$ where W is a subspace of vector space V. Since such an object is determined by vector subspace W, it can be represented via Plücker embedding by the projective class of a blade $\mathsf{w} \in \mathrm{Cl}(V)$ in the sense that for each $v \in V$ we have $v \wedge \mathbf{w} = 0$ if and only if $v \in W$. In particular, a subspace of dimension k is represented by a blade of grade k. In the language of geometric algebras, the representation of $M \cap W$ by such a blade w is called the *outer product null space (OPNS) representation.* In terms of the embedding $J : M \hookrightarrow V$ we have

$$M \cap W = \{x \in M : \; J(x) \wedge \mathbf{w} = 0\}. \tag{2.29}$$

The outer product computes joins of objects in GA in the following sense. The OPNS representation of object $M \cap (W_1 \oplus W_2)$ determined by a direct sum of subspaces is given by the projective class of a blade $\mathsf{w}_1 \wedge \mathsf{w}_2$, where w_1 and w_2 are OPNS representations of W_1 and W_2 respectively. Moreover, these subspaces have a nonempty intersection if and only if $\mathsf{w}_1 \wedge \mathsf{w}_2 = 0$. In a similar way, we can replace the outer product by the left contraction and we define the *inner product null space (IPNS) representation* w^* of $M \cap W$ as a blade of the dual algebra such that

$$M \cap W = \{x \in M : \ J(x) \cdot \mathbf{w}^* = 0\}. \tag{2.30}$$

It follows from the duality relation (2.5) that the blade $\mathsf{w}^* \in \Lambda V^*$ representing subspace W in this sense indeed is the dual blade to w with respect to duality (2.4), i.e. it is given by $\mathsf{w}^* = \mathsf{w} \cdot \mathsf{I}^*$, where I^* is a pseudoscalar in the dual algebra ΛV^*. It is easy to see from definitions that the dual blade w^* is the OPNS representation of the annihilator $W^\perp \subset V^*$, i.e. the space of linear forms that vanish on subspace W. Hence the outer product on the dual algebra represents the direct sum of annihilators, namely $\mathsf{w}_1^* \wedge \mathsf{w}_2^*$ is the OPNS representation of $W_1^\perp \oplus W_2^\perp = (W_1 \cap W_2)^\perp$ and hence its dual blade is the IPNS representation of the intersection of subspaces W_1 and W_2, thus the IPNS representation of the intersection of corresponding objects, if it is not zero. On the other hand, it is zero if and only if $W_1^\perp \cap W_2^\perp = (W_1 + W_2)^\perp = \emptyset$, which holds if and only if $W_1 + W_2 \neq V$. Hence, restricting V to the union $W_1 \cup W_2$ if necessary, the outer product in the dual algebra defines so called *regressive product* on ΛV

$$\mathsf{w}_1 \vee \mathsf{w}_2 := (\mathsf{w}_1^* \wedge \mathsf{w}_2^*)^* \tag{2.31}$$

which computes intersections of subspaces and hence intersection of corresponding objects. In the case of a nondegenerate bilinear form, we may identify the vector space V with its dual V^* according to (2.7) and consequently we may identify the induced Grassmann algebra with its dual. In this identification, the annihilator $W^\perp$ is equal to the orthogonal complement of W and its OPNS representation is w^* which can be viewed as an element of the same algebra as w.

The Clifford structure of GA allows a representation of orthogonal transformations by versors. Indeed, transformations of points in $M \cong G/H$ are given by group $G \subseteq \mathrm{SO}(V)$ which is represented in GA by formula (2.19) by definition. The linearity of this formula implies that it extends to blades and defines an automorphism of the Grassmann algebra, also called outermorphism in literature, i.e. for each versor $\mathsf{g} \in \mathrm{Cl}(V)$ and vectors $w_1, w_2 \in V$ we have

$$\mathsf{g}(w_1 \wedge w_2)\mathsf{g}^{-1} = \mathsf{g}w_1\mathsf{g}^{-1} \wedge \mathsf{g}w_1\mathsf{g}^{-1}.$$

Hence the induced transformations of objects are represented by the same formula as transformations of points. Namely, the reflection of object $M \cap W$ with OPNS representation w in a unit vector v is given by

$$\mathsf{w} \mapsto -v\mathsf{w}v \tag{2.32}$$

while its $\mathrm{SO}(V)$ transformation given by a spinor R, i.e. $\mathsf{R}\widetilde{\mathsf{R}} = 1$, reads

$$\mathsf{w} \mapsto \mathsf{R}\mathsf{w}\widetilde{\mathsf{R}}. \tag{2.33}$$

Next maps that can be easily expressed in terms of GA operations are orthogonal projections, i.e. linear maps with the idempotent property which are orthogonal with respect to bilinear form B. The classical formula $w \mapsto B(w, v^{-1})v$ for the projection of a vector to the one-dimensional subspace generated by $v \in V$ extends to blades as follows. The orthogonal projection of a blade w to a subspace represented by an invertible blade v is given by

$$\mathsf{w} \mapsto (\mathsf{w} \cdot \mathsf{v}^{-1}) \cdot \mathsf{v}. \tag{2.34}$$

In computations of orthogonal projections of objects in GA this formula is usually simplified to $(\mathsf{w} \cdot \mathsf{v}) \cdot \mathsf{v}$ since $\mathsf{v}^2 \in \mathbb{R}$ holds for each blade and thus v differs from its inverse v^{-1} only by a scalar multiple. It means that the two blades define the same projective class and thus represent the same object in GA if this multiple is not zero. The formula can be generalized also to null blades, $\mathsf{v}^2 = 0$, by replacing the inverse by the Clifford conjugate (2.11), see section 3.2 in [34] for more details.

3 Examples of geometric algebras and their applications

In this chapter, concrete examples of geometric algebras and their applications are briefly described. We start with algebras related to Euclidean space and Euclidean transformations since these have probably the highest potential towards applications in engineering. Namely, we treat the geometric algebras generated by quadratic spaces $V = \mathbb{R}^{n,0}$, $V = \mathbb{R}^{n,0,1}$ and $V = \mathbb{R}^{n+1,1}$. However we work with a general dimension the most interesting case $n = 3$ is emphasized. Then we introduce a geometric algebra which allows for working with general conics in plane. The last example is a complex geometric algebra which allows for a direct realization of quantum Dirac formalism.

3.1 Geometric algebra $\mathbb{G}_n$

The geometric algebra $\mathbb{G}_n = \mathrm{Cl}(\mathbb{R}^n)$ is defined by the positive definite form of signature $(n, 0)$. This form is usually called Euclidean since it induces the standard scalar product on $\mathbb{R}^n$ and the Euclidean distance. Nevertheless, it is not the geometric algebra for the Euclidean geometry in the sense described above because the algebra of bivectors is too "small". Namely, by (2.22) the algebra is isomorphic to $\mathfrak{so}(n)$, and not to the Euclidean Lie algebra. In other words, we cannot represent translations in this geometric algebra. Algebra $\mathbb{G}_n$ is suitable for representing vector subspaces of $V = \mathbb{R}^n$ and general rotations $SO(n)$, also reflections and orthogonal projections to vector subspaces. On the other hand, operations from $\mathbb{G}_n$ can used in the Euclidean space if we fix the origin and thus identify E_n with $\mathbb{R}^n$. So we can represent lines, planes etc. going through the origin and rotations around origin.

Concretely, a bivector $u \wedge v$ represents the plane through origin with direction vectors $u, v \in \mathbb{R}^n$ in the sense of (2.29), and its exponential represents a rotation in this plane. Indeed, the versor $\exp(u \wedge vt)$ commute with its generator $u \wedge v$ by the definition of exponential map (2.16), and hence this versor defines a rotation that leaves the plane given by vectors $u, v \in \mathbb{R}^n$ invariant. If we want a rotation by an explicit angle we need to normalize the generating bivector in the following way.

Proposition 3.1. *The rotation in plane defined by vectors $u, v \in \mathbb{R}^n \subset \mathbb{G}_n$ by angle φ is in geometric algebra $\mathbb{G}_n$ given by versor*

$$\mathsf{R} = \exp\left(\frac{1}{2}\varphi\frac{u \wedge v}{\sqrt{-(u \wedge v)^2}}\right) = \cos\frac{\varphi}{2} + \sin\frac{\varphi}{2}\frac{u \wedge v}{\sqrt{-(u \wedge v)^2}}. \tag{3.1}$$

Proof. Obviously, the bivector in the proposition is normed such that it squares to -1. The equality in (3.1) then follows directly from the definition of exponential map (2.16). To prove the rest let us choose vectors $\bar{u}, \bar{v}$ defining the plane such that they are of unit length and the angle between them is $\varphi/2$. Then the desired rotation can be expressed as a composition of two reflections of the form (2.18), one in vector $\bar{u}$ and the second in vector $\bar{v}$. Thus it is represented by versor

$$\mathsf{R} = \bar{v}\bar{u} = \bar{v} \cdot \bar{u} + \bar{v} \wedge \bar{u} = \cos\frac{\varphi}{2} + \sin\frac{\varphi}{2}\frac{\bar{u} \wedge \bar{v}}{\sqrt{-(\bar{u} \wedge \bar{v})^2}}.$$

The last equality follows from the fact that $B(u, v)$ in the algebra $\mathbb{G}_n$ is the standard inner product and for the square of a bivector we compute $(\bar{u} \wedge \bar{v})^2 = (\bar{u} \cdot \bar{v})^2 - \bar{u}^2\bar{v}^2 = -\sin^2\frac{\varphi}{2}$. The normed bivector obviously does not depend on the choice of vectors u, v in the plane $u \wedge v$. □

Example 3.2. The lowest dimensional geometric algebra suitable for the representation of rotations in 3D space is $\mathbb{G}_3$. Rotations from SO(3) are represented by rotors R in the sense of equation (2.33). The dimension $n = 3$ is special because the bivector representing the plane of rotation is dual to vector that represents the rotation axis. If the pseudoscalar satisfies $\mathsf{I}^2 = -1$, which is the case of the usual choice $\mathsf{I} = e_1e_2e_3$, where e_1, e_2, e_3 is an orthonormal basis of $\mathbb{R}^3$, and if n is the unit vector in the direction of the rotation axis, then n^* is the bivector that determines the rotation plane and that is normed $(n^*)^2 = -1$. Hence the rotor (3.1) for the rotation given by axis n and angle φ has a simple form in $\mathbb{G}_3$

$$\mathsf{R} = \exp(\frac{1}{2}\varphi n^*) = \cos\frac{\varphi}{2} + n^* \sin\frac{\varphi}{2}. \tag{3.2}$$

This representation of rotations actually is identical with the representation by quaternions (1.4), because the subalgebra of $\mathbb{G}_3$ of elements of even grade is isomorphic to the algebra of quaternions $\mathbb{H}$. Indeed, in an orthonormal basis e_1, e_2, e_3 this isomorphism is realized by $i \mapsto -e_2e_3, j \mapsto -e_3e_1, k \mapsto -e_1e_2$. The benefit of $\mathbb{G}_3$ in contrast to quaternions is that we can also easily represent all usual vector operations. For example, scalar product is $u \cdot v$, the cross product $u \times v = (u \wedge v)^*$, and the triple product $[uvw] = (u \wedge v \wedge w)^*$. In this sense, we may view $\mathbb{G}_3$ as a unification of quaternions and vector algebra.

Applications of $\mathbb{G}_n$

The benefits of quaternions in representing rotations in 3D space are well known and they are frequently used in various fields in computer graphics and engineering. As pointed out in the previous example, geometric algebra $\mathbb{G}_3$ gives an equivalent representation of rotations and basic vector operations in a new unified language. In particular, the rotor is easily constructed from the knowledge of the axis of rotation and the angle of rotation by (3.2) and a composition of rotations is just the geometric product of the corresponding elements in $\mathbb{G}_3$. In general, geometric algebra $\mathbb{G}_n$ gives a smart representation of rotations and reflections in orthogonal group $O(n)$ that can consequently help to solve or to reduce a given problem. An example of such an application is given in our paper [31] that is enclosed in Appendix 1. Here we used this geometric algebra representation to study a specific problem in control theory, namely an invariant control problem on Carnot group of step two.

3.2 Projective geometric algebra

To obtain a true geometric algebra for the Euclidean geometry we need to represent by versors not only rotations but also translations. The minimal such algebra is called the *projective geometric algebra* (PGA). It is the algebra $\mathbb{G}_{n,0,1}$ defined by a degenerate bilinear form of signature $(n, 0, 1)$ on $V = \mathbb{R}^{n+1}$. In this vector space (and hence in this algebra), the Euclidean space E^n can be identified with hyperplane $M = \{x_1 = 1\} \subset \mathbb{R}^{n+1} = V$ as described in 2.5, and the Euclidean transformations are given by orthogonal transformations of $\mathbb{R}^{n+1}$.

Indeed, quadratic space $\mathbb{R}^{n,0,1}$ is a direct sum of anisotropic part $\mathbb{R}^{n,0}$ and one dimensional kernel $\operatorname{Ker} B$, and thus for the algebra of bivectors we compute

$$\Lambda^2\mathbb{R}^{n,0,1} = \Lambda^2\mathbb{R}^{n,0} \oplus \big(\mathbb{R}^{n,0} \wedge \operatorname{Ker} B\big) \cong \mathfrak{so}(n) \oplus \mathbb{R}^n = \mathfrak{se}(n), \tag{3.3}$$

which is the Lie algebra to Euclidean Lie group. The bivectors from $\Lambda^2\mathbb{R}^{n,0}$ generate rotations while bivectors from $\mathbb{R}^{n,0} \wedge \operatorname{Ker} B$ generate translations. With respect to common conventions, it is convenient to choose $-1/2x \wedge e_\infty$, for the generator of translation by vector $x \in \mathbb{R}^n$, where we denote by $e_\infty \in \operatorname{Ker} B$ the element from the kernel. Since this bivector squares to zero, the corresponding versor, called *translator*, has the form

$$\mathbf{T} = \exp(-\frac{1}{2}x \wedge e_\infty) = 1 - \frac{1}{2}x \wedge e_\infty. \tag{3.4}$$

To get a model of Euclidean geometry we need a set of algebra elements on which these translators act simply transitively, i.e. transitively with a trivial stabilizer. In contrast to $\mathbb{G}_n$, points cannot be represented by projective classes of vectors, i,e, by their homogeneous coordinates, since the translators act trivially on $\mathbb{R}^{n,0,1}$, see Remark 3.4 below. Hence the points in PGA are represented in a dual way, by n-dimensional subspaces.

Proposition 3.3. *Projective classes of nonisotropic multivectors of grade n in $\mathbb{G}_{n,0,1}$ define a model of the n-dimensional Euclidean geometry.*

Proof. Let us first describe projective classes of nonisotropic multivectors in $\Lambda^n\mathbb{R}^{n,0,1} \subset \mathbb{G}_{n,0,1}$. Since space $\mathbb{R}^{n,0,1}$ decomposes into its anisotropic part $\mathbb{R}^{n,0}$ and kernel $\operatorname{Ker} B$, we get that

$$\Lambda^n\mathbb{R}^{n,0,1} = \Lambda^n\mathbb{R}^{n,0} \oplus (\operatorname{Ker} B \wedge \Lambda^{n-1}\mathbb{R}^{n,0}).$$

The square of elements of one-dimensional space $\Lambda^n\mathbb{R}^{n,0} \subset \mathbb{G}_{n,0,1}$ is nonzero because the restriction of bilinear form B to $\mathbb{R}^{n,0}$ is nondegenerate. On the other hand, each element $\mathsf{u} \in \operatorname{Ker} B \wedge \Lambda^{n-1}\mathbb{R}^{n,0}$ squares to zero since it contains a vector from the kernel. Moreover, also its product with elements of $\Lambda^n\mathbb{R}^{n,0}$ vanishes from the same reason. Hence the projective class of each nonisotropic multivector in $\Lambda^n\mathbb{R}^{n,0,1}$ is of the form $[\mathsf{I}_n + \mathsf{u}]$, where $\mathsf{I}_n \in \Lambda^n\mathbb{R}^{n,0}$ denotes the pseudoscalar of the anisotropic space $\mathbb{R}^{n,0} \subset \mathbb{R}^{n,0,1}$. It is easy to verify that the translators (3.4) really act transitively on these classes. Indeed, the infinitesimal action is

$$[x \wedge e_\infty, \mathsf{I}_n + \mathsf{u}] = -2e_\infty \wedge (x \lrcorner \mathsf{I}_n),$$

and we get each element of $\operatorname{Ker} B \wedge \Lambda^{n-1}\mathbb{R}^{n,0}$ by a suitable choice of vector $x \in \mathbb{R}^n$. Moreover, the stabilizer of pseudoscalar I_n is given by versors generated by bivectors $\Lambda^2\mathbb{R}^{n,0} \cong \mathfrak{so}(n)$. Hence the classes $[\mathsf{I}_n + \mathsf{u}]$ define a model of the Euclidean geometry, see 2.5. □

Remark 3.4. The Euclidean space cannot be represented in PGA by vectors since the generators of translations act trivially on $\mathbb{R}^{n,0,1} \subset \mathbb{G}_{n,0,1}$. Indeed, by the definitions of geometric and wedge product the infinitesimal action of a bivector $u \wedge e_\infty \in \mathbb{R}^{n,0} \wedge \operatorname{Ker} B$ on a vector $v \in \mathbb{R}^{n,0,1}$ is equal to

$$[u \wedge e_\infty, v] = B(u, v)e_\infty - B(e_\infty, v)u - B(v, u)e_\infty + B(u, e_\infty)v = 0.$$

The geometric objects in PGA are Euclidean (affine) subspaces. Indeed, the intersection of a linear subspace $W \subset \mathbb{R}^{n+1}$ with the hyperplane $M = \{x_{n+1} = 1\}$ that models the Euclidean space then defines uniquely an affine subspace $M \cap W$ in E^n. Thanks to isomorphism $\Lambda^k\mathbb{R}^{n+1} \cong \mathbb{R}^{(n+1)*}$, see (2.4), the Euclidean points can be also seen as elements in the dual projective space $\mathbb{P}\mathbb{R}^{(n+1)*}$, i.e. as projective classes of linear forms on $\mathbb{R}^{n+1}$. Each subspace W, and thus also $M \cap W$, is then uniquely represented by an element in the dual Grassmann algebra, see section 2.1. This dual picture implies that, in contrast to $\mathbb{G}_n$, the outer product computes intersections of affine subspaces and $\vee$ generates affine subspaces. Namely, if $\mathbf{X} \in \Lambda^n\mathbb{R}^{n,0,1} \subset \mathbb{G}_{n,0,1}$ represents a point $x \in E^n$, then multivector w represents affine subspace $W \cap M$ in the sense

$$\mathbf{X} \vee \mathbf{w} = 0 \text{ if and only if } x \in W \cap M. \tag{3.5}$$

Since the bilinear form in PGA is degenerate, the duality (2.4) cannot be understood as an operation on $\mathbb{G}_{n,0,1}$; there does not exist representation dual to (3.5) in PGA. The map given by $\mathsf{w} \mapsto \mathsf{w}^* = \mathsf{w}\mathsf{I}$, where I is the pseudoscalar in $\mathbb{G}_{n,0,1}$, does not determine affine subspace but represents only the orthogonal complement to vector space W.

In addition to Euclidean (affine) subspaces and their translations, one can represent general rotations, reflections and orthogonal projections in PGA. The most important formulas are summarized in the following proposition.

Proposition 3.5. *(a) Geometric objects in PGA. A point $x \in E^n$ is in $\mathbb{G}_{n,0,1}$ represented by a projective class of multivectors*

$$\mathbf{X} = \mathsf{I}_n + e_\infty \wedge (x \,\lrcorner\, \mathsf{I}_n), \tag{3.6}$$

where $\mathsf{I}_n \in \Lambda^n \mathbb{R}^{n,0}$ and $e_\infty \in \operatorname{Ker} B$. A Euclidean subspace of dimension k defined by points $\mathbf{X}_1, \dots, \mathbf{X}_{k+1}$ is represented by

$$\mathsf{w} = \mathbf{X}_1 \vee \cdots \vee \mathbf{X}_{k+1} \in \mathbb{P}(\Lambda^{n-k}\mathbb{R}^{n+1}). \tag{3.7}$$

and the intersection of subspaces $\mathsf{w}_1, \mathsf{w}_2$ is given by $\mathsf{w}_1 \wedge \mathsf{w}_2$.
(b) Transformations in PGA. A general rotation in point x is given by versor

$$\mathsf{R}_x = \mathbf{T}\mathsf{R}\tilde{\mathbf{T}}, \tag{3.8}$$

where $\mathbf{T}$ is a versor for translation given by (3.4)*, $\tilde{\mathbf{T}}$ is its reverse and R is a versor for rotation given by* (3.1)*. A reflection of Euclidean subspace w in hyperplane represented by vector $\pi \in \mathbb{R}^{n+1}$ is*

$$\mathsf{w} \mapsto \pi \mathsf{w} \pi, \tag{3.9}$$

and the orthogonal projection of Euclidean subspace w to subspace v is given by

$$\mathsf{w} \mapsto (\mathsf{w} \cdot \mathsf{v}) \wedge \mathsf{v}. \tag{3.10}$$

Proof. (a) In the Klein sense, a point of the Euclidean space is identified with the the translation to this point. Hence the mutivector (3.6) representing point $x \in E^n$ is obtained by acting by translator (3.4) to pseudoscalar I_n that represents the origin in our model

$$\mathbf{X} = \mathbf{T}\mathsf{I}_n\tilde{\mathbf{T}} = \mathsf{I}_n + e_\infty \wedge (x \,\lrcorner\, \mathsf{I}_n).$$

The representation of Euclidean subspaces (3.7) and their intersections follows directly from (3.5) and from the general description of geometric objects in section 2.6.
(b) The description of PGA bivectors in (3.3) implies that rotations are generated by elements of $\Lambda^2\mathbb{R}^{n,0}$, and thus the rotors have the same form in PGA as in $\mathbb{G}_n$, viz (3.1). Indeed, the rotation of point $\mathbf{X}$ given by formula (3.6) is given by

$$\mathsf{R}\mathbf{X}\tilde{\mathsf{R}} = \mathsf{I}_n + e_\infty \wedge (\mathsf{R}x\tilde{\mathsf{R}} \,\lrcorner\, \mathsf{I}_n) = \mathbf{X}_{\mathsf{R}x\tilde{\mathsf{R}}},$$

which is the rotated point by (3.1). The equality follows from the fact that the action of versors keep the products in geometric algebra invariant and that vector e_∞ is perpendicular to $\mathbb{R}^{n,0}$. However R still represents only rotations in the origin, a general rotation in point x can be expressed as a composition of translation $\tilde{\mathbf{T}}$ from the point x to origin, rotation R in origin and the translation $\mathbf{T}$ back to the point x. Since compositions of maps is represented by geometric products of corresponding versors, we get equation (3.8) for a general rotation.

In a similar way, by translating the situation to origin and by using the dual representation in $\mathbb{G}_n$, the equations for reflections and orthogonal projections can be derived. Concretely, a translator $\tilde{\mathbf{T}}$ that translates an affine hyperplane $\pi \in \mathbb{R}^{n,0,1}$ to origin satisfies $\tilde{\mathbf{T}}\pi\mathbf{T} = \pi'$, where $\pi' \in \mathbb{R}^{n,0}$. The reflection in π' is represented by $R_{\pi'}$ in $\mathbb{G}_n$, see (2.18), and thus the reflection in π is given by

$$\mathsf{w} \mapsto \mathbf{T}R_{\pi'}(\tilde{\mathbf{T}}\mathsf{w}\mathbf{T})\tilde{\mathbf{T}} = \mathbf{T}\pi'\tilde{\mathbf{T}}\mathsf{w}\mathbf{T}\pi'\tilde{\mathbf{T}} = \pi\mathsf{w}\pi.$$

The orthogonal projection of w to v is given by the intersection of Euclidean subspace v with subspace which contains w and orthogonal to v. Hence to prove equation (3.10) we only need to show that $\mathsf{w}\cdot\mathsf{v}$ represents the latter Euclidean subspace. To do that we decompose the blades as $\mathsf{w} = \mathsf{w}' + \mathsf{w}_\infty$, $\mathsf{v} = \mathsf{v}' + \mathsf{v}_\infty$ into blades that does not contain and contain the vector form kernel, respectively. The parts v', w' represent the associated vector spaces to the corresponding affine spaces while $\mathsf{v}_\infty, \mathsf{w}_\infty$ represents their distance from origin. Any contraction by these elements is always zero and the decomposition is invariant under translations. Hence we get

$$\mathsf{w} \cdot \mathsf{v} = \mathsf{v} \rfloor \mathsf{w} = \mathbf{T}(\mathsf{v}' \rfloor \mathsf{w}')\tilde{\mathbf{T}},$$

where translator $\tilde{\mathbf{T}}$ shifts w to origin, i.e. $\tilde{\mathbf{T}}\mathsf{w}\mathbf{T} = \mathsf{w}'$ holds. Indeed, $\mathsf{v}' \rfloor \mathsf{w}'$ is the dual representation in $\mathbb{G}_n$ of vector space that contains w' and is perpendicular to v'. □

Example 3.6. Let us treat PGA for the 3D Euclidean space in more detail. The algebra is induced by $\mathbb{R}^4$ with a degenerate quadratic form B of signature $(3,0,1)$. Let e_1, e_2, e_3 be an orthonormal frame with respect to B of the nondegenerate part $\mathbb{R}^{3,0} \subset \mathbb{R}^{3,0,1}$ and let e_∞ be a vector from the kernel of B. If we denote $\mathbf{e}_{ij} = e_i \wedge e_j$ for simplicity, then by (3.6) an Euclidean point with coordinates (x, y, z) in this basis is represented by the projective class of multivector

$$\mathbf{X} = \mathbf{e}_{123} + x\mathbf{e}_{\infty 23} - y\mathbf{e}_{\infty 13} + z\mathbf{e}_{\infty 12}. \tag{3.11}$$

The point can be also given by the intersection of three planes $\mathbf{p}_1 \wedge \mathbf{p}_2 \wedge \mathbf{p}_3$ or by the intersection of a plane and a line $\mathbf{p} \wedge \boldsymbol{\ell}$. A plane given by $ax + by + cz + d = 0$ is represented by the projective class of vector

$$\mathbf{p} = ae_1 + be_2 + ce_3 + de_\infty \tag{3.12}$$

and can be also given by a line and a point $\boldsymbol{\ell} \vee \mathbf{X}$. A line is represented by the projective class of multivector of degree two whose coefficients are the Plücker coordinates

$$\boldsymbol{\ell} = p_1\mathbf{e}_{23} + p_2\mathbf{e}_{31} + p_3\mathbf{e}_{12} + d_1\mathbf{e}_{1\infty} + d_2\mathbf{e}_{2\infty} + d_3\mathbf{e}_{3\infty}, \tag{3.13}$$

where (p_1, p_2, p_3) is the unit directional vector of the line and (d_1, d_2, d_3) is its distance from the origin. The line can be also defined by two points $\mathbf{X}_1 \vee \mathbf{X}_2$ or by the intersection of two planes $\mathbf{p}_1 \wedge \mathbf{p}_2$.

General rotations (3.8) in 3D can be easily expressed by the rotation axis. Namely, if $\boldsymbol{\ell}$ is a normalized representation of the axis in the sense that $\boldsymbol{\ell}^2 = -1$, i.e. $\boldsymbol{\ell}$ is of the form as in (3.13), then the rotation by angle φ is given by versor

$$\mathsf{R}_\ell = \exp(\frac{1}{2}\varphi\boldsymbol{\ell}) = \cos\frac{\varphi}{2} + \boldsymbol{\ell}\sin\frac{\varphi}{2}. \tag{3.14}$$

By proposition 3.5 we can easily express also the reflection in a plane and an orthogonal projection to a plane. A basic example of computation in 3D PGA is displayed in Figure 1. The figure is a screenshot from the visualization created by ganja.js – Geometric Algebra code generator for javascript, [7].

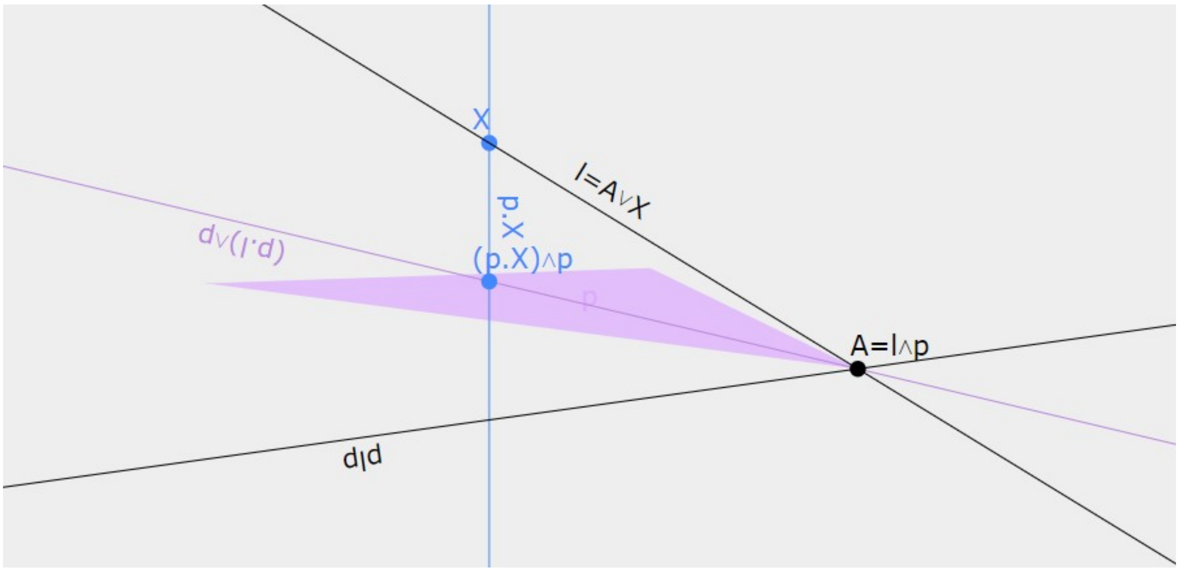


Figure 1: The reflection of line $\boldsymbol{\ell}$ in plane $\mathbf{p}$ is represented by the projective class of $\mathbf{p}\boldsymbol{\ell}\mathbf{p}$. The blade $\boldsymbol{\ell}\cdot\mathbf{p}$ represents the plane that contains line $\boldsymbol{\ell}$ and is perpendicular to plane $\mathbf{p}$, hence blade $(\boldsymbol{\ell}\cdot\mathbf{p})\wedge\mathbf{p}$ represents the orthogonal projection of line $\boldsymbol{\ell}$ to plane $\mathbf{p}$. Similarly, $\mathbf{X}\cdot\mathbf{p}$ is the perpendicular from $\mathbf{X}$ to plane $\mathbf{p}$ and thus $(\mathbf{X}\cdot\mathbf{p})\wedge\mathbf{p}$ is the orthogonal projection of point $\mathbf{p}$ to this plane.

Applications of PGA

As PGA is a "dual" to the classical homogeneous representation, it is an ideal representation of Euclidean space and Euclidean transformations. In particular, it allows a quaternion like representation of general rotations. Obviously, for applications the 3D case from the previous example is the most used. Promising applications appear mainly in computer graphics since the language of geometric algebra is intuitive and universal while minimal data are used. A significant research group in this field is the group in Belgium and Netherlands; their work includes [4, 7, 8]. There are indications that PGA could replace quaternions in all 3D engines in near future. However PGA is a minimal representation of Euclidean geometry, the degeneracy of generating bilinear form can cause problems, mainly in implementations. This problem can be solved by viewing PGA as an subalgebra in a bigger, non degenerate algebra as shown in our paper [29], enclosed in Appendix 2.

3.3 Conformal geometric algebra

If we want to define a geometric algebra for Euclidean geometry with a nondegenerate bilinear form, then it is necessary to add a vector e_0 that is the Witt counterpart to the null vector e_∞ from PGA. Then we have two null vectors $e_\infty^2 = e_0^2 = 0$ but the kernel of B is nontrivial since $B(e_0, e_\infty) = -1$. Since the isotropic space has an indefinite signature $(1,1)$, we get a nondegenerate bilinear form on $V = \mathbb{R}^{n+2}$ with signature $(n+1, 1)$ by this construction. The corresponding geometric algebra $\mathbb{G}_{n+1,1}$ is called the *conformal geometric algebra* (CGA).

According to (2.22), we have $\Lambda^2\mathbb{R}^{n+1,1} \cong \mathfrak{so}(n+1,1)$ for the algebra of bivectors in CGA. Under this isomorphism, the action of bivectors on $\mathbb{R}^{n+1,1}$ given by equation (2.21) corresponds to the standard matrix representation of Lie algebra described in 2.5. This action is transitive on the projectivization of the cone of null vectors and its stabilizator is the parabolic subalgebra $\mathfrak{p}$ described by equation (2.26). Indeed, the bivectors that act trivially on the null vector e_0 are exactly those that do not contain this vector and the bivector $e_0 \wedge e_\infty$, i.e.

$$\mathfrak{p} = \mathfrak{so}(n) \oplus \mathbb{R}^{n*} \oplus \mathbb{R} \cong \{u \wedge v, u \wedge e_\infty, e_0 \wedge e_\infty\}, \tag{3.15}$$

where $u, v \in \mathbb{R}^{n,0}$. Hence the action of versors on the projectivized null vectors in $\mathbb{G}_{n+1,1}$ models the conformal geometry, see section 2.5. The model of Euclidean geometry is then obtained by defining M to be the set of all but one projective classes of null vectors. Indeed, if we denote by $[e_\infty]$ the missing class and if we set $e_{n+1} = e_\infty$, then the inverse stereographic projection (2.27) defines a bijective mapping $E^n \to M$. The vector e_∞ is the Witt counterpart to null vector e_0 that represents the origin and it can be viewed as representing the infinity.

The intersection of vector subspace $W \subset \mathbb{R}^{n+1,1}$ of dimension $k+2$ with the null cone is a cone in W, hence the null lines on this cone can be identified with a sphere again, this time a sphere of dimension k. If this cone contains the line e_∞, i.e. the source of the projection, then W represents affine subspace of dimension k in E^n. Hence the geometric objects in CGA representable in the sense of (2.29) are exactly the generalized spheres, i.e. spheres and affine spaces which we see as spheres containing the infinity. In particular, a point is not natural objects and it should be considered as a limit sphere with zero radius, or as so called flat point $\mathbf{X} \wedge e_\infty$, which is a sphere of dimension zero going through the infinity.

Thanks to the nondegeneracy of the bilinear form the duality (2.3) defines an unary operation in CGA, and so we have two mutually dual representations for generalized spheres available in CGA. The outer representation (2.29) is suitable for construction of spheres from points or from spheres of lower dimension while from the inner representation (2.30) of the sphere we can easily read off its radius and its center.

Similarly as in PGA, there exist versors for translations and rotations in CGA. Moreover, we have an additional versor for scaling. Next to reflections in hyperplanes and orthogonal projections to affine spaces we can similarly represent also reflections in spheres and orthogonal projections on spheres. The main results are summarized in the following proposition which is left without proof. The proof follows easily from facts in the text above and is analogous to the proof of proposition 3.5.

Proposition 3.7. *(a) Geometric objects in CGA. A point $x \in E^n$ is represented in $\mathbb{G}_{n+1,1}$ by the projective class of vector*

$$\mathbf{X} = e_0 + x + \frac{1}{2}x^2 e_\infty, \tag{3.16}$$

where e_0, e_∞ is a Witt pair of null vectors. In particular, for normalized representations of points $\mathbf{X}^2 = \mathbf{Y}^2 = 1$ we have

$$\mathbf{X} \cdot \mathbf{Y} = -\frac{1}{2}\|x - y\|^2, \tag{3.17}$$

where $\|x - y\|$ is the Euclidean norm in $\mathbb{R}^n$. An affine space of dimension k defined by points $\mathbf{X}_1, \dots, \mathbf{X}_{k+1}$ is represented by blade

$$\mathsf{w} = \mathbf{X}_1 \wedge \cdots \wedge \mathbf{X}_{k+1} \wedge e_\infty \tag{3.18}$$

and its dual representation w^* coincides with its representation in PGA. A sphere of dimension k defined by points $\mathbf{X}_1, \dots, \mathbf{X}_{k+2}$ is represented by blade

$$\mathbf{s} = \mathbf{X}_1 \wedge \cdots \wedge \mathbf{X}_{k+2} \tag{3.19}$$

and the intersection of spheres $\mathbf{s}_1, \mathbf{s}_2$ is given by $\mathbf{s}_1 \vee \mathbf{s}_2$. If $\mathbf{s}$ is a sphere of the maximal dimension $n-1$, then its dual representation is given by vector

$$\mathbf{s}^* = \mathbf{S} + \frac{1}{2}\rho^2 e_\infty, \tag{3.20}$$

where $\mathbf{S}$ is the normalized representation of the sphere center and ρ is its radius.
(b) Transformations in CGA. Rotations and translations are represented by the same versors as in PGA. The scaling by factor r^2 is given by versor

$$\boldsymbol{\mathcal{S}} = \frac{r^2+1}{2r} + \frac{r^2-1}{2r} e_0 \wedge e_\infty \tag{3.21}$$

The reflection of generalized sphere $\mathbf{s}$ in a hyperplane $\boldsymbol{\pi} \in \Lambda^{n+1}\mathbb{R}^{n+1,1}$ is given by formula $\mathbf{s} \mapsto \boldsymbol{\pi}\mathbf{s}\boldsymbol{\pi}$, similarly as for reflections in PGA. If $\boldsymbol{\pi}$ represents a sphere of maximal dimension, then this formula defines a spherical inversion. The orthogonal projection of affine space w to a generalized sphere $\mathbf{s}$ is $\mathsf{w} \mapsto (\mathsf{w} \cdot \mathbf{s}) \cdot \mathbf{s}$.

Example 3.8. Let us describe the 3D case in more detail again. According to (3.16), a point of Euclidean space E^3 given in Cartesian coordinate system with unit direction vectors $e_1, e_3, e_3 \in \mathbb{R}^3$ by vector (x, y, z) is represented in CGA by vector

$$\mathbf{X} = e_0 + xe_1 + ye_2 + ze_3 + \frac{1}{2}(x^2 + y^2 + z^2)e_\infty. \tag{3.22}$$

The representation of a plane given by three points is $\mathbf{p} = \mathbf{X}_1 \wedge \mathbf{X}_2 \wedge \mathbf{X}_3 \wedge e_\infty$ and its dual representation $\mathbf{p}^*$ coincides with its representation in PGA, see equation (3.12). Similarly, a line is given by $\boldsymbol{\ell} = \mathbf{X}_1 \wedge \mathbf{X}_2 \wedge e_\infty$ and its dual coincides with (3.13). A sphere generated by four points in CGA is represented by $\mathbf{s} = \mathbf{X}_1 \wedge \mathbf{X}_2 \wedge \mathbf{X}_3 \wedge \mathbf{X}_4$ and its dual representation is $\mathbf{s}^* = \mathbf{S} + \frac{1}{2}\rho^2 e_\infty$. A circle can be expressed either by three generating points $\mathbf{c} = \mathbf{X}_1 \wedge \mathbf{X}_2 \wedge \mathbf{X}_3$ or in the dual way by $\mathbf{c}^* = (\mathbf{S} + \frac{1}{2}\rho^2 e_\infty) \wedge \boldsymbol{\pi}^*$, i.e. as the intersection of a sphere and the plane where the circle lies.

A rotation in E^3 by axis $\boldsymbol{\ell}$ is given by the same rotor as in PGA, see (3.14). Also reflections in planes and orthogonal projections to planes or lines have the same representations as in PGA. Moreover, in CGA we can represent in the analogous way spherical inversions and orthogonal projections to spheres. A concrete example of such a computation in CGA is shown in Figure 2. Again, the figure is a screenshot from a visualization created in ganja.js – Geometric Algebra code generator for java script, [7].

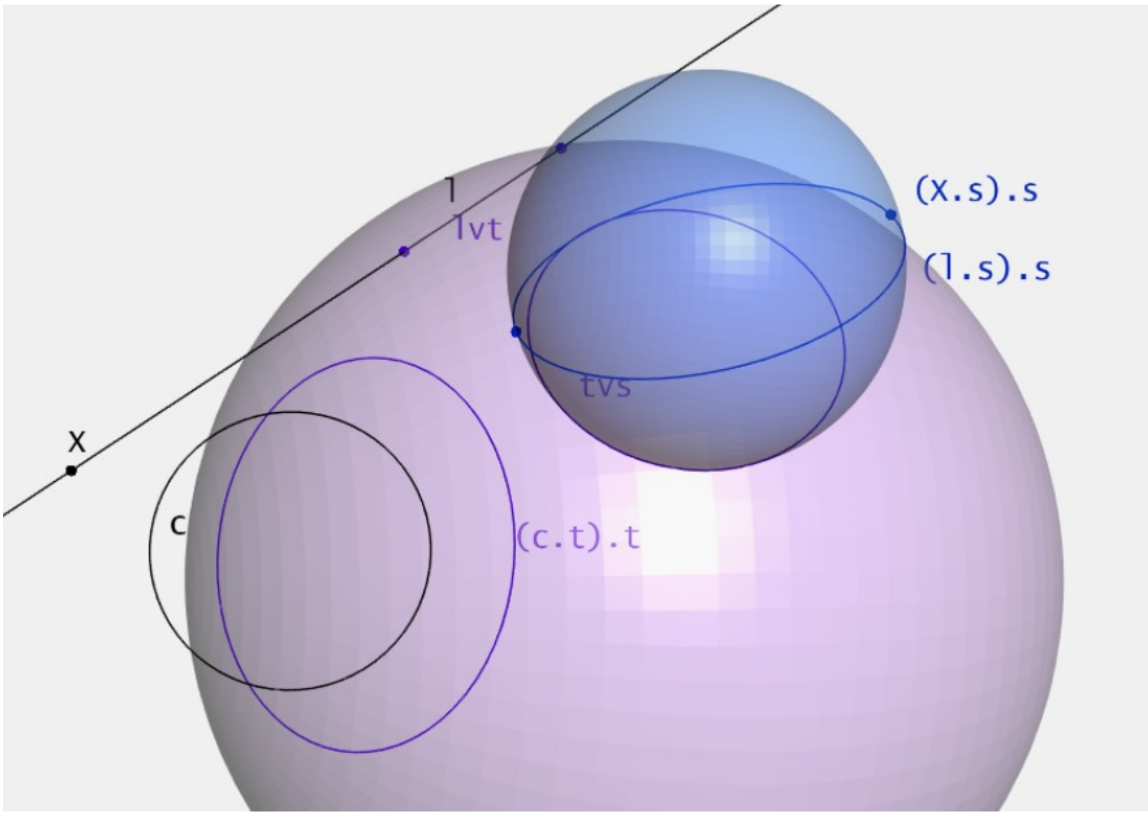


Figure 2: Given two spheres $\mathbf{s}, \mathbf{t}$, a circle $\mathbf{c}$, a line $\boldsymbol{\ell}$ and a point $\mathbf{X}$, the intersection of the spheres is the circle $\mathbf{s} \vee \mathbf{t}$ and the intersection of the line with sphere $\mathbf{t}$ is the pointpair $\boldsymbol{\ell} \vee \mathbf{t}$. The orthogonal projection of the line to sphere $\mathbf{s}$ is the circle $(\boldsymbol{\ell} \cdot \mathbf{s}) \cdot \mathbf{s}$ and the orthogonal projection of point $\mathbf{X}$ to sphere $\mathbf{s}$ is the point pair $(\mathbf{X} \cdot \mathbf{s}) \cdot \mathbf{s}$, where $\mathbf{X}$ is understood as the flat point $\mathbf{X} \wedge e_\infty$. The element $(\mathbf{c} \cdot \mathbf{t}) \cdot \mathbf{t}$ gives the circle that lies on sphere $\mathbf{t}$ and, in the same time, that lies on the sphere which contains circle $\mathbf{c}$ and is perpendicular to sphere $\mathbf{t}$.

Applications of CGA

CGA is the minimal nondegenerate geometric algebra that allows representing Euclidean geometry. We still have the smart quaternionic like representation of rotations and, moreover, we can represent scaling and spheres in CGA. The latter is advantageous especially in robotics - one can model both prismatic and revolute joints but there are also numerous other applications. One of the most active research groups in applications of CGA in robotics and computer vision is the group in Cambridge, see e.g. [9, 10, 11]. Our contribution to this topic is [15, 16, 17, 18, 19, 20, 22, 26] and [21, 23, 25, 27]. The paper [21] is enclosed in Appendix 3; there we show an application of CGA to a specific problem in binocular vision.

3.4 Geometric algebra for conics

Recall that two-dimensional CGA is geometric algebra $\mathbb{G}_{3,1}$ where an Euclidean point is given by (3.16). Hence the objects representable by vectors in $\mathbb{G}_{3,1}$ are linear combinations of $1, x, y, x^2+y^2$, i.e. circles, lines, point pairs and points. If we want to cover also general conics, we need to add two terms: $\frac{1}{2}(x^2-y^2)$ and xy. It turns out that we need two new infinities for that and also their two corresponding counterparts (Witt pairs), [33]. Thus the resulting dimension of the space generating the appropriate geometric algebra is eight.

Let $\mathbb{R}^{5,3}$ denote the eight–dimensional real coordinate space $\mathbb{R}^8$ equipped with a non–degenerate symmetric bilinear form of signature $(5, 3)$. The form defines Clifford algebra $\mathbb{G}_{5,3}$ and this is the Geometric Algebra for Conics in the algebraic sense. To describe an embedding of the Euclidean plane into $\mathbb{R}^{5,3}$, let us choose a basis such that the corresponding bilinear form is

$$B = \begin{pmatrix} 0 & 0 & -1_{3\times3} \\ 0 & 1_{2\times2} & 0 \\ -1_{3\times3} & 0 & 0 \end{pmatrix}, \tag{3.23}$$

where $1_{2\times2}$ and $1_{3\times3}$ denote unit matrices of the displayed size. Analogously to CGA and to the notation in [34], we denote the corresponding basis elements as follows

$$\bar{n}_+, \bar{n}_-, \bar{n}_\times, e_1, e_2, n_+, n_-, n_\times.$$

The form of (3.23) suggests that the basis elements e_1, e_2 will play the usual role of standard basis of the plane while the null vectors $\bar{n}$, n will represent either the origin or the infinity. Note that there are three orthogonal 'origins' $\bar{n}$ and three corresponding orthogonal 'infinities' n. In terms of this basis, a point of the plane $\mathbf{x} \in \mathbb{R}^2$ defined by $\mathbf{x} = xe_1 + ye_2$ is embedded using the operator $J : \mathbb{R}^2 \to \mathcal{C} \subset \mathbb{R}^{5,3}$, which is defined by

$$J(x, y) = \bar{n}_+ + xe_1 + ye_2 + \frac{1}{2}(x^2 + y^2)n_+ + \frac{1}{2}(x^2 - y^2)n_- + xyn_\times. \tag{3.24}$$

Note that $\bar{n}_\times, \bar{n}_-$ are missing and thus the image lies in a six–dimensional subspace of $\mathbb{R}^{5,3}$. Let us remark that the image is the analog of the conformal cone from CGA. It is a two–dimensional real projective variety determined by three homogeneous polynomials of degree two lying in the six–dimensional subspace. Geometric Algebra for Conics (GAC) is the Clifford algebra $\mathbb{G}_{5,3}$ together with the embedding $\mathbb{R}^2 \to \mathbb{R}^{5,3}$ given by (3.24) in the basis determined by matrix (3.23).

The GAC objects which are represented by vectors in the inner product representation are exactly conic sections. Indeed, it is an easy observation that the components of the vector (3.24) representing the embedded Euclidean point form the complete basis of polynomials of degree two and this is exactly the ideal of polynomials defining conic sections as algebraic varieties. Hence each conic has a unique IPNS representation (in the homogeneous sense) given by a vector in basis dual to basis present in (3.24), i.e. each conic in GAC is represented by a vector of the form

$$Q_I = \bar{v}^+ \bar{n}_+ + \bar{v}^- \bar{n}_- + \bar{v}^\times \bar{n}_\times + v^1 e_1 + v^2 e_2 + v^+ n_+. \tag{3.25}$$

The OPNS representation of a conic section is a multivector of degree five and it can be easily computed as the outer product of five GAC points that lie on the conic, i.e.

$$Q_O = P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge P_5, \tag{3.26}$$

where $P_1, \ldots, P_5$ are GAC representatives of the points. The duality between IPNS representation and OPNS representation reads

$$\begin{aligned} Q_O &= (Q_I \wedge n_- \wedge n_\times)^*, \\ Q_I &= (Q_O \wedge \bar{n}_- \wedge \bar{n}_\times)^*, \end{aligned}$$

where the star denotes the usual duality given by the multiplication by the inverse of pseudoscalar. Let us remark that the IPNS representation of intersections of conics, i.e. point quadruplets in general, are represented by the outer product of the two IPNS representations of conics. While the OPNS representation of a conic is easily obtained from the generating points, one can easily read off the internal parameters of the conic from its IPNS representation. For more details see sections 3 and 4 in [24].

Since the GAC embedding (3.24) is the same as CGA embedding up to the last two terms, the scalar product of two embedded points is proportional to the square of their Euclidean distance, see (3.17). In particular, each point is represented by a null vector and GAC is still a conformal model of Euclidean geometry, i.e. the transformations represented by a versor in GAC are conformal transformations. The GAC generators for rotations, translations in the x–direction, translations in the y–direction and for the scaling respectively are given by

$$\begin{aligned} r &= \tfrac{1}{2} e_1 \wedge e_2 + \bar{n}_\times \wedge n_- + n_\times \wedge \bar{n}_-, \\ t_1 &= -\tfrac{1}{2} e_1 \wedge n_+ - \tfrac{1}{2} e_1 \wedge n_- - \tfrac{1}{2} e_2 \wedge n_\times, \\ t_2 &= -\tfrac{1}{2} e_2 \wedge n_+ + \tfrac{1}{2} e_2 \wedge n_- - \tfrac{1}{2} e_1 \wedge n_\times, \\ s &= \tfrac{1}{2} (\bar{n}_+ \wedge n_+ + \bar{n}_- \wedge n_- + \bar{n}_\times \wedge n_\times). \end{aligned}$$

For more details on transformations in GAC see [24] and [28].

Example 3.9. Let us consider five points $[-1,1],[1,1],[2,2],[-1,2],[0,3]$ and their respective GAC representatives $P_1,\dots,P_5$ given by (3.24). These points span a conic with IPNS representation

$$Q_I = (P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge P_5 \wedge \bar{n}_- \wedge \bar{n}_\times)^* = -15\bar{n}_+ + 3\bar{n}_- + 6\bar{n}_\times + 6e_1 - 33e_2 - 18n_+$$

The IPNS representation is directly related to the conic matrix

$$\begin{bmatrix} 6 & -3 & 3 \\ -3 & 9 & -\frac{33}{2} \\ 3 & -\frac{33}{2} & 18 \end{bmatrix}$$

and thus one can see that it is an ellipse and one can easily derive all its internal parameters. Let us further consider a circle of radius 1 with center in point $[0,2]$. Its IPNS representation in GAC reads the same as the representation in algebra CGA, namely

$$C_I = \bar{n}_+ + 2e_2 + \frac{3}{2}n_+.$$

The intersection of circle C and ellipse E is a point quadruplet with IPNS representation $E_I \wedge C_I$. This quadruplet contains two real points $[-1,2],[0,3]$ and two 'infinities'. Indeed, it is easy to see that $J(-1,2) \wedge (E_I \wedge C_I)^* = 0$ and $J(0,3) \wedge (E_I \wedge C_I)^* = 0$.

Applications of GAC

GAC is a geometric algebra made by enlarging 2D CGA such that the geometric objects are not only circles but general conic sections. However the concept of orthogonality is lost by this construction, we can span conics, compute their intersections and conformal transformations similarly as in CGA. The algebra and its abilities are described in detail in our paper [18] that is enclosed in Appendix 4. In [28], we applied GAC to get a new geometric conic fitting algorithm. Other applications can be found in [13, 14].

3.5 Complex geometric algebras

When allowing for complex coefficients in the construction of Clifford algebra in 2.3, the same generators e_A produce by the same formulas the complex Clifford algebra which we denote by $\mathbb{C}_m = \mathrm{Cl}(m,\mathbb{C})$. Clearly, in the complex case no signature is involved, since each basis vector e_j may be multiplied by the imaginary unit i to change the sign of its square. Hence we may assume we start with the real Clifford algebra $\mathbb{G}_m$ with the inner product $e_ie_j = \delta_{ij}$, $1 \leq i,j \leq m$, and we construct the complex Clifford algebra as its complexification $\mathbb{C}_m := \mathbb{G}_m \oplus i\mathbb{G}_m$, i.e. any element $\varphi \in \mathbb{C}_m$ can be written as $\varphi = x + iy$, where $x,y \in \mathbb{G}_m$. The complex Clifford algebras for small m are well known; $\mathbb{C}_0$ are complex numbers itself, $\mathbb{C}_1$ is the algebra of bicomplex numbers and $\mathbb{C}_2$ is the algebra of biquaternions.

The construction via the complexification of $\mathbb{G}_m$ leads to the definition an important anti automorphism of $\mathbb{C}_m$, so-called Hermitian conjugation

$$\varphi^\dagger = (x+iy)^\dagger = \bar{x} - i\bar{y}, \tag{3.27}$$

where the bar notation stands for the Clifford conjugation in $\mathbb{G}_m$. Note that on the zero grade part of the complex Clifford algebra $\mathbb{C}_0 = \mathbb{C}$ it coincides with the usual complex conjugation. The elements satisfying $\varphi^\dagger = \varphi$ and $\varphi^\dagger = -\varphi$ will be called Hermitian and anti-Hermitian respectively. Hermitian conjugation is a very important anti-involution which is the Clifford analogue of the conjugate transpose in matrices. It leads to the definition of the Hermitian inner product on $\mathbb{C}_m$ given by

$$\langle \varphi | \psi \rangle = [\varphi^\dagger \psi]_0, \quad \varphi, \psi \in \mathbb{C}_m \tag{3.28}$$

where we recall that $[\ ]_0$ denotes the projection to the scalar part, i.e. the grade zero part. Indeed, it is easy to see that it is linear in the second slot and conjugate linear in the first slot; for each $z \in \mathbb{C}$ and $\varphi \in \mathbb{C}_m$ we have $(z\varphi)^\dagger = \varphi^\dagger z^\dagger = \bar{z}\varphi^\dagger$ since the Hermitian conjugation is anti-automorphism. The Hermitian symmetry of (3.28) follows from the involutivity of the Hermitian conjugation while its positive definiteness follows from the fact that

$$\langle \varphi | \varphi \rangle = [\varphi^\dagger \varphi]_0 = \sum_A \varphi_A^2,$$

where A is an arbitrary multi index and φ_A is the coefficient at the Grassmann blade e_A, i.e. $\varphi = \sum_A \varphi_A e_A$.

Let us assume a complex Clifford algebra generated by a vector space of an even dimension, i.e. $m = 2n$. Then there exists a special basis formed by null vectors, so called *Witt basis*, see 2.2. Namely, we define

$$\begin{aligned} f_j &= \frac{1}{2}(e_j - ie_{j+n}), \quad j = 1, \dots, n \\ f_j^\dagger &= \frac{1}{2}(e_j + ie_{j+n}), \quad j = 1, \dots, n, \end{aligned}$$

where $e_1, \dots, e_{2n}$ is the usual orthonormal basis. The Witt pairs $(f_j, f_j^\dagger)$ form a basis and the vectors are isotropic with respect to the geometric product, i.e. for each $j = 1, \dots, n$ they satisfy $f_j^2 = 0$ and $f_j^{\dagger 2} = 0$. They also satisfy the Grassmann identities

$$f_j f_k + f_k f_j = f_j^\dagger f_k^\dagger + f_k^\dagger f_j^\dagger = 0, \quad j, k = 1, \dots, n \tag{3.29}$$

and the duality identities

$$f_j f_k^\dagger + f_k^\dagger f_j = \delta_{jk}, \quad j, k = 1, \dots, n \tag{3.30}$$

The Witt basis of the whole complex Clifford algebra $\mathbb{C}_{2n}$ is then obtained, similarly to the basis of the real Clifford algebra, by taking the 2^{2n} possible geometric products of Witt basis vectors, i.e. it is formed by elements

$$(f_1)^{i_1}(f_1^\dagger)^{j_1} \cdots (f_n)^{i_n}(f_n^\dagger)^{j_n}, \quad i_k, j_k \in \{0, 1\} \text{ for } k = 1, \dots, n \tag{3.31}$$

In terms of the Witt basis it is easy to describe directly a realization of a spinor space in $\mathbb{C}_{2n}$. In the language of Clifford algebras, spinor space is defined as a minimal left ideal of the complex Clifford algebra and is realized explicitly by means of a self-adjoint primitive idempotent. We start by defining

$$I_j = f_j f_j^\dagger \text{ and } K_j = f_j^\dagger f_j, \quad j = 1, \dots, n.$$

Direct computations show that both I_j, K_j are mutually commuting self–adjoint idempotents. Moreover, the duality relations (3.30) between Witt basis vectors imply that $I_j + K_j = 1$ for each $j = 1, \dots, n$. Hence we get the resolution of the identity $1 = \prod_{j=1}^n (I_j + K_j)$. Consequently we get

$$\mathbb{C}_{2n} = \mathbb{C}_{2n} \prod_{j=1}^n (I_j + K_j) = \mathbb{C}_{2n} I_1 \cdots I_n \oplus \mathbb{C}_{2n} I_1 \cdots I_{n-1} K_1 \oplus \cdots \oplus \mathbb{C}_{2n} K_1 \cdots K_n,$$

a direct sum decomposition of the complex Clifford algebra into 2^n isomorphic realizations of the spinor space that are denoted according to the specific idempotent involved:

$$\mathbb{S}_{\{i_1 \cdots i_s\}\{k_1 \cdots k_t\}} = \mathbb{C}_{2n} I_{i_1} \cdots I_{i_s} K_{k_1} \cdots K_{k_t} \subset \mathbb{C}_{2n}, \tag{3.32}$$

where $s + t = n$ and the indices are pairwise different. Each such space has dimension 2^n and its basis is obtained by right multiplication of the basis of $\mathbb{C}_{2n}$ by the corresponding primitive idempotent $I_{i_1} \cdots I_{i_s} K_{k_1} \cdots K_{k_t}$. By the basic properties of the Witt basis elements (3.29) and (3.30) it is easy to see that this action is nonzero if and only if the element of $\mathbb{C}_{2n}$ actually lies in the Grassmann algebra generated by n-dimensional space $(f_{i_1}^\dagger, \dots, f_{i_s}^\dagger, f_{k_1}, \dots, f_{k_t})$, i.e. we may write

$$\mathbb{S}_{\{i_1 \cdots i_s\}\{k_1 \cdots k_t\}} = \Lambda(f_{i_1}^\dagger, \dots, f_{i_s}^\dagger, f_{k_1}, \dots, f_{k_t}) I_{i_1} \cdots I_{i_s} K_{k_1} \cdots K_{k_t}. \tag{3.33}$$

It is an easy observation that each such spinor space $\mathbb{S}$ has the structure of a Hilbert space of dimension 2^n due to the Hermitian product (3.28) and that the multiplication in $\mathbb{C}_{2n}$ makes it into a left $\mathbb{C}_{2n}$-module.

Proposition 3.10. *Each spinor space $\mathbb{S} \subset \mathbb{C}_{2n}$ is a Hilbert space of dimension 2^n with Hermitian product defined by* (3.28)*. The unitary group $U(2^n)$ is represented by elements $\lambda \in \mathbb{C}_{2n}$ such that*

$$\lambda^\dagger \lambda = 1 \tag{3.34}$$

Proof. The corresponding unitary group is represented on spinor space $\mathbb{S}$ by elements of the complex Clifford algebra that keep the Hermitian product invariant. For a $\lambda \in \mathbb{C}_{2n}$ and two spinors $\varphi, \psi \in \mathbb{S}$ we compute $\langle \lambda\varphi | \lambda\psi \rangle = [\varphi^\dagger \lambda^\dagger \lambda \psi]_0$ by definition and due to the antiautomorphism property of the Hermitian conjugation. Hence $\langle \lambda\varphi | \lambda\psi \rangle = \langle \varphi | \psi \rangle$ if λ satisfies (3.34). □

The unitary elements of $\mathbb{C}_{2n}$ also satisfy $\lambda\lambda^\dagger = 1$ and form an analogy of unitary matrices. Let us remark that this representation comes from the spin representation of the corresponding complex orthogonal group which is well known form the representation theory.

Applications of complex Clifford algebras

According to Proposition 3.10, states of a n-level quantum system and their evolution can be both described in complex Clifford algebra $\mathbb{C}_{2n}$. It means that we have a realization of the abstract Dirac formalism in a concrete Clifford algebra which is more straightforward in many aspects than the usual realization in matrix algebra. A detailed description of this concept in the theory of quantum computing is given in our paper [32] that is enclosed in Appendix 5.

References

[1] Meinrenken, Eckhard. *Clifford Algebras and Lie Theory.* A Series of Modern Surveys in Mathematics 58, Springer, 2013.

[2] Sharpe, R. W. *Differential geometry.* Graduate Texts in Mathematics 166, Springer, 1997.

[3] Cap, A. and Slovák, J. *Parabolic Geometries: Background and general theory.* Mathematical surveys and monographs, vol. 154, American Math. Soc., 2009

[4] Dorst, L., Fontijne, D., Mann, S. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry.* Morgan Kaufmann Publishers Inc., Burlington, 2007.

[5] Gunn, C.G.: Geometric Algebras for Euclidean Geometry Adv. Appl. Clifford Algebras (2017) 27: 185. https://doi.org/10.1007/s00006-016-0647-0

[6] Gunn, C.G., De Keninck, S.: *3D PGA Cheat Sheet, An extensive reference with 3D PGA formulas*, online: bivector.net

[7] De Keninck, S.: ganja.js, https://zenodo.org/record/3635774 (2020).

[8] De Keninck S.: *Non-parametric Realtime Rendering of Subspace Objects in Arbitrary Geometric Algebras*, (2019) Lecture Notes in Computer Science, pp. 549 - 555, DOI: 10.1007/978-3-030-22514-8_54

[9] Hadfield H., Lasenby J.: *Constrained Dynamics in Conformal and Projective Geometric Algebra*, (2020) Lecture Notes in Computer Science, pp. 459 - 471, DOI: 10.1007/978-3-030-61864-3_39

[10] Hadfield H., Achawal S., Lasenby J., Lasenby A., Young B.: *Exploring Novel Surface Representations via an Experimental Ray-Tracer in CGA*, (2021) Advances in Applied Clifford Algebras, 31 (2), DOI: 10.1007/s00006-021-01117-8

[11] Hadfield H., Wei L., Lasenby J.: *The Forward and Inverse Kinematics of a Delta Robot*, (2020) Lecture Notes in Computer Science, pp. 447 - 458, DOI: 10.1007/978-3-030-61864-3_38

[12] Hildenbrand, D.: Introduction to Geometric Algebra Computing. Boca Raton, USA: Chapman and Hall/CRC (2018).

[13] Byrtus R., Derevianko A., Vašík P., Hildenbrand D., Steinmetz C.: *On Specific Conic Intersections in GAC and Symbolic Calculations in GAALOPWeb*, (2022) Advances in Applied Clifford Algebras, 32 (1), DOI: 10.1007/s00006-021-01182-z

[14] Derevianko A., Loučka P.: *Search for Similarity Transformation Between Image Point Clouds Using Geometric Algebra for Conics*,(2022) Lecture Notes in Computer Science, pp. 215 - 226,DOI: 10.1007/978-3-030-98260-7_13

[15] Návrat, A., Matoušek, R.: *Trident snake control based on conformal geometric algebra*, Advances in Intelligent Systems and Computing 2015.

[16] Hrdina, J., Návrat, A., Vašík, P.: *Control of 3- Link Robotic Snake Based on Conformal Geometric Algebra*, Adv. Appl. Clifford Algebr. **26** (3), 2016, 1069-1080.

[17] Návrat, A., Vašík, P.: *On geometric control models of a robotic snake*, Note di Matematica 2017.

[18] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: *Nilpotent approximation of a trident snake robot controlling distribution*, Kybernetika 2017

[19] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: *Geometric control of the trident snake robot based on CGA,* Adv. Appl. Clifford Algebras (in print), 1–13 online, 2016.

[20] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: *CGA-based robotic snake control*, Adv. Appl. Clifford Algebr. **27**(1), 621–632, 2017.

[21] Hrdina, J., Návrat, A.: *Binocular Computer Vision Based on Conformal Geometric Algebra*, Adv. Appl. Clifford Algebras (2017) 27: 1945. https://doi.org/10.1007/s00006-017-0764-4

[22] Hrdina, J., Návrat, A., Vašík, P.: *Notes on Planar Inverse Kinematics Based on Geometric Algebra*, Adv. Appl. Clifford Algebr. 2018.

[23] Hrdina, J., Matoušek, R., Návrat, A., Vašík, P.: *Geometric algebras for uniform colour spaces*, Mathematical Methods in The Applied Sciences 41 (2018): 4117-4130.

[24] Hrdina, J., Návrat, A., Vašík, P.: *Geometric Algebra for Conics*, Adv. Appl. Clifford Algebras (2018) 28: 66. https://doi.org/10.1007/s00006-018-0879-2

[25] Hrdina, J., Matoušek, R., Návrat, A., Vašík, P.: *Fisheye correction by CGA non-linear transformation*, Mathematical Methods in The Applied Sciences (2018)

[26] Hildenbrand, D., Hrdina, J., Návrat, A., Vašík, P.: *Local Controllability of Snake Robots Based on CRA, Theory and Practice*, Adv. Appl. Clifford Algebras (2019)

[27] Hrdina, J., Návrat, A., Vašík, P.: *GAC Application to Corner Detection Based on Eccentricity*, Lecture Notes in Computer Science (2019)

[28] Hrdina, J., Návrat, A., Vašík, P.: *Conic Fitting in Geometric Algebra Setting*, Adv. Appl. Clifford Algebras 29, 72 (2019). https://doi.org/10.1007/s00006-019-0989-5

[29] Hrdina, J., Návrat, A., Vašík, P. et al.: *Projective Geometric Algebra as a Subalgebra of Conformal Geometric algebra*, Adv. Appl. Clifford Algebras 31, 18 (2021). https://doi.org/10.1007/s00006-021-01118-7

[30] Hrdina, J., Návrat, A., Zalabová, L.: *Symmetries in geometric control theory using Maple*, Mathematics and Computers in Simulation (2021).

[31] Hrdina, J., Návrat, A., Vašík, P. et al.: *Note on geometric algebras and control problems with SO(3)-symmetries*. Math Meth Appl Sci. 2022; 1- 17. doi:10.1002/mma.8662

[32] Hrdina, J., Návrat, A., Vašík, P.: *Quantum Computing Based on Complex Clifford Algebras*, Quantum Information Processing (2022) 21:310, 21:310 doi:10.1007/s11128-022-03648-w

[33] Lounesto, P.: *Clifford Algebra and Spinors*. 2nd edn. CUP, Cambridge, 2006.

[34] Perwass, Ch.: *Geometric Algebra with Applications in Engineering*. Springer Verlag, 2009.

[35] Selig, J. M.: *Geometric Fundamentals of Robotics*, Springer (2004)

[36] A. Agrachev, D. Barilari, U. Boscain, *A comprehensive introduction to sub-Riemannian geometry. From the Hamiltonian viewpoint*, Cambridge Studies in Advanced Mathematics, Vol. 181, Cambridge University Press, 2020 SISSA, 2017.

[37] F. Jean. *Control of Nonholonomic Systems: From Sub–Riemannian Geometry to Motion Planning*. Springer, 2014.

[38] Y. Minami, T. Sugie, M. Ishikawa. *Development and control experiment of the trident snake robot.* IEEE/ASME Trans. Mechatronics, 15(1), 2010.

[39] R. Montgomery. *A Tour of Subriemannian Geometries, Their Geodesics and Applications*. AMS, 2002.

Appendix 1

[31] Jaroslav Hrdina, Aleš Návrat, Petr Vašík, et al.: *Note on geometric algebras and control problems with SO(3)-symmetries.* Math Meth Appl Sci. 2022; 1- 17. doi:10.1002/mma.8662

Received: 22 November 2021 | Revised: 8 July 2022 | Accepted: 9 August 2022
DOI: 10.1002/mma.8662

SPECIAL ISSUE PAPER

WILEY

A note on geometric algebras and control problems with SO(3)-symmetries

Jaroslav Hrdina[1] | **Aleš Návrat**[1] | **Petr Vašík**[1] | **Lenka Zalabová**[2,3]

[1]Institute of Mathematics, Faculty of Mechanical Engineering, Brno University of Technology, Brno, Czech Republic

[2]Department of Mathematics and Statistic, Faculty of Science, Masaryk University, Brno, Czech Republic

[3]Institute of Mathematics, Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic

Correspondence
Jaroslav Hrdina, Institute of Mathematics, Faculty of Mechanical Engineering, Brno University of Technology, Technická 2896/2, 616 69 Brno, Czech Republic.
Email: hrdina@fme.vutbr.cz

Communicated by: W. Sprößig

Funding information
Grantová Agentura České Republiky, Grant/Award Number: 20-11473S; Vysoké Učení Technické v Brně, Grant/Award Number: FSI-S-20-6187

We study the role of symmetries in control systems through the geometric algebra approach. We discuss two specific control problems on Carnot groups of step 2 invariant with respect to the action of $SO(3)$. We understand the geodesics as the curves in suitable geometric algebras which allows us to assess a new algorithm for the local control.

KEYWORDS
Carnot groups, geometric algebras, local control and optimality, sub-Riemannian geodesics, symmetries

MSC CLASSIFICATION
15A67, 53C17

1 | INTRODUCTION

Geometric control theory uses geometric methods to control various mechanical systems.[1,2] We use the methods of sub-Riemannian geometry and Hamiltonian concept.[3,4] As a reasonable starting point, we consider mechanisms moving in the plane, typically wheeled mechanisms like cars (with or without trailers) or robotic snakes.[5,6] The movement of a planar mechanisms is always invariant with respect to the action of the Euclidean group $SE(2)$. As the prototypes of planar mechanisms, we choose those consisting of the body in the shape of a triangle and three legs connected to the vertices of the body by joints of various types and combinations, see the Figure 1. Although such mechanisms have almost the same shape, the configuration spaces may differ. In particular, possible motions of the mechanism induce a specific filtration in the configuration space. We present two examples that carry the filtration $(3, 6)$ and $(4, 7)$, respectively.[6,7]

To control the mechanisms locally, we consider the nilpotent approximations of the original control systems.[8] Although the configuration spaces and their approximations have the same filtration, the approximations form Carnot groups that are generally endowed with more symmetries.[9] One gets the symmetries generated by the right-invariant vector fields, and there may be additional symmetries acting nontrivially on the distribution. Our Carnot groups of filtrations $(3, 6)$ and $(4, 7)$ carry subgroups of the symmetries isomorphic to $SO(3)$.[7,10] This observation leads to the idea of the local control in geometric algebra approach.

We reformulate the control problems in the concept of geometric algebras $\mathbb{G}_3$ and $\mathbb{G}_4$.[11–13] We use the natural $SO(3)$-invariant operations in geometric algebras to reduce the set of geodesics to a simpler set of curves in the geometric algebra.[14] Namely, each geodesic is a linear combination of orthogonal vectors, and $SO(3)$ acts on the geodesics by means of

 © 2022 John Wiley & Sons, Ltd.

the action on the appropriate orthonormal system of vectors. So it is sufficient to study geodesics for one fixed orthonormal basis, that is, we can study just geodesics in the moduli space over the action of $SO(3)$.

We present the local control algorithm for finding geodesics passing through the origin and an arbitrary point in its neighborhood. The algorithm is based on the use of rotors in order to relate two orthogonal bases. We provide an efficient method to such comparison using geometric algebras. We illustrate our algorithm on two specific examples.

2 | NILPOTENT CONTROL PROBLEMS

We focus on two control problems such that their symmetry groups contain $SO(3)$ as their subgroups. The first system has the growth vector $(3, 6)$, and the other one has the growth vector $(4, 7)$.[7,9]

2.1 | Control problems on Carnot groups of step 2

By nilpotent control problems, we mean the invariant control problems on Carnot groups and we consider the Carnot groups G of step 2 with the filtration (m, n).[3,15,16]. If we denote the local coordinates by $(x, z) \in \mathbb{R}^m \oplus \mathbb{R}^{n-m}$, we can model the corresponding Lie algebra $\mathfrak{g}$ of vector fields

$$\begin{aligned} X_i &= \partial_{x_i} - \frac{1}{2}\sum_{l=1}^{n-m}\sum_{j=1}^{m} c_{ij}^l x_j \partial_{z_l} \;\; j = 1, \dots , m \\ X_{m+j} &= \partial_{z_j} \;\; j = 1, \dots , m - n, \end{aligned} \tag{1}$$

where c_{jl}^k are the structure constants of the Lie algebra $\mathfrak{g}$ and the symbol ∂ stands for partial derivative. We discuss the related optimal control problem

$$\dot{q}(t) = u_1 X_1 + \dots + u_m X_m \tag{2}$$

for $t > 0$ and q in G and the control $u = (u_1(t), \dots , u_m(t)) \in \mathbb{R}^m$ with the boundary condition $q(0) = q_1$, $q(T) = q_2$ for fixed points $q_1, q_2 \in G$, where we minimize the cost functional $\frac{1}{2}\int_0^T (u_1^2 + \dots + u_m^2)\mathrm{d}t$. The solutions $q(t)$ then correspond to the sub-Riemannian geodesics, that is, admissible curves parametrized by a constant speed whose sufficiently small arcs are the length minimizers.

We use the Hamiltonian approach to this control problem.[3] There are no strict abnormal extremals for the step 2 Carnot groups, so we focus on the normal geodesics and address them just as geodesics. The left-invariant vector fields X_i, $i = 1, \dots , m$ form a basis of TG and determine the left-invariant coordinates on G. We define the corresponding left-invariant coordinates $h_i, i = 1, \dots , m$ and $w_i, i = 1, \dots , n-m$ on the fibers of T^*G by $h_i(\lambda) = \lambda(X_i)$ and $w_i(\lambda) = \lambda(X_{m+i})$, for arbitrary 1-forms λ on G. Thus, we use (x_i, w_i) as the global coordinates on T^*G.

The geodesics are exactly the projections of normal Pontryagin extremals, that is, the integral curves of the left-invariant normal Hamiltonian

$$H = \frac{1}{2}(h_1^2 + h_2^2 + \dots + h_m^2), \tag{3}$$

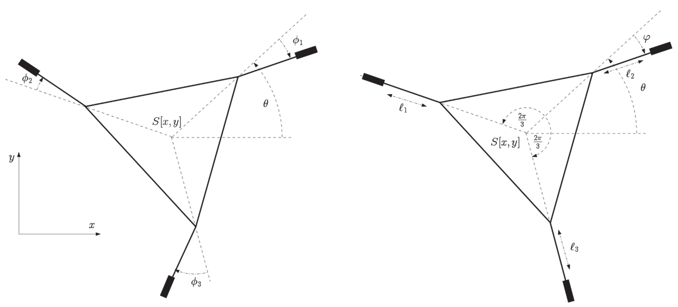


FIGURE 1 Generalized trident snakes

on G. Assume that $\lambda(t) = (x_i(t), z_i(t), h_i(t), w_i(t))$ in T^*G is a normal extremal. Then the controls u_j to system (2) satisfy $u_j(t) = h_j(\lambda(t))$ and the base system takes the form of

$$\begin{aligned} \dot{x}_i &= h_i, \ i = 1, \dots, m \\ \dot{z}_j &= -\frac{1}{2}\sum_{i=1}^{m} c_{ik}^j h_i x_k, \ j = 1, \dots, n-m \end{aligned} \tag{4}$$

for $q = (x_i, z_i)$. Using $u_j(t) = h_j(\lambda(t))$ and the equation $\dot{\lambda}(t) = \vec{H}(\lambda(t))$ for the normal extremals, we write the fiber system as

$$\begin{aligned} \dot{h}_i &= -\sum_{l=1}^{m-n}\sum_{j=1}^{m} c_{ij}^l h_j w_l, \ i = 1, \dots, m, \\ \dot{w}_j &= 0, \quad j = 1, \dots, n-m, \end{aligned} \tag{5}$$

where c_{ij}^l are the structure constants of the Lie algebra $\mathfrak{g}$ for the basis X_i. The solutions w_i, $i = 1, \dots, n-m$ are constants that we denote by

$$w_1 = K_1, \dots, w_{n-m} = K_{n-m}. \tag{6}$$

If $K_1 = \dots = K_{n-m} = 0$, then $h(t) = h(0)$ is a constant and the geodesic $(x_i(t), z_i(t))$ is a line in G such that $z_i(t) = 0$. If at least one of K_i is nonzero, the first part of the fiber system (5) forms a homogeneous system of ODEs $\dot{h} = -\Omega h$ with constant coefficients for $h = (h_1, \dots, h_m)^T$ and the system matrix Ω. Its solution is given by $h(t) = e^{-t\Omega}h(0)$, where $h(0)$ is the initial value of the vector h at the origin.

2.2 | Left-invariant control problem with the growth vector (3, 6)

Let us consider three vector fields on $\mathbb{R}^6$ with the local coordinates $(x_1, x_2, x_3, z_1, z_2, z_3)$ in the form

$$\begin{aligned} X_1 &= \partial_{x_1} + \frac{x_3}{2}\partial_{z_2} - \frac{x_2}{2}\partial_{z_3}, \\ X_2 &= \partial_{x_2} + \frac{x_1}{2}\partial_{z_3} - \frac{x_3}{2}\partial_{z_1}, \\ X_3 &= \partial_{x_3} + \frac{x_2}{2}\partial_{z_1} - \frac{x_1}{2}\partial_{z_2}. \end{aligned} \tag{7}$$

The only nontrivial Lie brackets are

$$X_4 = [X_1, X_2] = \partial_{z_3}, \ X_5 = [X_1, X_3] = -\partial_{z_2}, \ X_6 = [X_2, X_3] = \partial_{z_1}. \tag{8}$$

These six vector fields determine a step 2 nilpotent Lie algebra $\mathfrak{m}$ with the multiplication table given by Table 1.

There is a Carnot group M such that the fields X_i, $i = 1 \dots, 6$ are left-invariant for the corresponding group structure. When identified with $\mathbb{R}^6 = \mathbb{R}^3 \oplus \mathbb{R}^3$, the group structure on M reads as

$$(x, z) \cdot (x', z') = \left(x + x', z + z' + \frac{1}{2}x \times x'\right) \tag{9}$$

TABLE 1 Lie algebra $\mathfrak{m}$

$\mathfrak{m}$	X_1	X_2	X_3	X_4	X_5	X_6
X_1	0	X_4	X_5	0	0	0
X_2	$-X_4$	0	X_6	0	0	0
X_3	$-X_5$	$-X_6$	0	0	0	0
X_4	0	0	0	0	0	0
X_5	0	0	0	0	0	0
X_6	0	0	0	0	0	0

for $x = (x_1, x_2, x_3)$ and $z = (z_1, z_2, z_3)$, where $\times$ stands for the vector product on $\mathbb{R}^3$. In particular, $\mathcal{M} = \langle X_1, X_2, X_3 \rangle$ forms a three-dimensional left-invariant distribution on M. We define the left-invariant sub-Riemannian metric g_M on $\mathcal{M}$ by declaring X_1, X_2, X_3 orthonormal.

The geodesics of the control problem are the solutions to control systems (4),(5), with $(m, n) = (3, 6)$, and one can read the structure constants in Table 1. Hence, the fiber system is given by $w_1 = K_1, w_2 = K_2, w_3 = K_3$ for the constants K_1, K_2, K_3 and $\dot{h} = -\Omega h$ for $h = (h_1, h_2, h_3)^T$ and

$$\Omega = \begin{pmatrix} 0 & K_1 & K_2 \\ -K_1 & 0 & K_3 \\ -K_2 & -K_3 & 0 \end{pmatrix}. \tag{10}$$

Its solution is given by the exponential $h(t) = e^{-t\Omega}h(0)$, where $h(0)$ is the initial value of the vector h at the origin. We write an explicit formula for the general solution in terms of eigenvectors of (10). If at least one of the constants K_i is nonzero, the kernel of Ω, that is, zero-eigenspace, is one-dimensional, generated by the vector $(K_3, K_2, K_1)^T$. Its orthogonal complement corresponds to the sum of eigenspaces to the eigenvalues $\pm iK$, where $K := \sqrt{K_1^2 + K_2^2 + K_3^2}$ and is generated by the vectors $(-K_1K_3, -K_1K_2, K_2^2 + K_3^2) \pm i(K_2, -K_3, 0)$. Thus, solution to the fiber system can be written as follows:

$$h(t) = (C_1 \cos(Kt) - C_2 \sin(Kt))v_1 + (C_1 \sin(Kt) + C_2 \cos(Kt))v_2 + C_3 v_3, \tag{11}$$

where v_1, v_2, v_3 is the eigenspace-adapted real orthonormal basis

$$v_1 = \frac{1}{K\sqrt{K_2^2 + K_3^2}} \begin{pmatrix} -K_1K_3 \\ K_1K_2 \\ K_2^2 + K_3^2 \end{pmatrix}, v_2 = \frac{1}{\sqrt{K_2^2 + K_3^2}} \begin{pmatrix} -K_2 \\ -K_3 \\ 0 \end{pmatrix}, v_3 = \frac{1}{K} \begin{pmatrix} K_3 \\ -K_2 \\ K_1 \end{pmatrix}$$

and C_1, C_2, C_3 are the constants that satisfy the level set condition $H = 1/2$, that is, $\|h(t)\| = 1$ that reads $C_1^2 + C_2^2 + C_3^2 = 1$. Let us note that the choice $C_1 = C_2 = 0$ leads to the constant solutions that are irrelevant as the control functions. Thus, we assume that at least one of the constants C_1, C_2 is nonzero.

Let us emphasize that the base system (4) can be written in terms of a vector product as follows:

$$\begin{aligned} \dot{x} &= h, \\ \dot{z} &= \frac{1}{2} x \times h \end{aligned} \tag{12}$$

for vectors $x = (x_1, x_2, x_3)^T$ and $z = (z_1, z_2, z_3)^T$. One obtains the general solution by substituting (11) for h and by consequent direct integration. We are interested in the solutions passing through the origin, that is, we impose the initial condition

$$x_i(0) = 0, z_i(0) = 0, \ i = 1, 2, 3. \tag{13}$$

However, it may be difficult to find the integration constants giving the geodesics through a fixed target point.

2.3 | Left-invariant control problem with the growth vector (4, 7)

Let us consider four vector fields on $\mathbb{R}^7$ with the local coordinates $(x, \ell_1, \ell_2, \ell_3, y_1, y_2, y_3)$ in the form

$$\begin{aligned} Y_0 &= \partial_x - \frac{\ell_1}{2}\partial_{y_1} - \frac{\ell_2}{2}\partial_{y_2} - \frac{\ell_3}{2}\partial_{y_3}, \\ Y_1 &= \partial_{\ell_1} + \frac{x}{2}\partial_{y_1}, \ Y_2 = \partial_{\ell_2} + \frac{x}{2}\partial_{y_2}, \ Y_3 = \partial_{\ell_3} + \frac{x}{2}\partial_{y_3}. \end{aligned} \tag{14}$$

The only nontrivial Lie brackets are as follows:

$$Y_4 = [Y_0, Y_1] = \partial_{y_1}, \ Y_5 = [Y_0, Y_2] = \partial_{y_2}, \ Y_6 = [Y_0, Y_3] = \partial_{y_3}. \tag{15}$$

TABLE 2 Lie algebra $\mathfrak{n}$

$\mathfrak{n}$	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
Y_0	0	Y_4	Y_5	Y_6	0	0	0
Y_1	$-Y_4$	0	0	0	0	0	0
Y_2	$-Y_5$	0	0	0	0	0	0
Y_3	$-Y_6$	0	0	0	0	0	0
Y_4	0	0	0	0	0	0	0
Y_5	0	0	0	0	0	0	0
Y_6	0	0	0	0	0	0	0

These seven fields determine a step 2 nilpotent Lie algebra $\mathfrak{n}$ with the multiplication table given by Table 2.

There is a Carnot group N such that the fields Y_i, $i = 1, \dots, 7$ are left-invariant for the corresponding group structure. The group structure on N, when identified with $\mathbb{R}^7 = \mathbb{R} \oplus \mathbb{R}^3 \oplus \mathbb{R}^3$, yields

$$(x, \ell, y) \cdot (x', \ell', y') = \left(x + x', \ell + \ell', y + y' + \frac{1}{2}\ell \times \ell'\right) \tag{16}$$

for $\ell = (\ell_1, \ell_2, \ell_3)$ and $y = (y_1, y_2, y_3)$. In particular, $\mathcal{N} = \langle Y_0, Y_1, Y_2, Y_3\rangle$ forms a four-dimensional left-invariant distribution on N. Moreover, there is a natural decomposition

$$\mathcal{N} = \langle Y_0\rangle \oplus \langle Y_1, Y_2, Y_3\rangle \tag{17}$$

into a one-dimensional distribution and a three-dimensional involutive distribution, both left-invariant. We define the left-invariant sub-Riemannian metric g_N on $\mathcal{N}$ by declaring Y_0, Y_1, Y_2, Y_3 orthonormal.

The geodesics of the control problem are solutions to the control systems (4),(5), with $(m, n) = (4, 7)$, and we read the structure constants in Table 2. Hence, the first part of the fiber system (5) is given by $w_1 = K_1, w_2 = K_2, w_3 = K_3$, where K_1, K_2, K_3 are constants. The second part of the fiber system takes the form $\dot{h} = -\Omega h$, where $h := (h_0, h_1, h_2, h_3)^T$ and

$$\Omega = \begin{pmatrix} 0 & K_1 & K_2 & K_3 \\ -K_1 & 0 & 0 & 0 \\ -K_2 & 0 & 0 & 0 \\ -K_3 & 0 & 0 & 0 \end{pmatrix}. \tag{18}$$

Its solution is given by $h(t) = e^{-t\Omega}h(0)$, where $h(0)$ is the initial value of the vector h at the origin, and we write its explicit form in terms of the eigenvectors of (18). If $K_1 = K_2 = K_3 = 0$, then $h(t) = h(0)$ is a constant and the geodesic $(x(t), \ell_i(t), y_i(t))$ is a line in N such that $y_i = 0$. If at least one of the constants K_i is nonzero, the kernel of Ω, that is, zero-eigenspace, is two-dimensional and is generated by the vectors $(0, -K_3, 0, K_1)^T$ and $(0, -K_2, K_1, 0)^T$. Its orthogonal complement corresponds to the sum of the eigenspaces to the eigenvalues $\pm iK$, where $K := \sqrt{K_1^2 + K_2^2 + K_3^2}$ and is generated by the eigenvectors $(0, K_1, K_2, K_3)^T \pm i(K, 0, 0, 0)^T$. Thus, the solution to the vertical system for nonzero K takes the form

$$\begin{aligned} h_0 &= K(C_2 \cos(Kt) - C_1 \sin(Kt)) \\ \bar{h} &= K(C_2 \sin(Kt) + C_1 \cos(Kt))r_1 + Cr_2 \end{aligned} \tag{19}$$

where $\bar{h} = (h_1, h_2, h_3)^T$ and r_1, r_2 are eigenspace-adapted real orthonormal vectors

$$r_1 = \frac{1}{K}\begin{pmatrix} K_1 \\ K_2 \\ K_3 \end{pmatrix}, \quad r_2 = \frac{1}{C}\left(C_3 \begin{pmatrix} -K_3 \\ 0 \\ K_1 \end{pmatrix} + C_4 \begin{pmatrix} -K_2 \\ K_1 \\ 0 \end{pmatrix}\right)$$

with the constants C_1, C_2, C_3, C_4 and the normalization factor $C = \sqrt{(C_3K_3 + C_4K_2)^2 + K_1^2(C_3^2 + C_4^2)}$. The level set condition $\|h(t)\| = 1$ reads $C_1^2 + C_2^2 + C_3^2 = 1$. Let us note that the choice $C_1 = C_2 = 0$ leads to the constant solutions that are irrelevant as the control functions. Thus, we assume that at least one of the constants C_1, C_2 is nonzero.

The base system (4) takes the explicit form of

$$\begin{aligned} \dot{x} &= h_0, \\ \dot{\ell} &= \bar{h}, \\ \dot{y} &= \frac{1}{2}(x\bar{h} - h_0\ell). \end{aligned} \tag{20}$$

We are interested in the solutions passing through the origin; that is, we impose the initial condition

$$x(0) = 0, \ell_i(0) = 0, y_i(0) = 0, \ i = 1, 2, 3. \tag{21}$$

By substitution of (19), system (20) can be directly integrated. Again, it may be difficult to find the geodesics through a fixed target point. In Section 4, we show how the symmetries of the system and the geometric algebra approach are used for finding a geodesic towards a given point.

2.4 | Symmetries of the control systems

Symmetries of the control system in question coincide with the symmetries of the corresponding left–invariant sub–Riemannian structure $(M, \mathcal{M}, g_M)$ and $(N, \mathcal{N}, g_N)$, respectively. These are precisely the automorphisms on groups preserving the distributions and sub–Riemannian metrics. The group $SO(3)$ acts on $\mathbb{R}^3$ and preserves the vector product which implies the following statement.

Proposition 1. *For each $R \in SO(3)$, the map*

$$(x, z) \mapsto (Rx, Rz) \tag{22}$$

maps the geodesics of the system from Section 2.2 starting at the origin to the geodesics starting at the origin. For each $R \in SO(3)$, the map

$$(x, \ell, y) \mapsto (x, R\ell, Ry) \tag{23}$$

maps the geodesics of the system from Section 2.3 starting at the origin to the geodesics starting at the origin.

Proof. Follows from the invariance of (12) and (20) with respect to the action of $R \in SO(3)$. □

3 | GEOMETRIC ALGEBRA

The construction of the universal real geometric algebra is well-known.[11–13,17] We provide only a brief description in a special case $\mathbb{G}_m$ that we use later. In general, geometric algebras are based on symmetric bilinear forms of arbitrary signature. Here, we deal with the real vector space $\mathbb{R}^m$ endowed with a positive definite symmetric bilinear form B only.

3.1 | Geometric product

Let us consider a positive definite symmetric bilinear form B on $\mathbb{R}^m$ and the associated orthonormal basis $(e_1, \dots, e_m)$, that is,

$$B(e_i, e_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \text{ where } 1 \leq i, j \leq m.$$

The Grassmann algebra $\Lambda(\mathbb{R}^m)$ is an associative algebra with the anti-symmetric outer product $\wedge$ defined by the rule

$$e_i \wedge e_j + e_j \wedge e_i = 0 \text{ for } 1 \leq i, j \leq m.$$

The Grassmann blade of grade r is $e_A = e_{i_1} \wedge \dots \wedge e_{i_r}$, where the multi-index A is a set of indices ordered in the natural way $1 \leq i_1 \leq \dots \leq i_r \leq m$, and we put $e_\emptyset = 1$. Blades of grades $0 \leq r \leq m$ form the basis of the Grassmann algebra $\Lambda(\mathbb{R}^m)$ and for the outer product we have

$$e_j \wedge e_A = \begin{cases} e_j \wedge e_{i_1} \wedge \cdots \wedge e_{i_r} & \text{if } j \notin A, \\ 0 & \text{if } j \in A \end{cases}$$

and $1 \wedge e_A = e_A$. For the vectors from $\mathbb{R}^m$, the inner product $e_i \cdot e_j = B(e_i, e_j)$ and the outer product $e_i \wedge e_j$ lead to the so-called geometric product

$$e_i e_j = e_i \cdot e_j + e_i \wedge e_j, \quad 1 \leq i, j \leq m.$$

The definitions of inner and geometric products then extend to blades of the grade r as follows. For the inner product we put $1 \cdot e_A = 0$ and

$$e_j \cdot e_A = e_j \cdot (e_{i_1} \wedge \cdots \wedge e_{i_r}) = \sum_{k=1}^{r} (-1)^k B(e_j, e_{i_k}) e_{A \setminus \{i_k\}},$$

where $e_{A\setminus\{i_k\}}$ is the blade of grade $r-1$ created by deleting e_{i_k} from e_A. This product is also called the left contraction in literature. For the geometric product, we define

$$e_j e_A = e_j \cdot e_A + e_j \wedge e_A.$$

These definitions extend linearly to the whole vector space $\Lambda(\mathbb{R}^m)$. Thus, we get an associative algebra over this vector space, the so-called real geometric algebra, denoted by $\mathbb{G}_m$. Note that this algebra is naturally graded; the grade zero and grade one elements are identified with $\mathbb{R}$ and $\mathbb{R}^m$, respectively. Finally, we can define the norm of a blade as the magnitude of the blade $|e_A| = \sqrt{e_A \cdot \tilde{e}_A}$. Note that $e_A \cdot \tilde{e}_A$, where $e_A \neq 0$ is always positive in $\mathbb{G}_m$.

3.2 | Objects

The vectors in $\mathbb{R}^m$ with the coordinates $(x_1, \dots, x_m)$ are given by $x = x_1 e_1 + \cdots + x_m e_m$, and the square with respect to the geometric product $x^2 = x_1^2 + \cdots + x_m^2 \in \mathbb{R}$ coincides with the square of the Euclidean norm of x. A vector x represents a one-dimensional subspace (line) p in $\mathbb{R}^m$ given by the scalar multiples of x which in $\mathbb{G}_m$ is expressed by the formula $u \in p \iff u \wedge x = 0$. In the same way, a plane π generated by two vectors x and y is represented by $x \wedge y$ in the sense $u \in \pi \iff u \wedge x \wedge y = 0$. In general, any r-dimensional subspace $V_r \subseteq \mathbb{R}^m$ is represented by a blade A_r of grade r such that

$$V_r = NO(A_r) = \{x \in \mathbb{R}^m : x \wedge A_r = 0\}. \tag{24}$$

Such a representation is called the outer product null space (OPNS) representation in the literature. In particular, the whole space $\mathbb{R}^m$ is represented by a blade of maximal grade, so-called pseudoscalar. Similarly, one defines the inner product null space (IPNS) representation A^*_{m-r} of V_r as a blade of grade $m-r$ such that $x \in V_r \iff x \cdot A^*_{m-r} = 0$. The OPNS and IPNS representations are mutually dual with respect to the duality on $\mathbb{G}_m$ defined by the multiplication by pseudoscalar, namely,

$$A^* = AI,$$

where A is a blade and I is the pseudoscalar. Indeed, one can show that $(x \wedge A)I = x \cdot (AI)$ for each vector $x \in \mathbb{R}^m$, in particular

$$x \wedge A = 0 \iff x \cdot A^* = 0.$$

Remark 1. OPNS representations of objects in $\mathbb{G}_3$ are summarized in Table 3. For example, the plane generated by the vectors u, v has the OPNS representation $u \wedge v$. Its IPNS representation $(u \wedge v)^*$ is a vector perpendicular to the plane. More specifically, for the pseudoscalar $I = e_1 \wedge e_2 \wedge e_3 = e_1 e_2 e_3$, we receive the usual vector product in geometric algebra form as

$$u \times v = -(u \wedge v)I. \tag{25}$$

TABLE 3 Blades of geometric algebra $\mathbb{G}_3$

Grade	Name	Blades	Dimension	Objects
0	Scalars	1	1	Numbers
1	Vectors	$e_1,\ e_2,\ e_3$	3	Lines
2	Bivectors	$e_1 \wedge e_2,\ e_1 \wedge e_3,\ e_2 \wedge e_3$	3	Planes
3	Pseudoscalars	$e_1 \wedge e_2 \wedge e_3$	1	Volume forms

3.3 | Transformations

Let us fix a vector $n \in \wedge^1\mathbb{R}^m \subset \mathbb{G}_m$ such that $n \cdot n = n^2 = 1$ and $x \in \wedge^1\mathbb{R}^m \subset \mathbb{G}_m$ arbitrary. The negative conjugation $-nxn$ defines the reflection with respect to the hyperplane orthogonal to n, because

$$\begin{aligned} -nx^{\perp}n &= -(n \wedge x^{\perp})n = (x^{\perp} \wedge n)n = x^{\perp}nn = x^{\perp}, \\ -nx^{||}n &= -x^{||}, \end{aligned}$$

where $x = x^{||} + x^{\perp}$ is the orthogonal decomposition of x with respect to n. Conjugation preserves the grades of blades and is an outermorphism $n(u_1 \wedge \cdots \wedge u_l)n = (nu_1n) \wedge \cdots \wedge (nu_ln)$ for any vectors $u_1, \ \dots\ , u_l$; thus, minus the conjugation is the (anti)outermorphism depending on the dimension m. Since each rotation is a composition of two reflections, a rotation in $\mathbb{G}_m$ is represented by the conjugation with respect to the geometric product of two vectors. To find a rotor between vectors x and y, we have a nice formula at hand.

Lemma 1. *Let x and y be the unit vectors in $\mathbb{G}_m$, that is, $x, y \in \wedge^1\mathbb{G}_m$, then the formula*

$$R_{xy} = \widehat{1 + yx}, \tag{26}$$

where the hat symbol stands for the normalization $\hat{u} = u/\sqrt{u \cdot u}$, defines the rotation in the plane $x \wedge y$ which maps vector x to y and acts trivially on $(x \wedge y)^*$.

Proof. Multiplication of two vectors $x, y \in \wedge^1\mathbb{R}^m \subset \mathbb{G}_m$, such that $x^2 = y^2 = 1$, defines a multivector

$$yx = \cos(\theta) + \sin(\theta)\widehat{y \wedge x},$$

where θ is the angle between x and y, Lemma 4.2[12]. The conjugation by such multivector yx represents the rotation in the plane $x \wedge y$ with respect to angle 2θ in the positive way. Using standard trigonometric formulas, we can see by straightforward calculation that

$$\begin{aligned} R_{xy} = \widehat{1 + yx} &= \frac{1 + \cos(\theta) + \widehat{y \wedge x}\sin(\theta)}{\sqrt{(1 + \cos(\theta))^2 + \sin^2(\theta)}} = \frac{1 + \cos(\theta) + \widehat{y \wedge x}\sin(\theta)}{\sqrt{2 + 2\cos(\theta)}} \\ &= \sqrt{\frac{1 + \cos(\theta)}{2}} + \widehat{y \wedge x}\sqrt{\frac{1 - \cos^2(\theta)}{2(1 + \cos(\theta))}} = \sqrt{\frac{1 + \cos(\theta)}{2}} + \widehat{y \wedge x}\sqrt{\frac{1 - \cos(\theta)}{2}} \\ &= \cos\left(\frac{\theta}{2}\right) + (y \wedge x)\sin\left(\frac{\theta}{2}\right). \end{aligned}$$

So R_{xy} is the rotation in the plane $\widehat{x \wedge y}$ in the positive way about the angle between x and y so the vector x goes to the vector y.

Finally, $(x \wedge y)^*$ is orthogonal to x and y and the straightforward computation $xy(x \wedge y)^*yx = xyyx(x \wedge y)^* = (x \wedge y)^*$ proves the rest of the statement. □

Remark 2. One can see that yxy is an axial symmetry with respect to y, and thus, $x + yxy = 2\cos(\theta)y$. We can compute the square of the norm

$$(1 + yx)(1 + xy) = 1 + xy + yx + 1 = 2 + 2\cos(\theta)$$

and with the help of the geometric product, we compute

$$(1 + yx)x(1 + xy) = (x + y)(1 + xy) = (x + 2y + yxy) = (2 + 2\cos(\theta))y$$

so the conjugation by (26) maps x to y.

3.4 | Rotor construction

Let $(x_1, \dots, x_m)$ and $(y_1, \dots, y_m)$ be a pair of bases of $\mathbb{R}^m$ such that

1. $x_i \cdot x_j = y_i \cdot y_j$ for all $i, j = 1, \dots, m$, that is, all the scalar products are equal, and
2. $x_1 \wedge \cdots \wedge x_m = y_1 \wedge \cdots \wedge y_m$, that is, the pseudoscalars are equal.

Let us remind that a complete flag $\{V\}$ in an increasing sequence of subspaces of the vector space $\mathbb{R}^m$

$$\{0\} \subset V_1 \subset V_2 \subset \cdots \subset V_m = \mathbb{R}^m,$$

such that $\dim(V_i) = i$. We use the complete flags to find the explicit rotation R such that $Rx_i\tilde{R} = y_i$ for all $i = 1, \dots m$. Our method can be summarized as follows:

- We consider the complete flags $\{V\}$ and $\{W\}$ by setting $V_i = \langle x_1, \dots, x_i\rangle = NO(x_1 \wedge \cdots \wedge x_i)$ and $W_i = \langle y_1, \dots, y_i\rangle = NO(y_1 \wedge \cdots \wedge y_i)$, respectively.
- We map the complete flag $\{V\}$ to the complete flag $\{W\}$ inductively in m steps. In the j^{th} step, we assume $V_i = W_i$ for $i > j$ and we find the rotation R_i such that $R_iV_i\tilde{R}_i = W_i$ for $i > j - 1$.

Before we formulate the construction in detail, we need several technical lemmas.

Lemma 2. *Let $(x_1, \dots, x_m)$ and $(y_1, \dots, y_m)$ be a pair of bases such that*

1. $x_i \cdot x_j = y_i \cdot y_j$ *for all* $i, j = 1, \dots, m$, *and*
2. $x_1 \wedge \cdots \wedge x_i = y_1 \wedge \dots \wedge y_i$ *for all* $i = 1, \dots, m$.

If $\{V\}$ and $\{W\}$ are the corresponding complete flags, respectively, and if $V_i = W_i$ for all $i = 1, \dots, m$, then $x_i = y_i$ for all $i = 1, \dots, m$.

Proof. The equality $x_1 = y_1$ holds trivially from the assumptions. Then $x_2 \cdot x_2 = y_2 \cdot y_2$ reads that x_2, y_2 are of the same length, $x_2 \cdot x_1 = y_2 \cdot y_1$ reads that the angles between x_1, x_2 and y_1, y_2 are identical and $x_1 \wedge x_2 = y_1 \wedge y_2$ reads that they have the same orientation. Then $V_2 = W_2$ and $x_1 = y_1$ imply $x_2 = y_2$ and so on for all the basis vectors. □

Lemma 3. *Let $(x_1, \dots, x_i, z)$ and $(y_1, \dots, y_i, z)$ be a pair of sets of independent vectors such that $x_1 \wedge \cdots \wedge x_i \wedge z = y_1 \wedge \cdots \wedge y_i \wedge z$. If $NO(x_1 \wedge \cdots \wedge x_i) = NO(x_1 \wedge \cdots \wedge x_i)$ then $x_1 \wedge \cdots \wedge x_i = y_1 \wedge \cdots \wedge y_i$.*

Proof. The independence of the sets of vectors implies $x_1 \wedge \cdots \wedge x_i \wedge z \neq 0$, $y_1 \wedge \cdots \wedge y_i \wedge z \neq 0$. If $NO(x_1 \wedge \cdots \wedge x_i) = NO(y_1 \wedge \cdots \wedge y_i)$, then $x_1 \wedge \cdots \wedge x_i = \beta x_1 \wedge \cdots \wedge x_i$ for $\beta \in \mathbb{R}$.
Then

$$\begin{aligned} x_1 \wedge \cdots \wedge x_i \wedge z &= y_1 \wedge \cdots \wedge y_i \wedge z, \\ (x_1 \wedge \cdots \wedge x_i - y_1 \wedge \cdots \wedge y_i) \wedge z &= 0, \\ (1 - \beta)x_1 \wedge \cdots \wedge x_i \wedge z &= 0. \end{aligned}$$

Finally $\beta = 1$ and $x_1 \wedge \cdots \wedge x_i = y_1 \wedge \cdots \wedge y_i$. □

Lemma 4. *Consider two complete flags $\{V\}$ and $\{W\}$ in $\mathbb{R}^m$ and $i \leq m$ such that $V_j = W_j$ for $j > i$. The rotor R_i between the hyperplanes $V_i \oplus V_{i+1}^{\perp}$ and $W_i \oplus W_{i+1}^{\perp}$ constructed by the formula (26) maps V_i to W_i.*

Proof. The property $V_i \subset V_{i+1}$ implies that $V_{i+1}^{\perp} \subset V_i^{\perp}$ and thus $V_i \oplus V_{i+1}^{\perp}$ is a hyperplane equipped with an orthogonal decomposition. Recall that $V_{i+1} = W_{i+1}$, and thus, $V_{i+1}^{\perp} = W_{i+1}^{\perp}$. Any rotation preserves an orthogonal decomposition, and thus, R_i acts as the identity on $V_{i+1}^{\perp} = W_{i+1}^{\perp}$, so it maps V_i to W_i. □

We use all these lemmas to provide a constructive proof of the following theorem.

Theorem 3.5. *Let $(x_1, \dots, x_m)$ and $(y_1, \dots, y_m)$ be a pair of bases of $\mathbb{R}^m$ such that*

1. $x_i \cdot x_j = y_i \cdot y_j$ *for all* $i, j = 1, \dots, m$, *and*
2. $x_1 \wedge \cdots \wedge x_m = y_1 \wedge \cdots \wedge y_m$.

Then we can construct a rotor R such that $Rx_i\tilde{R} = y_i$ for all $j, i = 1, \dots m$.

Proof. Let $\{V\}$ and $\{W\}$ be a pair of the corresponding complete flags $V_i = NO(x_1 \wedge \dots \wedge x_i)$ and $W_i = NO(y_1 \wedge \cdots \wedge y_i)$. We construct a rotor $R = R_1 \dots R_m$ mapping the complete flag $\{V\}$ to the complete flag $\{W\}$ so that $V_i = W_i$ for all $i = 1, \dots m$. The result on bases x_i, y_i then follows by Lemmas 2 and 3.

We define R_m as the identity and proceed inductively. It follows from Lemma 1 that there is a rotation R_{m-1} between the hyperplanes $V_{m-1} \oplus V_m^{\perp} \cong V_{m-1}$ and $W_{m-1} \oplus W_m^{\perp} \cong W_{m-1}$ which maps the complete flag $\{V\}$ to the complete flag $\{RV\tilde{R}\}$ in such a way that $W_{m-1} = RV_{m-1}\tilde{R}$, where $R = R_{m-1}R_m$.

As the induction step, we consider the rotor $R = R_j \cdots R_m$ such that $RV_i\tilde{R} = W_i$ for all indices $i \geq j$. According to Lemma 4, the rotation R_{j-1} between the hyperplanes $(RV_{j-1}\tilde{R}) \oplus (RV_j\tilde{R})^{\perp}$ and $W_{j-1} \oplus W_j^{\perp}$ maps the complete flag $\{RV\tilde{R}\}$ to the complete flag $\{R_jRV\tilde{R}\tilde{R}_j\}$ in such a way that $W_i = R_jRV_i\tilde{R}\tilde{R}_j$ for all $i \geq j-1$.

After m steps, the rotor $R = R_1 \dots R_m$ maps the complete flag $\{V\}$ to the complete flag $\{W\}$ in such a way that $V_i = W_i$ for all $i = 1, \dots, m$ and so $Rx_i\tilde{R} = y_i$ for all $j, i = 1, \dots, m$ because of Lemma 2. □

The explicit construction in the proof of Theorem 3.5 gives us the following algorithm.

Calculate the rotor $R = R_1 \dots R_m$

Require: $x_i \cdot x_j = y_i \cdot y_j$ and $x_1 \wedge \cdots \wedge x_m = y_1 \wedge \cdots \wedge y_m$
Ensure: $y_i = Rx_i\tilde{R}$
  for $m > i > 0$ **do**
    $V_i \leftarrow x_1 \wedge \cdots \wedge x_i$
    $W_i \leftarrow y_1 \wedge \cdots \wedge y_i$
  end for
  $R \leftarrow Id$
  for $m > i > 0$ **do**
    $V_i \leftarrow RV_i\tilde{R}$
    $H_V \leftarrow V_i \wedge W_{i+1}^*$
    $H_W \leftarrow W_i \wedge W_{i+1}^*$
    $R_i \leftarrow 1 + \widehat{\hat{H}_V^* \hat{H}_W^*}$
    $R \leftarrow R_iR$
  end for

4 | NILPOTENT CONTROL PROBLEMS IN GA APPROACH

We use the symmetries of $SO(3)$ to define an equivalence relation on the set of geodesics passing through the origin; see Proposition 1. We find a convenient representative of any equivalence class and describe the moduli space in the language of GA.

4.1 | Geodesics of (3, 6)

Since the vector product $x \times h$ coincides with minus the dual of wedge product $x \wedge h$ according to (25), the horizontal system (12) can be written in the form

$$\begin{aligned} \dot{x} &= h, \\ \dot{z} &= -\frac{1}{2} x \wedge h, \end{aligned} \tag{27}$$

where $x \in \wedge^1 \mathbb{R}^3$ represents a line and $z \in \wedge^2 \mathbb{R}^3$ represents a plane in $\mathbb{R}^3$. In this way, we see the geodesics as curves in the geometric algebra $\mathbb{G}_3$.

Proposition 2. *Each arc-length parameterized sub-Riemannian geodesic satisfying the initial condition $x_i(0) = 0, z_i(0) = 0$, $i = 1, 2, 3$ is equivalent to a curve in $M \cong \wedge^1\mathbb{R}^3 \oplus \wedge^2\mathbb{R}^3 \subset \mathbb{G}_3$ and up to the action of a suitable $R \in SO(3)$, it takes the form*

$$\begin{aligned} q(t) = x(t) + z(t) = {}& \frac{D}{K}(1 - \cos(Kt))e_1 + \frac{D}{K}\sin(Kt)e_2 + C_3 t e_3 - \frac{D^2}{2K^2}(Kt - \sin(Kt))e_1 \wedge e_2 \\ & - \frac{C_3 D}{2K^2}(Kt - 2\sin(Kt) + Kt\cos(Kt))e_3 \wedge e_1 + \frac{C_3 D}{2K^2}(2 - Kt\sin(Kt) - 2\cos(Kt))e_2 \wedge e_3, \end{aligned} \tag{28}$$

where $K > 0$ and D, C_3 satisfy the level set equation $D^2 + C_3^2 = 1$.

Proof. The solution to the vertical system (11) can be rewritten as

$$h(t) = D\sin(Kt)\bar{v}_1 + D\cos(Kt)\bar{v}_2 + C_3 v_3, \tag{29}$$

where we denote $D = \sqrt{C_1^2 + C_2^2}$ and the orthonormal vectors $\bar{v}_1, \bar{v}_2$ are obtained by the rotation of orthonormal vectors v_1, v_2 as

$$\bar{v}_1 = \frac{1}{\sqrt{C_1^2 + C_2^2}}(-C_1 v_1 + C_2 v_2), \ \bar{v}_2 = \frac{1}{\sqrt{C_1^2 + C_2^2}}(C_2 v_1 + C_1 v_2).$$

Thus, the vectors $\bar{v}_1, \bar{v}_2, v_3$ are orthonormal with respect to the Euclidean metric on $\mathbb{R}^3$. So, there is an orthogonal matrix $R \in SO(3)$ that aligns vectors $\bar{v}_1, \bar{v}_2, v_3$ with the standard basis of $\mathbb{R}^3$. Thus, we get

$$\bar{v}_1 = Re_1, \ \bar{v}_2 = Re_2, \ v_3 = Re_3,$$

where e_1, e_2, and e_3 are the elements of the standard Euclidean basis of $\mathbb{R}^3$. According to (22), the rotor R defines a representative of a geodesic class $(R^T x(t), R^T z(t))$ which is a solution to (27) for $h(t) = D\sin(Kt)e_1 + D\cos(Kt)e_2 + C_3 e_3$. The solution (28) then follows by a direct integration when the initial condition is applied. Equation for the level set follows from the definition of D. □

The action of $SO(3)$ on $M \cong \mathbb{R}^6$ given by Equation (22) defines a moduli space $M/SO(3)$. We see M as a subset of $\mathbb{G}_3$ and the group $SO(3)$ is represented by rotors instead of matrices, which act on M by conjugation. The action preserves the vector and bivector parts, inner product, norm, and dualization with respect to $*$. We can see the elements of M as the pairs consisting of lines and planes. The natural invariants are the norms of lines' directional vectors, norms of the planes' normal vectors and angles between these pairs of vectors. Square norm of the normal vector of the plane $z^* \cdot z^*$ is $-z \cdot z$. Scalar product between the directional vector of the line x and the normal vector of the plane z can be rewritten as $(x \wedge z)^*$ because $(x \cdot z^*)^* = x \wedge z$ and $x \cdot z^* = (x \wedge z)^*$. Altogether, we consider three invariants

- the square norm of the vector x, that is, $x \cdot x$,
- the square norm of the bivector z, that is, $z \cdot z$,
- the element $(x \wedge z)^*$,

where $\cdot$ coincides with the inner product on $\mathbb{G}_3$. In particular, these invariants form a coordinate system on the moduli space $M/SO(3)$.

Proposition 3. *Each geodesic starting at the origin defines a curve in the moduli space $M/SO(3)$, which is determined by the invariants in the following way*

$$
\begin{aligned}
x \cdot x &= -\frac{2D^2}{K^2}(\cos(Kt) - 1) + C_3^2 t^2,\\
z \cdot z &= -\frac{D^2}{4K^4}((4C_3^2K^2 - 4C_3^2 - D^2)\cos(Kt)^2 + 2KC_3^2(2t(K-1)\sin(Kt) + t^2K - 4)\cos(Kt)\\
&\quad - 2Kt(4C_3^2 + D^2)\sin(Kt) + t^2(2C_3^2 + D^2)K^2 + D^2 + 8C3^2),\\
(x \wedge z)^* &= \frac{D^2C_3}{2K^3}((-2K+2)\cos(Kt)^2 + (2K+2)\cos(Kt) + K^2t^2 + Kt\sin(Kt) - 4).
\end{aligned}
\tag{30}
$$

Proof. Follows directly from (28). □

4.2 | Geodesics of (4, 7)

The base system (20) can be seen as a system in geometric algebra $\mathbb{G}_4$

$$
\begin{aligned}
\dot{x} + \dot{\ell} &= h_0 + \bar{h},\\
\dot{y} &= -x \wedge \bar{h} - \ell \wedge h_0,
\end{aligned}
\tag{31}
$$

where we assume that x and h_0 are collinear with e_1 and $\ell, \bar{h}$ in the subspace generated by e_2, e_3, e_4. The form of the second equation implies that y is given by minus the wedge product of e_1 and a vector from this subspace. Hence, the solution $y(t)$ can be viewed as a curve of planes in $\mathbb{G}_4$.

Proposition 4. *Each arc-length parameterized sub-Riemannian geodesic satisfying the initial condition $x(0) = 0, \ell_i(0) = 0, y_i(0) = 0, i = 1,2,3$ is equivalent to a curve in $N \cong \wedge^1\mathbb{R}^4 \oplus \wedge^2\mathbb{R}^4 \subset \mathbb{G}_4$ and up to the action of suitable $R \in SO(3)$, it takes the form*

$$
\begin{aligned}
q(t) = x(t) + \ell(t) + y(t) &= (C_1\cos(Kt) + C_2\sin(Kt) - C_1)e_1 + (C_1\sin(Kt) - C_2\cos(Kt) + C_2)e_2 + Cte_3\\
&+ \frac{1}{2}(C_1^2 + C_2^2)(tK - \sin(Kt))e_1 \wedge e_2 + \frac{C}{2K}((2C_1 - C_2Kt)\sin(Kt) - (C_1Kt + 2C_2)\cos(Kt) + 2C_2 - tC_1K)e_1 \wedge e_3,
\end{aligned}
\tag{32}
$$

where $K > 0$ and the constants C_1, C_2, C satisfy the level condition $K^2(C_1^2 + C_2^2) + C^2 = 1$.

Proof. According to the vertical system (19), the vector $\bar{h}(t)$ lies in the subspace generated by the vectors r_1, r_2 for any t. Since the vectors r_1 and r_2 are orthonormal, there is an orthogonal matrix $R \in SO(3)$ that aligns these vectors with the second and third vectors of the standard basis of $\mathbb{R}^3$, that is,

$$r_1 = Re_2, \ \ r_2 = Re_3.$$

Due to the symmetry of this system, see (23) this rotor defines a representative of the geodesic class $(x(t), R^T\ell(t), R^Ty(t))$ which is the solution to the horizontal system (20) for

$$
\begin{aligned}
h_0 &= K(C_2\cos(Kt) - C_1\sin(Kt)),\\
\bar{h}(t) &= K(C_2\sin(Kt) + C_1\cos(Kt))e_1 + Ce_2,
\end{aligned}
$$

or, equivalently, a curve in $\mathbb{R}^4 \oplus \Lambda^2\mathbb{R}^4 \in \mathbb{G}_4$ given by the solution of (31). By direct integration of this equation and by imposing the initial conditions, we get the formula (32) for the solution. □

The action of $SO(3)$ on $N \cong \mathbb{R}^7$ given by (23) defines a moduli space $N/SO(3)$. We see N as a subset of $\mathbb{G}_4$, and the group $SO(3)$ is represented by rotors instead of matrices, which act on N by the conjugation. The action preserves the vector and bivector part, the split $x + \ell$, inner product, norm, and dualization with respect to $*$. The orbits of this action are determined by natural invariants. For the same reason as in the case of $(3, 6)$ and due to the invariant split, we have three invariants as follows:

- the value of the coordinate x,
- the square of the norm of the vector ℓ, that is, $\ell \cdot \ell$,
- the square of the norm of the bivector y, that is, $y \cdot y$.

We need one more invariant for the dimensional reasons, but the element $(\ell \wedge y)^*$ is not scalar but vector. On the other hand, $(\ell \cdot y)$ is a multiple of the vector e_1, so the value of $(\ell \cdot y)e_1$ is a scalar. As the last invariant, we consider

- the value of $(\ell \cdot y)e_1$.

These form the coordinate system on the moduli space $N/SO(3)$.

Proposition 5. *Each geodesic starting at the origin defines a curve in the moduli space $N/SO(3)$, which is determined by the invariants in the following way*

$$\begin{aligned}
x &= C_1(\cos Kt - 1) + C_2 \sin Kt, \\
\ell \cdot \ell &= (C_1 \sin Kt + C_2(1 - \cos Kt))^2 + (Ct)^2, \\
(\ell \cdot y)e_1 &= \frac{1}{2}\left((C_1^2 + C_2^2)(C_1 \sin Kt + C_2(1 - \cos Kt))(Kt - \cos Kt) + \frac{C^2}{K} t (C_1(2 \sin Kt - Kt \cos Kt - Kt)\right. \\
&\quad \left. + \ C_2(2 - 2\cos Kt - Kt \sin Kt))\right), \\
y \cdot y &= \frac{1}{4}\left((C_1^2 + C_2^2)^2 (Kt - \cos Kt)^2 + \frac{C^2}{K^2}(C_1(2 \sin Kt - Kt \cos Kt - Kt) + C_2(2 - 2\cos Kt - Kt \sin Kt))^2\right).
\end{aligned} \tag{33}$$

Proof. Follows directly from (32). □

5 | EXAMPLES

In the sequel, we present two examples of controls based on the symmetries in geometric algebra approach. We have the following scheme based on Algorithm.

1. For the target point q_t compute the invariants of the chosen particular control system (2).
2. Solve the system of non-linear Equations (30) or (33) in the moduli space.
3. Find the family of curves (28) or (32) going from the origin to the same point q_o that belongs to the same $SO(3)$ orbit of q_t.
4. Find $R \in SO(3)$, such that $R(q_o) = q_t$.
5. Apply R on the set of curves (28) or (32) to get a family of curves going from the origin to the target point q_t.

The explicit calculations were acquired using a CAS system Maple similarly to the paper.[18]

5.1 | Example in $(3, 6)$

Our goal is to find the geodesic going from the origin to the target point

$$q_t = (x_t, z_t) = 2e_1 - e_2 + 3e_3 + e_1 \wedge e_2 - 2e_1 \wedge e_3 - 2e_2 \wedge e_3$$

using the invariants (30) in the target point. We have

$$x \cdot x = 14, \quad z \cdot z = -9, \quad (x \wedge z)^* = 3,$$

and together with the level set condition, we get the system with the invariants at q_t. We solve the system numerically in Maple and present the solution with rounding up to four decimal digits

$$C_3 = 0.7252, D = 0.6885, K = 0.9886, \tag{34}$$

$$t = 5.0236, \tag{35}$$

Using constant (34), we get the geodesic in the moduli space from the origin to the point q_o in the form

$$\begin{aligned} q = (x, z) &= (0.6965(1 - \cos(0.9886t)))e_1 + 0.6965\sin(0.9886t)e_2 + 0.7252te_3 \\ &- (0.2425(0.9886t - \sin(0.9886t)))e_1 \wedge e_2 + (0.2555(0.9886t - 2\sin(0.9886t) + 0.9886t\cos(0.9886t)))e_1 \wedge e_3 \\ &+ (0.2555(2 - 0.9886t\sin(0.9886t) - 2\cos(0.9886t)))e_2 \wedge e_3 \end{aligned}$$

and at the time $t = 5.0236$, we reach the point

$$q_o = 0.5216e_1 - 0.6741e_2 + 3.643e_3 - 1.439e_1 \wedge e_2 + 2.082e_1 \wedge e_3 + 1.611e_2 \wedge e_3. \tag{36}$$

We are looking for the rotor which maps the multivector q_o on the multivector q_t. We consider the complete flags

$$\begin{aligned} &\{0\} \subset NO(x_t) \subset NO(x_t \wedge z_t^*) \subset NO(z_t \wedge z_t^*) \cong \mathbb{R}^3, \\ &\{0\} \subset NO(x_o) \subset NO(x_o \wedge z_o^*) \subset NO(z_o \wedge z_o^*) \cong \mathbb{R}^3. \end{aligned}$$

We set $R_m = R_3 = \mathrm{id}$ and map the plane $x_o \wedge z_o^*$ to the plane $x_t \wedge z_t^*$ by the rotor $R_{m-1} = R_2$ according to formula (26). Explicitly,

$$R_2 := 0.1334 + 0.7083e_1 \wedge e_2 - 0.5483e_1 \wedge e_3 - 0.4242e_2 \wedge e_3$$

and we can map the multivector q_o on the multivector $q_s = (x_s, z_s) = R_2 q_o \tilde{R}_2$ in such a way that x_s and z_s lie in the plane $x_o \wedge z_o^*$. Explicitly,

$$q_s = (x_s, z_s) = -2.8510e_1 + 2.3208e_2 - 0.6956e_3 - 2.641e_1 \wedge e_2 + 1.2523e_1 \wedge e_3 + 0.6767e_2 \wedge e_3.$$

Finally, we map the plane $x_s \wedge (x_s \wedge z_s)^*$ to the plane $x_t \wedge (x_t \wedge z_t)^*$ by rotor

$$R_{m-2} = R_1 = 0.3727 + 0.1716e_1 \wedge e_2 + 0.6863e_1 \wedge e_3 - 0.6005e_2 \wedge e_3.$$

Altogether, we found the rotor $R = R_1 R_2 R_3$ and, when applied on (28), we got a geodesic going from the origin to the point q_t in the form

$$\begin{aligned} q = (x, z) &= (-0.5302 + 0.4673t + 0.5302\cos(0.9886t) - 0.05136\sin(0.9886t))e_1 \\ &- (0.008(22.124 + 36.347t - 22.124\cos(0.9886t) + 76.628\sin(0.9886t)))e_2 \\ &+ (-0.4156\cos(0.9886t) + 0.4723t + 0.4156 - 0.3267\sin(0.9886t))e_3 \\ &- (0.2(-1.5244 + 0.7536t\sin(0.9886t) + 0.4086\sin(0.9886t) \\ &+ 1.5244\cos(0.9886t) - 0.5923t\cos(0.9886t) + 0.1884t))e_1 \wedge e_2 \\ &+ (-0.3184t + 0.1299 - 0.2223t\cos(0.9886t) - 0.1299\cos(0.9886t) \\ &- 0.06419t\sin(0.9886t) + 0.547\sin(0.9886t))e_1 \wedge e_3 \\ &+ (-0.389 + 0.1923t\sin(0.9886t) - 0.1358t + 0.0186t\cos(0.9886t) \\ &+ 0.389\cos(0.9886t) + 0.1186\sin(0.9886t))e_2 \wedge e_3. \end{aligned}$$

In Figure 2, we present the trajectories (x_1, x_2, x_3) and (z_1, z_2, z_3), respectively.

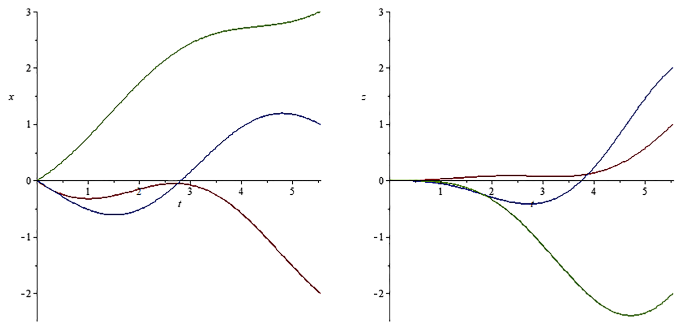


FIGURE 2 Trajectories (x_1, x_2, x_3) and (z_1, z_2, z_3) [Colour figure can be viewed at wileyonlinelibrary.com]

5.2 | Example in (4, 7)

Our goal is to find the geodesic going from the origin to the target point

$$q_t = (x_t, \ell_t, y_t) = e_1 + 2e_2 + e_3 + 3e_4 - e_1 \wedge e_2 + 2e_1 \wedge e_3 + 2e_1 \wedge e_4$$

using the invariants (33) at the target point. We compute

$$x = 1, \quad \ell \cdot \ell = 14, \quad y \cdot y = -9, \quad (\ell \cdot y)e_1 = -6,$$

and together with the level set condition, we get the system with the invariants at q_t. We solve the system numerically in Maple, and we present the solution with the constants rounded up to four decimal digits as follows:

$$C = 0.6126, C_1 = -0.7816, C_2 = -0.5324, K = 0.8358, \tag{37}$$

$$t = 6.0748. \tag{38}$$

Using constant (37), we get a geodesic in the moduli space from the origin to the point q_o in the form

$$\begin{aligned} q = (x, \ell, y) = {} & (-0.7816\cos(0.8358t) - 0.5324\sin(0.8358t) + 0.7816)e_1 \\ & + (-0.7816\sin(0.8358t) + 0.5324\cos(0.8358t) - 0.5324)e_2 \\ & + 0.6126te_3 + 0.4471(0.8358t - \sin(0.8358t))e_1 \wedge e_2 \\ & + 0.3665\left((0.4449t - 1.563)\sin(0.8358t) + (0.6533t + 1.065)\cos(0.8358t) - 1.065 + 0.6533t\right) e_1 \wedge e_3 \end{aligned}$$

and at the time $t = 6.0748$, we reach the point

$$q_o = (x_o, \ell_o, y_o) = e_1 + 0.3878e_2 + 3.722e_3 + 2.688e_1 \wedge e_2 + 1.332e_1 \wedge e_3. \tag{39}$$

We are looking for the rotor which maps the multivector q_o on the multivector q_t. We shall consider the complete flags starting with the line $NO(\ell)$ and ending with the space $NO(\ell \wedge y)$. To find the middle one, we can use the projection of the line $NO(\ell)$ onto the plane $NO(\ell \wedge y)$. Thus, we get

$$\{0\} \subset NO(\ell_t) \subset NO(\ell_t \wedge (\ell_t \wedge y_t^*)^*) = NO(\ell_t \wedge (\ell_t \cdot y_t)) \subset NO(\ell_t \wedge y_t) \subset NO(y_t \wedge y_t^*) \cong \mathbb{R}^4,$$
$$\{0\} \subset NO(\ell_o) \subset NO(\ell_o \wedge (\ell_o \wedge y_o^*)^*) = NO(\ell_o \wedge (\ell_o \cdot y_o)) \subset NO(\ell_o \wedge y_o) \subset NO(y_o \wedge y_o^*) \cong \mathbb{R}^4.$$

First, we map the hyperplane $\ell_o \wedge y_o$ to the hyperplane $\ell_t \wedge y_t$ by the rotor R_3 according to formula (26). We obtain

$$R_3 := 0.4863 - 0.4335e_2 \wedge e_4 - 0.7587e_3 \wedge e_4.$$

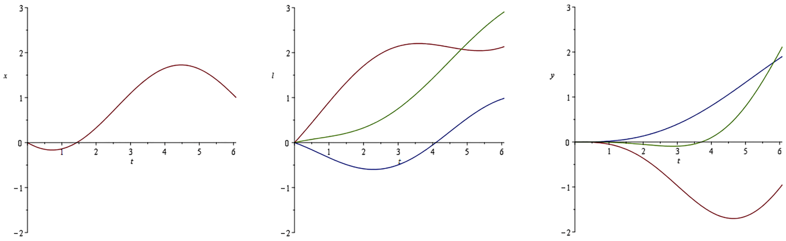


FIGURE 3 Trajectories x, (ℓ_2, ℓ_3, ℓ_4) and (y_1, y_2, y_3) [Colour figure can be viewed at wileyonlinelibrary.com]

The next step is to map the hyperplane $NO(\ell_o \wedge (\ell_o \cdot y_o) \wedge (\ell_o \wedge y_o)^*)$ to the hyperplane $NO(\ell_t \wedge (\ell_t \cdot y_t) \wedge (\ell_t \wedge y_t)^*)$ by the rotor R_2 according to formula (26). Explicitly,

$$R_2 := 0.0387 + 0.9993 e_2 \wedge e_3. \tag{40}$$

Finally, we map the hyperplane $NO(\ell_o \wedge (\ell_o \wedge (\ell_o \wedge y_o^*)^*)^*)$ to the hyperplane $NO(\ell_t \wedge (\ell_t \wedge (\ell_t \wedge y_t^*)^*)^*)$ by the rotor R_1 according to the formula (26). It turns out that $R_1 = 1$. Altogether, $R = R_1 R_2 R_3$ and, when applied on (5.2), we get a geodesic going from the origin to the point q_t as

$$\begin{aligned}
q = (x, \ell, y) &= (-0.7816\cos(0.8358t) - 0.5324\sin(0.8358t) + 0.7816)e_1 \\
&+ (0.5261\sin(0.8358t) - 0.3584\cos(0.8358t) + 0.3946t + 0.3584)e_2 \\
&+ (-0.4749\sin(0.8358t) + 0.3235\cos(0.8358t) + 0.1235t - 0.3235)e_3 \\
&+ ((0.2513 + 0.1542t)\cos(0.8358t) + (-0.06803 + 0.1050t)\sin(0.8358t) - 0.2514 - 0.09731t)e_1 \wedge e_2 \\
&+ ((0.07868 + 0.04827t)\cos(0.8358t) + (-0.3871 + 0.03287t)\sin(0.8358t) - 0.07869 + 0.2753t)e_1 \wedge e_3 \\
&+ ((0.2880 + 0.1767t)\cos(0.8358t) + (0.1203t - 0.6112)\sin(0.8358t) + 0.3342t - 0.2880)e_1 \wedge e_4.
\end{aligned}$$

In Figure 3 we present trajectories x, (ℓ_2, ℓ_3, ℓ_4) and (y_1, y_2, y_3), respectively.

6 | CONCLUSION

We presented the use of geometric algebra for the control systems invariant with respect to the orthogonal transformations. The main contribution of GA lies in a construction of the rotor between two bases of a vector space based only on algebraic computations in a chosen GA. This allows us to use the geometric objects effectively, and analogously to quaternions, the implementations are faster than the usual computations with matrices. We assessed an algorithm and illustrated its use on two particular examples with filtration $(3, 6)$ corresponding to a trident snake robot control and $(4, 7)$ corresponding to the control of a trident snake with flexible leg. All calculations were acquired using Maple packages Clifford[19] and DifferentialGeometry.[20]

ACKNOWLEDGEMENTS

The first three authors were supported by grant no. FSI-S-20-6187. The fourth author is supported by grant no. 20-11473S, symmetry and invariance in analysis, geometric modeling ,and control theory from the Czech Science Foundation. Finally, we thank the referee for valuable comments.

CONFLICT OF INTEREST

This work does not have any conflicts of interest.

ORCID

Jaroslav Hrdina https://orcid.org/0000-0003-2460-7571
Aleš Návrat https://orcid.org/0000-0002-8100-4032
Petr Vašík https://orcid.org/0000-0001-6725-7378
Lenka Zalabová https://orcid.org/0000-0002-3612-7068

REFERENCES

1. Bloch AM. *Nonholonomic Mechanics and Control*. Springer; 2015.
2. Jurdjevic V. *Geometric control theory*, Cambridge Studies in Advanced Mathematics. Cambridge: Cambridge University Press; 1996.
3. Agrachev A, Barilari D, Boscain U. *A comprehensive introduction to sub-Riemannian geometry. From the Hamiltonian viewpoint*, Cambridge Studies in Advanced Mathematics, Vol. 181. Cambridge University Press; 2020.
4. Agrachev AA, Sachkov YL. *Control theory from the geometric viewpoint*, Encyclopaedia of Mathematical Sciences, Control Theory and Optimization: Springer; 2006.
5. Hrdina J, Návrat A, Vašík P, Matoušek R. Geometric control of the trident snake robot based on CGA. *Adv Appl Clifford Algebras*. 2017;27:633. doi:10.1007/s00006-016-0693-7
6. Ishikawa M. Trident snake robot: locomotion analysis and control. *IFAC Symp Nonlin Control Syst*. 2004;6:895-900. doi:10.1016/S1474-6670(17)31339-3
7. Hrdina J, Zalabová L. Local geometric control of a certain mechanism with the growth vector (4,7). *J Dyn Control Syst*. 2020;26:199-216. doi:10.1007/s10883-019-09460-7
8. Bellaiche A. The tangent space in sub-Riemannian geometry. *Sub-Riemannian geometry*. Basel: Birkhäuser; 1996:1-78.
9. Myasnichenko O. Nilpotent (3, 6) sub–Riemannian problem. *J Dyn Control Syst*. 2002;8(4):573-597. doi:10.1023/A:1020719503741
10. Jean F. *Control of Nonholonomic Systems: From Sub-Riemannian Geometry to Motion Planning*. Springer; 2014.
11. Hildenbrand D. *Foundations of Geometric Algebra Computing*. Springer Verlag; 2013.
12. Perwass C. *Geometric Algebra with Applications in Engineering*. Springer Verlag; 2009.
13. Lounesto P. *Clifford Algebra and Spinors*, 2nd ed. Cambridge: CUP; 2006.
14. Hrdina J, Návrat A, Zalabová L. On symmetries of Sub–Riemannian structure with growth vector (4,7). *Annali di Matematica*. 2022. doi:10.1007/s10231-022-01242-6
15. Montanari A, Morbidelli G. On the sub–Riemannian cut locus in a model of free two-step Carnot group. *Calc Var*. 2017;56:36. doi:10.1007/s00526-017-1149-1
16. Rizzi L, Serres U. On the cut locus of free, step two Carnot groups. *Proc Amer Math Soc*. 2017;145:5341-5357. doi:10.1090/proc/13658
17. Hrdina J, Vašík P. Notes on differential kinematics in conformal geometric algebra approach. *Advances in intelligent systems and computing*, Vol. 378. Cham, Switzerland: Springer; 2015:363-374. doi:10.1007/978-3-319-19824-8_30
18. Hrdina J, Návrat A, Zalabová L. Symmetries in geometric control theory using Maple. *Math Comput Simul*. 2021;190:474-493.
19. Ablamowicz R, Fauser B. Mathematics of Clifford - a Maple package for Clifford and graßmann algebras. *Adv Appl Clifford Algebr*. 2005;15:157-181. doi:10.1007/s00006-005-0009-9
20. Anderson IM, Torre CG. New symbolic tools for differential geometry, gravitation, and field theory. *J Math Phys*. 2015;53:13511. doi:10.1063/1.3676296

How to cite this article: Hrdina J, Návrat A, Vašík P, Zalabova L. A note on geometric algebras and control problems with SO(3)-symmetries. *Math Meth Appl Sci*. 2022;1-17. doi:10.1002/mma.8662

Appendix 2

[29] Hrdina, J., Návrat, A., Vašík, P. et al.: *Projective Geometric Algebra as a Subalgebra of Conformal Geometric algebra*, Adv. Appl. Clifford Algebras 31, 18 (2021). https://doi.org/10.1007/s00006-021-01118-7

Adv. Appl. Clifford Algebras (2021) 31:18
© 2021 The Author(s), under exclusive licence to Springer Nature Switzerland AG part of Springer Nature
0188-7009/020001-14
published online February 22, 2021
https://doi.org/10.1007/s00006-021-01118-7

Advances in Applied Clifford Algebras

Projective Geometric Algebra as a Subalgebra of Conformal Geometric algebra

Jaroslav Hrdina, Aleš Návrat*, Petr Vašík and Leo Dorst

Communicated by Dietmar Hildenbrand

Abstract. We show that if Projective Geometric Algebra (PGA), i.e. the geometric algebra with degenerate signature (n, 0, 1), is understood as a subalgebra of Conformal Geometric Algebra (CGA) in a mathematically correct sense, then flat primitives share the same representation in PGA and CGA. Particularly, we treat duality in PGA in the framework of CGA. This leads to unification of PGA and CGA primitives which is important especially for software implementation and symbolic calculations.

Mathematics Subject Classification. Primary 15A66, Secondary 51N25.

Keywords. Conformal geometric algebra, Projective geometric algebra, Euclidean geometry.

1. Introduction

Projective geometric algebra (PGA) is a model for Euclidean geometry and computations with flat primitives. We use "PGA" to refer to geometric algebras with degenerate signature $(n, 0, 1)$, in particular it covers both Euclidean PGA and dual Euclidean PGA, see [4,5]. Conformal geometric algebra (CGA) defined by nondegenerate signature $(n + 1, 1)$ contains the same model and, moreover, allows Euclidean transformations of round primitives and dilation (conformal geometry), see [8–11]. Clearly, PGA is a subalgebra of CGA but the representation of Euclidean geometry looks very different at the first sight. PGA representation of a point is a multivector of grade $n - 1$ while a CGA point is of grade 1. This indicates that we have to think dually, or in other words in a complementary way. In what follows, we clarify how PGA can be viewed in CGA. We note that such an inclusion has been introduced

The first three authors were supported by a Grant No.: FSI-S-20-6187.

*Corresponding author.

Birkhäuser

in [12], yet we propose that in our notation it is clear that the flat primitives precisely coincide in PGA and CGA. By a choice of basis and signs we show that PGA duality can be completely described on the elements of CGA using the CGA duality. Therefore our notation slightly differs compared to [6] and Table 1. Consequently, a unified approach to both algebras is introduced. We treat the case $n = 3$ in this paper, however, the results hold for arbitrary dimension n, in particular also for $n = 2$. Our observation reads that if one needs to use calculations in PGA, it is enough to implement CGA only and therefore there is no need to implement a structure with degenerate metric.

First, we briefly introduce the frameworks of CGA and PGA as models of Euclidean geometry and we summarize basic formulae in Sect. 2. In Sect. 3, we show that there are two naturally related copies of PGA in CGA, see Proposition 3.1. After the identification of the two copies, duality in PGA is obtained in terms of CGA operations, see Proposition 3.2. The duality directly describes the correspondence between flat primitives and versors for Euclidean transformations in CGA, and the objects and versors in PGA, see Proposition 3.5. Basic ideas are then demonstrated on a simple example.

1.1. Notation

We denote elements of geometric algebras by bold letters - capitals for general multivectors and lower case letters for vectors. We also set the notation such that we can easily distinguish the elements of different algebras. Namely, $\mathbf{A}, \mathbf{B}, \dots$ will denote multivectors in PGA, $\mathbf{A}_c, \mathbf{B}_c, \dots$ will denote multivectors in CGA and $\mathbf{A}_E, \mathbf{B}_E, \dots$ will denote the elements of $\mathbb{G}_3$, i.e. the geometric algebra of three dimensional space. Similar notation will be used for objects and transformations of algebras, namely $\mathbf{P}, \boldsymbol{\ell}, \mathbf{p}$ will denote a point, line and plane in PGA, respectively, while $\mathbf{P}_c, \boldsymbol{\ell}_c, \mathbf{p}_c$ will denote direct representations of the respective objects in CGA. The corresponding dual representations will be accented by star superscript. By $\mathbf{P}_E$ we mean the representation of a Euclidean point in $\mathbb{G}_3$. The versors for rotations and translations in PGA and CGA will be denoted by $\mathbf{R}, \mathbf{T}$ and $\mathbf{R}_c, \mathbf{T}_c$, respectively. In order to distinguish among dualities in different algebras and to be consistent with the notation in the literature we denote the duality in CGA by an ordinary star symbol, by $*_P$ the duality in PGA and by $*_E$ the duality in $\mathbb{G}_3$, respectively.

2. Geometric Algebras for Euclidean Geometry

The geometric algebra $\mathbb{G}_3$ can describe vector geometry and rotations, see e.g. [1] for the geometric background. If we want to describe Euclidean geometry, we need to add a null vector in order to represent translations. The minimal way is to raise the dimension by one and add a one-dimensional space of null vectors. This model is known as the projective geometric algebra (PGA). However, this procedure returns a Clifford algebra with a degenerate quadratic form. Alternatively, we may raise the dimension by two and add a symplectic two–dimensional vector space. Then the quadratic form is nondegenerate with indefinite signature and we have even two linearly independent null vectors. The resulting model is known as conformal geometric algebra (CGA). We list the basic facts and formulae of both models.

2.1. Euclidean Geometry in CGA

From the algebraic point of view, CGA for dimension 3 Euclidean space is a Clifford algebra defined by a nondegenerate quadratic form of signature $(4, 1, 0)$. Vectors $\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_\infty$ denote an orthogonal basis of the generating vector space $\mathbb{R}^{4,1}$ with inner product given by the quadratic form

$$B = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1_{3\times 3} & 0 \\ -1 & 0 & 0 \end{pmatrix}. \tag{1}$$

Hence $\mathbf{e}_0, \mathbf{e}_\infty$ are null vectors and $\mathbf{e}_0 \cdot \mathbf{e}_\infty = \mathbf{e}_\infty \cdot \mathbf{e}_0 = -1$. The duality in CGA is defined by

$$\mathbf{A}_c^* = \mathbf{A}_c \mathbf{I}_c^{-1} = \mathbf{A}_c \cdot \mathbf{I}_c^{-1}, \tag{2}$$

where $\mathbf{I}_c = \mathbf{e}_{0123\infty}$ is the conformal *pseudoscalar*, $\mathbf{I}_c^{-1} = -\mathbf{I}_c$. The geometry in CGA is defined by the following embedding of a Euclidean point $\mathbf{P}_E$ into the geometric algebra $\mathbb{G}_3$, particularly onto an element $\mathbf{P}_c$ of the form

$$\mathbf{P}_c = \mathbf{e}_0 + \mathbf{P}_E + \tfrac{1}{2}(\mathbf{P}_E \cdot \mathbf{P}_E)\mathbf{e}_\infty, \tag{3}$$

where $\mathbf{P}_E \cdot \mathbf{P}_E$ coincides with the square of the Euclidean norm. In coordinates, if $\mathbf{P}_E = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$, then we get the well known formula

$$\mathbf{P}_c = \mathbf{e}_0 + x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3 + \tfrac{1}{2}(x^2 + y^2 + z^2)\mathbf{e}_\infty \tag{4}$$

together with the standard property $\mathbf{P}_c \cdot \mathbf{P}_c = 0$. The nondegeneracy of the quadratic form (1) implies that we have two mutually dual representations of geometric primitives in CGA. Namely, a multivector $\mathbf{A}_c$ is the *direct representation* (also called Outer Product Null Space (OPNS) representation) of an object in CGA if and only if the object is formed exactly by points $\mathbf{P}_c$ satisfying

$$\mathbf{P}_c \wedge \mathbf{A}_c = 0. \tag{5}$$

The duality in CGA given by (2) defines a *dual representation* (or IPNS representation). Namely, the same object can be also represented by $\mathbf{A}_c^*$ in the sense that it is formed exactly by points $\mathbf{P}_c$ satisfying

$$\mathbf{P}_c \cdot \mathbf{A}_c^* = 0, \tag{6}$$

where the dot denotes the *inner product*. Note that this duality of representations follows from the duality between the inner and outer product. The direct representation is useful for constructing geometric primitives as a join of points while the advantage of the dual representation is that one can easily read off the internal parameters of the primitives and find intersection of spheres.

Taking outer products of points in CGA we get representatives of general spheres spanned by these points, i.e. point pairs, circles, spheres and also flat primitives if one of the points lies at infinity. Thus for a flat point $\mathbf{FP}_c$, a line

$\boldsymbol{\ell}_c$ spanned by points $\mathbf{P}_{1c}, \mathbf{P}_{2c}$ and a plane $\mathbf{p}_c$ spanned by points $\mathbf{P}_{1c}, \mathbf{P}_{2c}, \mathbf{P}_{3c}$ we have the following respective representations

$$\mathbf{FP}_c = \mathbf{P}_c \wedge \mathbf{e}_\infty, \tag{7}$$

$$\boldsymbol{\ell}_c = \mathbf{P}_{1c} \wedge \mathbf{P}_{2c} \wedge \mathbf{e}_\infty, \tag{8}$$

$$\mathbf{p}_c = \mathbf{P}_{1c} \wedge \mathbf{P}_{2c} \wedge \mathbf{P}_{3c} \wedge \mathbf{e}_\infty. \tag{9}$$

Let us also recall that Euclidean transformations are represented by versors which act on objects by conjugation. The versor for translation by vector $\vec{t}$, which we identify with $\mathbf{t}_E \in \mathbb{G}_3$, is given by

$$\mathbf{T}_c = e^{-\frac{1}{2}\mathbf{t}_E \mathbf{e}_\infty} = 1 - \tfrac{1}{2}\mathbf{t}_E \mathbf{e}_\infty. \tag{10}$$

The versor for rotation by an angle α and with normalised dual representation of the rotation axis $\boldsymbol{\ell}^*$, i.e. $\boldsymbol{\ell}^* \cdot \boldsymbol{\ell}^* = -1$, is represented by CGA element

$$\mathbf{R}_c = e^{\frac{1}{2}\alpha \boldsymbol{\ell}^*} = \cos\tfrac{\alpha}{2} + \sin\tfrac{\alpha}{2}\boldsymbol{\ell}^*. \tag{11}$$

Remark 2.1. For general dimension n, we consider the orthogonal basis $\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_n, \mathbf{e}_\infty \in \mathbb{R}^{n+1,1}$ with inner product given by matrix (1), where the central block is an $n \times n$ identity matrix. The duality prescription (2) remains, however, the pseudoscalar satisfies

$$\mathbf{I}_c^{-1} = (-1)^{n(n-1)/2}\mathbf{I}_c, \tag{12}$$

thus the duality map is either an involution or an anti–involution depending on the dimension. Equation (3) defining a CGA point is independent of the dimension as well as formulae (10) and (11) for CGA transformations. The representations of Euclidean primitives (by which we understand flat primitives only) in CGA are $\mathbf{P}_{1c} \wedge \cdots \wedge \mathbf{P}_{kc} \wedge \mathbf{e}_\infty$ for any $1 \leq k \leq n$.

2.2. Euclidean Geometry in PGA

We shall use conventions and notation as close to [4,5] and [6] as possible, however, we modify the sign conventions for the duality in PGA slightly. We want to stress that these changes do not influence the validity of general formulae in [6] for incidence relations, projections, rejections, etc. We also note that in [3] the author strictly distinguishes geometric algebras generated by $\mathbb{R}^{3,0,1*}$ and $\mathbb{R}^{3,0,1}$, i.e. plane-based and point-based models, respectively. This is correct but algebraically these are isomorphic vector spaces and as algebras, $(\mathbb{G}_{3,0,1}, \vee) \cong (\mathbb{G}^*_{3,0,1}, \wedge)$ via Poincare isomorphism. Indeed, the identification of regressive product and wedge product via this duality allows us to use one algebra with both operations. This concept allows representatives of both points and planes to be elements of a single algebra. Algebraically, PGA for 3D Euclidean space is a Clifford algebra generated by a degenerate quadratic form of signature $(3, 0, 1)$. We consider a basis of the generating vector space $\mathbb{R}^{3,0,1}$ in which the quadratic form is given by the matrix

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 1_{3\times 3} \end{pmatrix}, \tag{13}$$

and we denote this basis by $\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$, i.e. $\mathbf{e}_0$ is a null vector. Note that the symbols for basis elements are the same as symbols for the corresponding basis elements in CGA in this notation, which is usual in the literature. Indeed, PGA can be viewed as a subalgebra of CGA spanned by these elements. However for relating the geometry of PGA and CGA, the $\mathbf{e}_0$ from PGA plays rather the role of the element $\mathbf{e}_\infty$ in CGA notation. This relation is discussed in detail in the following sections.

The duality in PGA cannot be obtained in the same way as the duality in CGA, by division with the pseudoscalar, because the quadratic form (13) defining PGA is degenerate. One can use Hodge duality approach, [13], i.e. that the dual to a basis element $\mathbf{A}$ can be defined as the complement to the projective pseudoscalar, $\mathbf{A} \wedge \mathbf{A}^* = \mathbf{I}$ or in the reverse order. However, such a map is neither involutive nor anti–involutive in general and the signs coming from such a definition are not compatible with the CGA duality. Therefore we use the idea of a Poincare duality approach, [4,5], which is a mapping between an algebra and its dual. In our concept, both these algebras can be understood as isomorphic subalgebras of CGA which leads to the Definition 3.2, see the next section. Note that similar concept of duality has been introduced in [12].

The representation of the Euclidean geometry in PGA is given by the embedding of a point. In coordinates, a Euclidean point $\mathbf{P}_E = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$ is represented in PGA by a multivector of grade three

$$\mathbf{P} = x\mathbf{e}_{032} + y\mathbf{e}_{013} + z\mathbf{e}_{021} + \mathbf{e}_{123}. \tag{14}$$

The formula for a point in PGA can be written in a coordinate free way using the Euclidean duality $*_E$, given by the division with the Euclidean pseudoscalar $\mathbf{I}_E = \mathbf{e}_{123}$, particularly $\mathbf{P} = \mathbf{e}_0 \mathbf{P}_E^{*_E} + \mathbf{I}_E$, which can be rewritten as

$$\mathbf{P} = \mathbf{I}_E + \mathbf{P}_E^{*_E} \mathbf{e}_0, \tag{15}$$

since the grade (width of the basis blades) of $\mathbf{P}_E^{*_E}$ is two. The degeneracy of the PGA quadratic form (13) causes us to have only one representation of geometric primitives in PGA. Since the grade of points is three, therefore sub–maximal, the only way to represent primitives is as the null space of the regressive product (RPNS representation). Recall that the regressive product is dual to the outer product, i.e. $(\mathbf{A} \vee \mathbf{B})^* = \mathbf{A}^* \wedge \mathbf{B}^*$. Hence a point represented by $\mathbf{P}$ belongs to an object represented by $\mathbf{A}$ if and only if

$$\mathbf{P} \vee \mathbf{A} = 0. \tag{16}$$

By regressive products of respective points in PGA we get representatives of flat primitives. Thus for a line $\boldsymbol{\ell}$ spanned by points $\mathbf{P}_1, \mathbf{P}_2$ and a plane $\mathbf{p}$ spanned by points $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ we have the representations

$$\boldsymbol{\ell} = \mathbf{P}_1 \vee \mathbf{P}_2, \tag{17}$$

$$\mathbf{p} = \mathbf{P}_1 \vee \mathbf{P}_2 \vee \mathbf{P}_3. \tag{18}$$

One can find several formulae for PGA representations of Euclidean transformations in [4], however a direct formula for translation by vector $\mathbf{t}_E$ is missing. If one reads between the lines, the versor for translation in PGA is given by

$$\mathbf{T} = e^{-\frac{1}{2}\mathbf{e}_0 \mathbf{t}_E} = 1 - \tfrac{1}{2}\mathbf{e}_0 \mathbf{t}_E. \tag{19}$$

Note that this formula is a direct consequence of formula (10) for the translator in CGA and Proposition 3.5 which will be proved later. Rotation is realized by (11), the same form of a versor as in CGA, i.e. $\mathbf{R}_c$. However, the dual line $\boldsymbol{\ell}^*$ in the formula must be replaced by $\boldsymbol{\ell}$ given by (17) in this case.

Remark 2.2. For general dimension n, we consider the orthogonal basis $\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_n \in \mathbb{R}^{n,0,1}$ with the inner product given by matrix (13), where the lower–right block is an $n \times n$ identity matrix. Equation (15) defines a PGA point in any dimension. The representations of transformations are independent of the dimension and Euclidean primitives are given by $\mathbf{P}_1 \vee \cdots \vee \mathbf{P}_k$ for any $1 \leq k \leq n$. The duality in nD PGA is discussed in the next Section, see Remark 3.4 .

3. PGA in CGA

Our main finding will be that the usual choice of basis and inner product in CGA, gives two distinct subalgebras of CGA which are both algebraically isomorphic to PGA and also an involutive isomorphism on CGA that relates these two subalgebras. In such identification, the duality in PGA can be seen as a "twisted" CGA duality. We note that this concept has been already published in [12]. Indeed, the author represents the projectivised exterior algebras of $\mathbb{R}^{n+1}$ and $\mathbb{R}^{n+1*}$ as subalgebras of CGA and defines the duality by means of two different pseudoscalars of these spaces by assertion (18.37). The form of this duality corresponds to our concept up to the sign. In the notation of [12], $\hat{W}$ corresponds to CGA_0 and $\hat{W}^*$ to CGA_∞ . Furthermore, $\hat{J}$ corresponds to $\sharp$. We used notation that is more familiar to mathematicians and provide connections with CGA to clarify transitions between algebras.

3.1. Duality in PGA

In CGA, the subalgebra formed by elements which do not contain $\mathbf{e}_\infty$ is present as well as the subalgebra formed by elements which do not contain $\mathbf{e}_0$. The former is generated by $\{\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ and will be denoted by CGA_0 and the latter is generated by $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_\infty\}$ and will be denoted by CGA_∞. The notation accents the null vector present and stresses the fact that both are subalgebras of CGA.

The choice of CGA basis and the inner product defines also a distinct involution in CGA which relates subalgebras CGA_0 and CGA_∞. Namely, the choice defines an isomorphism between the two vector spaces, $\mathbb{R}^{4,1}$ and its dual, by taking the usual dual basis. The quadratic form (1) defines another isomorphism between these spaces which is known as the musical isomorphism in the literature. The composition of these two isomorphisms is a bijective linear map of $\mathbb{R}^{4,1}$ onto itself, for $i = 1, 2, 3$ given by

$$\sharp : \mathbf{e}_i \mapsto \mathbf{e}_i, \mathbf{e}_0 \mapsto -\mathbf{e}_\infty, \mathbf{e}_\infty \mapsto -\mathbf{e}_0, \tag{20}$$

where we used the notation of musical isomorphism in order to distinguish between the duality on vector space $\mathbb{R}^{4,1}$ and the usual CGA duality. The minus signs correspond to the choice of inner product (1), i.e. $\mathbf{e}_0 \cdot \mathbf{e}_\infty = -1$. This map is a linear involution and preserves the quadratic form in

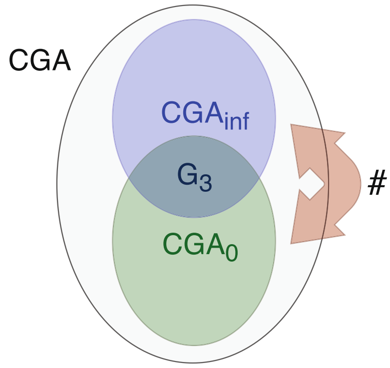


FIGURE 1. Two copies of PGA inside CGA

CGA, thus it defines a unique extension to CGA as a homomorphism of Clifford algebras. In the following, we will use the symbol $\sharp$ also for this extension. Using this notation we can summarize the above observations into the following statement.

Proposition 3.1. *The choice of basis of null vectors* $\mathbf{e}_0, \mathbf{e}_\infty$ *in CGA defines two subalgebras* CGA_0 *and* CGA_∞ *both isomorphic to PGA and an involutive isomorphism* $\sharp$ *between them which acts by replacing* $\mathbf{e}_0$ *with* $-\mathbf{e}_\infty$ *and vice versa in each basis blade.*

Let us describe the structure of subalgebras CGA_0 and CGA_∞ and the isomorphism $\sharp$ in a more detailed way. The intersection of these subalgebras is generated by $\langle \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \rangle$ with inner product given by the identity matrix, thus it forms the algebra $\mathbb{G}_3$. The map $\sharp$ that switches between the two subalgebras acts as identity on this intersection. On the complement of the union of the subalgebras to CGA, it acts as minus identity. Note that the union of subalgebras contains the elements with either e_0 or e_∞ only, therefore the complement to CGA is not empty containing elements with both e_0 and e_∞. Schematically, this is depicted in Fig. 1.

We stress that $\sharp$ is an isomorphism of Clifford algebras and thus it preserves all products in the algebra.

Once we know how to identify PGA with any of the two subalgebras in CGA, it is easy to understand duality in PGA. It can be defined in the same way as duality in CGA, by multiplication with a suitable pseudoscalar inverse. But we have to multiply by the inner product and also the pseudoscalar inverse must be taken with respect to the inner product. Such an inversion exists but it lies in the other subalgebra. Indeed, it is easy to see that $\mathbf{I} \cdot \mathbf{I}^\sharp = 1$ and thus for $\mathbf{I} = \mathbf{e}_{0123} \in \mathrm{CGA}_0$, the inner product inverse is $\mathbf{I}^\sharp = \mathbf{e}_{123\infty} \in \mathrm{CGA}_\infty$ and vice versa: for $\mathbf{I}^\sharp = \mathbf{e}_{123\infty} \in \mathrm{CGA}_\infty$, the inner product inverse is $(\mathbf{I}^\sharp)^\sharp = \mathbf{I}$.

Definition 3.2. Understanding PGA as a subalgebra of CGA, *PGA duality* is an involution defined for each $\mathbf{A} \in \mathrm{PGA}$ by

$$\mathbf{A}^{*_P} = (\mathbf{A} \cdot \mathbf{I}^\sharp)^\sharp = \mathbf{A}^\sharp \cdot \mathbf{I}. \tag{21}$$

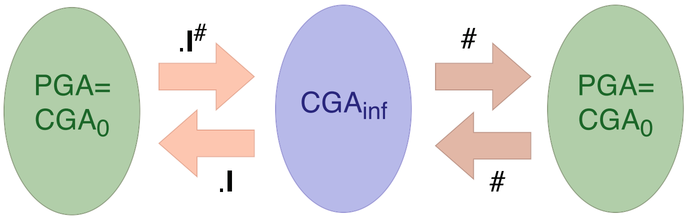


FIGURE 2. Duality in PGA viewed as a subalgebra of CGA

Note that the equality in the definition follows from the fact that $\sharp$ preserves the inner product. In the diagram visualisation, this can be described by the commutative diagram in Fig. 2. The duality in PGA is the composition from left to right, or equivalently from right to left, since it is an involution.

We also note that using a pseudoscalar in another subspace to move between subspaces is not new, see e.g. [2,3,12].

Having the duality in PGA defined according to the Definition 3.2, it is easy to relate it to the standard duality in CGA. Namely, it is a CGA duality twisted by a null vector as follows.

Proposition 3.3. *For* $\mathbf{A} \in \mathrm{PGA}$ *the following identities hold*

$$\mathbf{A}^{*_P} = (\mathbf{A} \wedge \mathbf{e}_\infty)^{*\sharp} = (\mathbf{A}^\sharp \wedge \mathbf{e}_0)^*. \tag{22}$$

Proof. By definition of PGA duality (21), we need to show $(\mathbf{A} \wedge \mathbf{e}_\infty)^* = \mathbf{A} \cdot \mathbf{I}^\sharp$ in order to prove the first identity. Indeed, we compute

$$(\mathbf{A} \wedge \mathbf{e}_\infty)^* = (\mathbf{A} \wedge \mathbf{e}_\infty) \cdot \mathbf{I}_c^{-1} = \mathbf{A} \cdot (\mathbf{e}_\infty \cdot \mathbf{I}_c^{-1}) = \mathbf{A} \cdot \mathbf{I}^\sharp.$$

The first equality is the CGA duality in terms of the inner product, see (2). The inner product coincides with the left contraction in this case since $\mathbf{I}_c^{-1}$ is of the highest grade. Then the second equality is a general property of left contraction. The third equality follows from the fact that $\mathbf{e}_\infty \cdot \mathbf{I}_c^{-1} = \mathbf{e}_\infty \cdot \mathbf{e}_{0321\infty} = (\mathbf{e}_\infty \cdot \mathbf{e}_0)\mathbf{e}_{321\infty} = -\mathbf{e}_{321\infty} = \mathbf{I}^\sharp$. The second identity in the proposition follows from the definition of isomorphism $\sharp$. Namely, $(\mathbf{A} \wedge \mathbf{e}_\infty)^\sharp = -\mathbf{A}^\sharp \wedge \mathbf{e}_0$ and $(\mathbf{I}_c^{-1})^\sharp = -\mathbf{I}_c^{-1}$. □

We also get a sort of PGA duality between the inner product and outer product similar to the duality between these products in CGA. Namely, supposing the grade of a blade $\mathbf{A}$ is less or equal to the grade of a blade$\mathbf{B}$, we have

$$(\mathbf{A} \wedge \mathbf{B})^{*_P} = \mathbf{A}^\sharp \cdot \mathbf{B}^{*_P}. \tag{23}$$

This formula holds for general multivectors $\mathbf{A}, \mathbf{B}$ if we replace the inner product on the right–hand side by left contraction. It is worth to look also at the relation between the duality in PGA and the usual Euclidean duality. For that we need to express a multivector $\mathbf{A} \in \mathrm{PGA}$ as a sum $\mathbf{C}_E + \mathbf{D}_E \wedge \mathbf{e}_0$, where $\mathbf{C}_E, \mathbf{D}_E \in \mathbb{G}_3$. Then we compute

$$\mathbf{A}^{*_P} = (\mathbf{C}_E + \mathbf{D}_E \wedge \mathbf{e}_0)^{*_P} = -\mathbf{D}_E^{*_E} + \mathbf{C}_E^{*_E} \wedge \mathbf{e}_0. \tag{24}$$

Table 1. PGA duality

	1	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_{01}$	$\mathbf{e}_{02}$	$\mathbf{e}_{03}$	$\mathbf{e}_{12}$	$\mathbf{e}_{13}$	$\mathbf{e}_{23}$	$\mathbf{e}_{012}$	$\mathbf{e}_{013}$	$\mathbf{e}_{023}$	$\mathbf{e}_{123}$	I
$\mathbf{A}$	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
$\mathbf{A}^{*_P}$	p	o	−n	m	−l	−k	j	−i	−h	g	−f	−e	d	−c	b	a

This formula is particularly convenient for an explicit computation of dual basis blade coefficients, see Table 1.

Remark 3.4. The definition of the isomorphism $\sharp$ is independent of the dimension. However, for general dimension n we have

$$\mathbf{I} \cdot \mathbf{I}^\sharp = (-1)^{n(n+1)/2}, \tag{25}$$

thus the inner product inversion to $\mathbf{I}$ gains this sign and it should also enter in formula (21) for the definition of the PGA duality. Note that PGA duality is an involution or anti–involution depending on the dimension. Formulae (22) and (23) hold in any dimension. Indeed, the only change in the proof of Proposition 3.3 is that $\mathbf{e}_\infty \cdot \mathbf{I}_c^{-1} = (-1)^{n(n+1)/2}\mathbf{I}^\sharp$. Equation (24) changes in nD to

$$(\mathbf{C}_E + \mathbf{D}_E \wedge \mathbf{e}_0)^{*_P} = (-1)^n \mathbf{D}_E^{*_E} + \mathbf{C}_E^{*_E} \wedge \mathbf{e}_0.$$

3.2. Geometry

All geometric primitives in PGA are constructed by means of the regressive product which is dual to the outer product, c.f. (17) and (18). Let us start by computing the projective dual of a point. By applying formula (21) for the PGA duality to the PGA point (15) we get

$$\mathbf{P}^{*_P} = \mathbf{e}_0 + \mathbf{P}_E. \tag{26}$$

Hence a point in PGA is the dual to its usual homogeneous representation in $\mathbb{R}^4$. Now we compute the representation of a PGA point in the subalgebra CGA$_\infty$. Applying the isomorphism $\sharp$ to formula (15) for a point we get

$$\mathbf{P}^\sharp = \mathbf{I}_E - \mathbf{P}_E^{*_E}\mathbf{e}_\infty. \tag{27}$$

We will show in the following Proposition 3.5 that this is a formula for the dual representation of a flat point in CGA. Indeed, looking at the direct representations of flat primitives and Euclidean transformations in CGA we observe that they all lie in the subalgebra CGA$_\infty$. Hence this subalgebra is the suitable copy of PGA in CGA for geometric purposes and the map $\sharp$ gives a geometric embedding in the following sense.

Proposition 3.5. *Let $\mathbf{P}, \boldsymbol{\ell}, \mathbf{p}$ be representations of a point, a line and a plane in PGA, respectively. Then $\mathbf{P}^\sharp, \boldsymbol{\ell}^\sharp, \mathbf{p}^\sharp$ are the dual (IPNS) representations of the same point (viewed as a flat point), line and plane in CGA. Moreover, if $\mathbf{V}$ is a versor for a Euclidean transformation in PGA, then $\mathbf{V}^\sharp$ is the versor for the same transformation in CGA.*

Proof. At first we prove the correspondence of points. We need to show that (27) is the form of the dual flat point $\mathbf{FP}_c^* = (\mathbf{P}_c \wedge \mathbf{e}_\infty)^*$. Indeed, we compute

$$\mathbf{FP}_c^* = (\mathbf{e}_{0\infty} + \mathbf{P}_E \mathbf{e}_\infty)^* = \mathbf{I}_E - \mathbf{P}_E^{*_E} \mathbf{e}_\infty = \mathbf{P}^\sharp. \tag{28}$$

For the proof of the correspondence between other geometric primitives we use the formula for dual point in terms of a conformal point. Since the dual point (26) actually is the conformal representation of a point without the quadratic part, we can write

$$\mathbf{P}^{*_P} = \mathbf{P}_c + (\mathbf{e}_0 \cdot \mathbf{P}_c)\mathbf{e}_\infty. \tag{29}$$

Now we apply this formula to a PGA line $\boldsymbol{\ell} = \mathbf{P}_1 \vee \mathbf{P}_2$. By the definition of the regressive product and formula (22) for the PGA duality we get

$$\begin{aligned}\boldsymbol{\ell}^\sharp &= [(\mathbf{P}_{1c} + (\mathbf{e}_0 \cdot \mathbf{P}_{1c})\mathbf{e}_\infty) \wedge (\mathbf{P}_{2c} + (\mathbf{e}_0 \cdot \mathbf{P}_{2c})\mathbf{e}_\infty) \wedge \mathbf{e}_\infty]^* \\ &= (\mathbf{P}_{1c} \wedge \mathbf{P}_{2c} \wedge \mathbf{e}_\infty)^* = \boldsymbol{\ell}_c^*.\end{aligned}$$

Similarly, for a plane $\mathbf{p} = \mathbf{P}_1 \vee \mathbf{P}_2 \vee \mathbf{P}_3$ we get

$$\mathbf{p}^\sharp = (\mathbf{P}_{1c} \wedge \mathbf{P}_{2c} \wedge \mathbf{P}_{3c} \wedge \mathbf{e}_\infty)^* = \mathbf{p}_c^*,$$

thus the representation of any object in PGA coincides with its dual representation in CGA. The claim about versors follows from the correspondence between objects and the fact that $\sharp$ preserves the geometric product and that it commutes with the reversion operation. ☐

Remark 3.6. The duality in PGA is defined in such way that dual PGA points always coincide with their homogeneous representations, i.e. the formula (26) holds for arbitrary dimension. On the other hand, the correspondence in Proposition 3.5 changes sign according to the sign of (25). Namely, in dimension n we have

$$\mathbf{P}^\sharp = (-1)^{n(n+1)/2}(\mathbf{P}_c \wedge \mathbf{e}_\infty)$$

and also the remaining objects are endowed with the same sign.

3.3. Example

It is obvious that we do not need the quadratic parts of points in CGA when we deal with flat primitives only. PGA is certainly more efficient in that case. However, we do not need to abandon the CGA concept while using all the amazing formulae from PGA. We only have to keep in mind that points correspond to flat points and that the PGA duality differs from conformal duality, see (21) and (22).

Let us consider an elementary example to demonstrate a convenient way of using PGA inside CGA. We have a point $\mathbf{P}$ lying on a line $\boldsymbol{\ell}$ which intersects a plane $\mathbf{p}$ and we have a sphere $\mathbf{s}$ which also intersects the plane, see Fig. 3.

Then we can use formulae from PGA to compute the intersection of the line and plane $\boldsymbol{\ell} \wedge \mathbf{p}$ and the orthogonal projection of the point to the plane $(\mathbf{P} \cdot \mathbf{p})\mathbf{p}$, the orthogonal projection of the line to the plane $(\boldsymbol{\ell} \cdot \mathbf{p})\mathbf{p}$. If we need to calculate with the sphere, we only need to replace $\mathbf{e}_0$ by $-\mathbf{e}_\infty$ in the representations of flat primitives, which is realized by the map $\sharp$, and then we can use all formulae known in CGA. For instance, the intersection of the

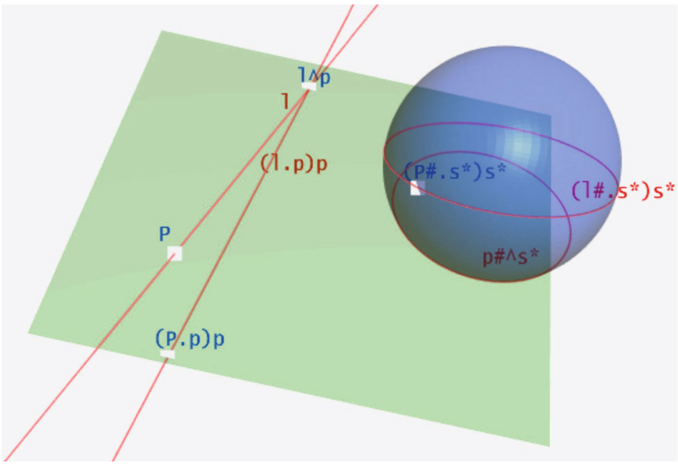


FIGURE 3. Intersections and projections of flats and rounds

sphere and the plane is a dual circle $\mathbf{p}^\sharp \wedge \mathbf{s}^*$, the orthogonal projection of the point to the sphere is $(\mathbf{P}^\sharp \cdot \mathbf{s}^*)\mathbf{s}^*$ and the orthogonal projection of the line to the sphere is a circle $(\boldsymbol{\ell}^\sharp \cdot \mathbf{s}^*)\mathbf{s}^*$.

Figure 3 was created by the web–based experimantal platform of Ganja.js [7]. The corresponding full code follows.

```
// Create a Clifford Algebra with 4,1 metric for 3D CGA.
Algebra(4,1,()=>{

// We start by defining a null basis, and upcasting for points
// which we need for rounds only
var ni = 1e4+1e5, no = .5e5-.5e4;
var up = (x)=> no + x + .5*x*x*ni;

// Sharp map in both directions in terms of CGA products
var IN = (x)=> x + ni^(no<<x) + no^(no<<x);
var NI = (x)=> x + ni^(ni<<x) + no^(ni<<x);

// PGA pseudoscalar
var I = no^1e1^1e2^1e3;

// PGA duality and upcasting to PGA
var dual = (x)=> NI(x)<<I;
var upP = (x)=> dual(no+x);

// Definition of regressive product for 2 and three inputs
var reg = (x,y)=>dual(dual(x)^dual(y));
var reg3 = (x,y,z)=>dual(dual(x)^dual(y)^dual(z));

// Formulas from PGA can be used in CGA

// We define 4 points
var P = upP(0.5e1-1.5e3);
var P1 = upP(1e1), P2 = upP(1e2), P3 = upP(-1e3);
```

```
// Constructing line and plane from points
var l = ()=>reg(P,P2);
var p = ()=>reg3(P1,P2,P3);

// Intersection of line and plane
var Q = ()=>p^l;

// Projection of point and line to plane
var Pperp = ()=>(p<<P)*p;
var lperp = ()=>(p<<l)*p;

// If we want we can also intersect with a sphere or project on it
var s = ()=>up(0.5e2+1e1)-0.5*0.25*ni;

var c = ()=>s^NI(p);
var cperp = ()=>(s<<NI(l))*s;

// Graph the items as CGA elements A -> !NI(A)
document.body.appendChild(this.graph([
0xE0880000, !NI(P), "P",  // point
0xE0880000, !NI(Q), "l^p",  // point
0xE0880000, !NI(Pperp), "(P.p)p",  // point
0xE0000000,!NI(lperp) , "(l.p)p",  // line
0xE00000FF,!cperp , "(l.s*)s*",  // circ
0xE00000FF,!c , "p^s*",  // circ
0xE0000000, !NI(l), "l",  // line
0xE0008800, !NI(p), "p",  // plane
0xE00000FF, !s, "s"  // sphere
],{conformal:true,gl:true,grid:false}));
});
```

4. Conclusion

We introduced a notation for PGA primitives that is compatible with their CGA description, once PGA is understood as a subalgebra of CGA. We also solved the issues with the noninvertibility of the PGA pseudoscalar in computing duality and showed the exact forms of dual counterparts to geometric primitives in PGA using another copy of PGA in CGA. This has great potential for symbolic calculations and their software implementation, because we can flexibly switch between PGA notation which is efficient for flat object manipulation and CGA operations on round elements in our problems. The next step is to show the advantages of this approach in various computational platforms, together with code optimisation and applications. Generally, the idea is that it is enough to implement CGA only and there is no need to implement an extra structure with degenerate metric.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] Dorst, L., Fontijne, D., Mann, S.: Geometric algebra for computer science: an object-oriented approach to geometry. Morgan Kaufmann Publishers Inc., Burlington (2007)

[2] Du, J., Goldman, R., Mann, S.: Modeling 3D Geometry in the Clifford Algebra R(4, 4). Adv. Appl. Clifford Algebras **27**, 3039–3062 (2017). https://doi.org/10.1007/s00006-017-0798-7

[3] Gunn C.: On the Homogeneous model of euclidean geometry. In: Dorst L., Lasenby J. (eds) Guide to geometric algebra in practice. Springer, London (2011). https://doi.org/10.1007/978-0-85729-811-9_15

[4] Gunn, C.G.: Doing euclidean plane geometry using projective geometric algebra. Adv. Appl. Clifford Algebras **27**, 1203 (2017). https://doi.org/10.1007/s00006-016-0731-5

[5] Gunn, C.G.: Geometric algebras for Euclidean geometry. Adv. Appl. Clifford Algebras **27**, 185 (2017). https://doi.org/10.1007/s00006-016-0647-0

[6] Gunn, C.G., De Keninck, S.: *3D PGA Cheat Sheet, An extensive reference with 3D PGA formulas*, online: bivector.net

[7] De Keninck, S.: ganja.js, https://zenodo.org/record/3635774 (2020)

[8] Hildenbrand, D.: Foundations of geometric algebra computing. Springer Science & Business Media (2013)

[9] Hildenbrand, D.: Introduction to geometric algebra computing. Chapman and Hall/CRC, Boca Raton (2018)

[10] Hrdina, J., Návrat, A., Vašík, P.: Control of 3-link robotic snake based on conformal geometric algebra. Adv. Appl. Clifford Algebr. **26**(3), 1069–1080 (2016)

[11] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: CGA-based robotic snake control. Adv. Appl. Clifford Algebr. **27**(1), 621–632 (2017)

[12] Lasenby, A.: Rigid body dynamics in a constant curvature space and the '1d-up' approach to conformal geometric algebra. In: Dorst, L., Lasenby, J. (eds.) Guide to geometric algebra in practice. Springer, London (2011)

[13] Lounesto, P.: Clifford algebra and spinors, 2nd edn. CUP, Cambridge (2006)

Jaroslav Hrdina, Aleš Návrat and Petr Vašík
Institute of Mathematics, Faculty of Mechanical Engineering
Brno University of Technology
Brno
Czech Republic
e-mail: navrat.a@fme.vutbr.cz

Jaroslav Hrdina
e-mail: hrdina@fme.vutbr.cz

Petr Vašík
e-mail: vasik@fme.vutbr.cz

Leo Dorst
Informatics Institute, Faculty of Science
University of Amsterdam
Amsterdam
The Netherlands
e-mail: l.dorst@uva.nl

Received: April 24, 2020.
Accepted: January 13, 2021.

Appendix 3

[21] Hrdina, J., Návrat, A.: *Binocular Computer Vision Based on Conformal Geometric Algebra*, Adv. Appl. Clifford Algebras (2017) 27: 1945. https://doi.org/10.1007/s00006-017-0764-4

Adv. Appl. Clifford Algebras 27 (2017), 1945–1959
© 2017 Springer International Publishing
0188-7009/031945-15
published online February 20, 2017
DOI 10.1007/s00006-017-0764-4

Advances in Applied Clifford Algebras

CrossMark

Binocular Computer Vision Based on Conformal Geometric Algebra

Jaroslav Hrdina* and Aleš Návrat

Abstract. We apply the conformal geometric algebra (CGA) to the generalized binocular vision problem. More precisely, we reconstruct a 3D line from its images on the image planes of two cameras whose mutual position is specified by a given Euclidean transformation which depends on an arbitrary number of parameters. We represent all transformations by CGA elements which allows us to derive the general equations of 3D line reconstruction by formal CGA elements manipulation. The transformation equations can be solved w.r.t. either motor or projection unknown parameters. We present two specific examples, show the explicit form of two particular motors and solve the appropriate equations completely.

Mathematics Subject Classification. Primary 15A66; Secondary 68T45.

Keywords. Conformal geometric algebra, Clifford algebra, Binocular vision, Projective geometry.

1. Introduction

One of the main problems of binocular robot vision is a 3D object pose estimation from the data obtained by a pair of cameras. It can be solved in two steps. First, the symmetry axes are extracted from the camera image planes. Second, the pose estimation problem for these axes is solved. We discuss the second step in terms of conformal geometric algebra and find a solution of pose estimation problem for lines (axes) in a general case of two cameras in arbitrary positions. Our particular task relates to the case where the camera motions are restricted by construction with a certain degree of freedom and thus their positions in space are given by specific transformations depending on a certain number of parameters. The task is to define the camera position transformation to achieve the prescribed shape of an object in question whose position is given as part of the input data. Typically, this is solved to center the observed object on one camera. Furthermore, we assume that the cameras

This work was supported by Grants of the Grant Agency of the Czech Republic 17-21360S, “Advances in Snake-like Robot Control”.
*Corresponding author.

Birkhäuser

are in arbitrary mutual position the parameters of which are the only data needed. In classical literature, e.g. [7,8], the coordinate system is attached to one of the cameras which will be referred to as the first one. In our case, we choose the coordinate system connected to the base of the camera stand to simplify the correspondence of the rotation axes. We consider both cameras to be the pin-hole model and introduce the description of each of them as a pair of algebra elements, more precisely as a point F (focus) and a point pair $K = P \wedge Q$ from the camera projection plain such that the lines $F \wedge P \wedge e_\infty$ and $K \wedge e_\infty$ are orthogonal.

In this setting, the problem is solved by symbolic computations with conformal geometric algebra (CGA) objects in general, but we discuss the projective geometric algebra (PGA) setting simultaneously. Indeed, the last Section shows one of the possible directions that we intend to explore in the future, particularly the human vision where PGA is not sufficient tool. Another direction to be considered is e.g. the image analysis from the omni-directional camera.

The reason to use the GA (CGA) structures to solve such topics lies in the fact that the properties are independent of the particular mechanism. In particular, we derive formulas (3) and (4) which hold in general for any system. In Sect. 5 we show on two elementary examples the application of our computations onto two particular mechanisms and we present the numeric results for several specific settings. In the sense of an object oriented approach, we divide the binocular vision problem into three parts. The most abstract one contains the elaboration with the GA elements by means of standard operations and exploits their natural properties as the direct and dual representation of the Euclidean object. Our aim is to achieve a mechanism-independent geometric solution. Afterwards, we choose a particular GA (despite the fact that in this text we assume to work within CGA and use several arguments valid for CGA only, from the remarks in the text it is obvious that e.g. in case of a pin-hole camera the choice of CGA is not completely necessary). Finally, we introduce the equations of the particular mechanism which follows from the forward kinematic chain. Thus in general, the motors transforming the camera position from the initial ideal state to the actual one are obtained. Note that the computations are performed in Maple, package `CLIFFORD` [1], and the final set of equations, whose form does not demand deeper knowledge of CGA, can be used in engineering applications.

2. Conformal Geometric Algebra: CGA

We recall some elementary facts about CGA and specify our particular setting. Note that the properties and definitions of conformal geometric algebras can be found in e.g. [2,4,7] and their applications in engineering in e.g. [3,5]. Classically, for modeling a 3D robot, the CGA is the Clifford algebra $\mathcal{C}l(4,1)$ and an embedding

$$c : \mathbb{R}^3 \to \mathbb{K}^4 \subset \mathbb{R}^{4,1}$$

of the 3D Euclidean space is considered, where $\mathbb{K}^4$ is a null cone in the Minkowski space $\mathbb{R}^{4,1}$. We describe the embedding explicitly in a suitably chosen basis which in this paper is denoted by $e_0, e_1, e_2, e_3, e_\infty \in \mathbb{R}^{4,1}$. This basis is chosen so that the matrix of the quadratic form is

$$B = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1_{3\times 3} & 0 \\ -1 & 0 & 0 \end{pmatrix},$$

where $1_{3\times 3}$ is an identity 3×3 matrix. With these conventions, the embedding is of the form

$$c(x) = x + \frac{1}{2}x^2 e_\infty + e_0.$$

Note that e_0 and e_∞ play the role of the origin and the infinity, respectively. Note also that x^2 is a scalar which is equal to $\|x\|^2$, the square of usual Euclidean norm. We recall that the algebra operation is called the *geometric (Clifford) product* and two further elementary operations on CGA are derived; the *inner product (left contraction)* and *outer (wedge) product*. We recall just the basic properties used in this text. The wedge product of two basis blades E_k and E_l of grades k and l, respectively, is defined as

$$E_k \wedge E_l := \langle E_k E_l \rangle_{k+l}$$

and the left contraction is defined as

$$E_k \cdot E_l := \langle E_k E_l \rangle_{l-k} \quad \text{if } l \geq k \text{ and } 0 \text{ otherwise},$$

where $\langle\ \rangle_k$ is the grade projection into grade k. In the special case $k = l = 1$, the factors are vectors and the inner product and outer product correspond to the symmetric and antisymmetric part of the geometric product, respectively. These products can be used to define the mapping $c^{-1} : \mathbb{K}^4 - \{0, e_\infty\} \to \mathbb{R}^3$ by

$$c^{-1}(X) = P^{\perp}_{e_\infty \wedge e_0}\left(\frac{X}{-X \cdot e_\infty}\right) = \frac{X - (X \cdot (e_\infty \wedge e_0))(e_\infty \wedge e_0)}{-X \cdot e_\infty},$$

where $P^\perp$ denotes the orthogonal complement to the projection onto $e_\infty \wedge e_0$, which is the left inversion of c, i.e. $c^{-1}(c(x)) = x$ and allows an identification of the projectivized null cone $P\mathbb{K}^4$ and one point compactification $\mathbb{R}^3 \cup \{\infty\}$.

The embedding c has the fundamental property that the inner product of two conformal points is, up to the factor $-1/2$, the square of the Euclidean distance. Indeed,

$$\begin{aligned} c(x) \cdot c(y) &= \left(x + \tfrac{1}{2}x^2 e_\infty + e_0\right) \cdot \left(y + \tfrac{1}{2}y^2 e_\infty + e_0\right) \\ &= \left\langle xy + \tfrac{1}{2}(y^2 x - x^2 y)e_\infty + (x-y)e_0 + \tfrac{1}{2}x^2 e_\infty e_0 + \tfrac{1}{2}y^2 e_0 e_\infty \right\rangle_0 \\ &= xy - \tfrac{1}{2}x^2 - \tfrac{1}{2}y^2 = -\tfrac{1}{2}(x-y)^2 = -\tfrac{1}{2}\|x-y\|^2. \end{aligned}$$

Thus the embedding c, together with the inner product, gives a linearization of a squared distance. As a consequence, the round objects which are given

Table 1. Wedge basis of $\mathbb{G}_{4,1}$

Scalars	1
Vectors	$e_0,\ e_1,\ e_2,\ e_3,\ e_\infty$
2-blades	$e_0\wedge e_1,\ e_0\wedge e_2,\ e_0\wedge e_3,\ e_0\wedge e_\infty,\ e_1\wedge e_2,$ $e_1\wedge e_3,\ e_1\wedge e_\infty,\ e_2\wedge e_3,\ e_2\wedge e_\infty, e_3\wedge e_\infty$
3-blades	$e_0\wedge e_1\wedge e_2,\ e_0\wedge e_1\wedge e_3,\ e_0\wedge e_1\wedge e_\infty,\ e_0\wedge e_2\wedge e_3,$ $e_0\wedge e_2\wedge e_\infty,\ e_0\wedge e_3\wedge e_\infty,\ e_1\wedge e_2\wedge e_3,\ e_1\wedge e_2\wedge e_\infty,$ $e_1\wedge e_3\wedge e_\infty,\ e_2\wedge e_3\wedge e_\infty$
4-blades	$e_0\wedge e_1\wedge e_2\wedge e_3,\ e_0\wedge e_1\wedge e_2\wedge e_\infty,\ e_0\wedge e_1\wedge e_3\wedge e_\infty,$ $e_0\wedge e_2\wedge e_3\wedge e_\infty,\ e_1\wedge e_2\wedge e_3\wedge e_\infty$
Pseudoscalar	$I=e_0\wedge e_1\wedge e_2\wedge e_3\wedge e_\infty$

by quadratic expressions (spheres) are represented by linear objects and the Euclidean motions are represented by orthogonal transformations.

Algebraically, we consider the geometric algebra on $\mathbb{R}^{4,1}$, i.e. the algebra denoted as $\mathbb{G}_{4,1}$ which is the Clifford algebra $\mathcal{C}l(4,1)$, see [6]. In terms of the basis $e_0, e_1, e_2, e_3, e_\infty$ of $\mathbb{R}^{4,1}$, a basis of $\mathbb{G}_{4,1}$ is given by Grassmann monomials (blades) as displayed in Table 1.

The Euclidean objects which can be represented by elements in this algebra are spheres of any dimension, i.e. point pairs, circles, spheres, points (spheres of zero radius), linear objects of any dimension, i.e. flat points, lines, planes, and also Euclidean directional and tangential elements. They are represented in the following way. Let $\mathcal{S}\subseteq\mathbb{R}^3$ be one of the Euclidean elements listed above. Then it is viewed as an element $S\in\mathbb{G}_{4,1}$ such that

$$x\in\mathcal{S}\Leftrightarrow c(x)\wedge S=0.$$

Note that this is so-called outer product representation or direct representation. Dually, in the inner product representation, $\mathcal{S}$ is represented by an element S^* and the condition reads

$$x\in\mathcal{S}\Leftrightarrow c(x)\cdot S^*=0.$$

Note that on the algebra level the duality is obtained by dividing an algebra element S by the pseudoscalar I, i.e. $S^*=S/I$.

In the direct representation, the outer product $\wedge$ indicates the construction of a geometric object with the help of points P_i that lie on it. A point pair (0D sphere), is defined by two points $P_1\wedge P_2$. A circle (1D sphere) is defined by three points $P_1\wedge P_2\wedge P_3$ or a point pair and a point. Finally, a sphere (2D sphere) is defined by four points $P_1\wedge P_2\wedge P_3\wedge P_4$ or two point pairs, etc. A plane and line can also be defined by points that lie on it and by the point at infinity, i.e. a line is represented by $P_1\wedge P_2\wedge e_\infty$ and a plane by $P_1\wedge P_2\wedge P_3\wedge e_\infty$. In the dual representation, a sphere can be represented by its center P and its radius r as $P-\frac{1}{2}r^2e_\infty$. A plane is defined as $n+de_\infty$, where n is the unit normal vector of the plane and d is the distance to the origin.

In this sense, the wedge product is a constructive operator, i.e. $A \wedge B$ is an object spanned by A and B. The duality operator allows a definition of the dual to wedge product, so called *meet*,

$$A \vee B = (A^* \wedge B^*)^*.$$

Geometrically, this gives a CGA representative of the intersection of objects A and B.

Next advantage of CGA is that the Euclidean motions are also represented by certain algebra elements. Namely, an Euclidean transformation of an element S is realized in CGA by conjugation with an invertible element $T \in \mathbb{G}_{4,1}$, i.e.

$$S \mapsto TST^{-1},$$

such that $T\tilde{T} = 1$. Note that the inverse T^{-1} can be replaced by reverse $\tilde{T}$, see [7] for details. For instance, the translation in the direction $t = t_1e_1 + t_2e_2 + t_3e_3$ is realized by the multivector (*translator*)

$$T = 1 - \frac{1}{2}te_\infty$$

and the rotation around the origin and the normalized axis $L = L_1e_1+L_2e_2+L_3e_3$ by an angle ϕ is realized by the multivector (*rotor*)

$$R = \mathrm{e}^{-\frac{1}{2}l\phi} = \cos\frac{\phi}{2} - l\sin\frac{\phi}{2},$$

where $l = L^*_{3\mathrm{D}} = L(e_1 \wedge e_2 \wedge e_3) = L_1(e_2 \wedge e_3) + L_2(e_3 \wedge e_1) + L_3(e_1 \wedge e_2)$. The rotation around a general point and axis is given by conjugation with an element $TR\tilde{T}$. A general composition of a translator with a rotor is called a motor.

Remark 2.1. For some applications it is sufficient to consider an embedding in a 4D projective space only. To generate three dimensional projective geometric algebra (PGA), Euclidean vectors are embedded in an four dimensional affine space:

$$p(x) = xe_1 + ye_2 + ze_3 + e_0$$

where the e_i are the canonical basis of $\mathbb{R}^4$. If $a \in \mathbb{R}^3$, then the corresponding homogenized vector $p(a)$ is denoted as A and given vector $A \in \mathbb{G}_4$, the corresponding Euclidean vector is

$$a = \frac{A}{A \cdot e_0} - e_0.$$

Using the concept of direct representation the bivector $A \wedge B$ represents the line trough a and b (after homogenisation), where $A = p(a), B = p(b)$ are the points. Finally, the outer product of three homogeneous vectors in $\mathbb{G}_3$ represents a plane in the direct representation. Note that this PGA approach is implicitly contained in the CGA approach.

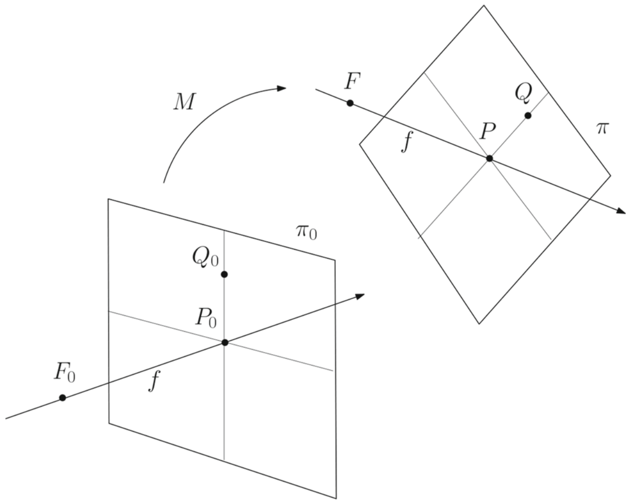


Figure 1. Description of camera position

3. Camera Position

First of all, we need to effectively a position of a camera in space. We represent the camera by two CGA elements:

- the conformal point F representing the camera center (focus),
- the point pair $P \wedge Q$ defining the camera image plane π,

as illustrated in Fig. 1.

We obtain the classical camera description by straightforward computations as

- focal distance $f = -2\sqrt{F \cdot P}$,
- camera axis $(F - P) \wedge e_\infty$,
- camera plane $\pi = P \wedge Q \wedge (F \wedge P \wedge e_\infty)^*$.

Note that in our settings the line $P \wedge Q \wedge e_\infty$ is always orthogonal to the line $F \wedge P \wedge e_\infty$, i.e.

$$(F \wedge P \wedge e_\infty) \cdot (P \wedge Q \wedge e_\infty) = 0.$$

Given an initial position F_0, P_0, Q_0, the actual position in the space is obtained by an Euclidean transformation which in CGA is given by a conjugation with a motor M, see Fig. 1 again. The advantage of CGA is that we can transform whole geometric objects, i.e. any geometric entity constructed from F, P, Q. For instance, the actual position of the camera center is

$$F = MF_0\tilde{M}, \tag{1}$$

and the actual position of the image plane is given by

$$\pi = M\pi_0\tilde{M}. \tag{2}$$

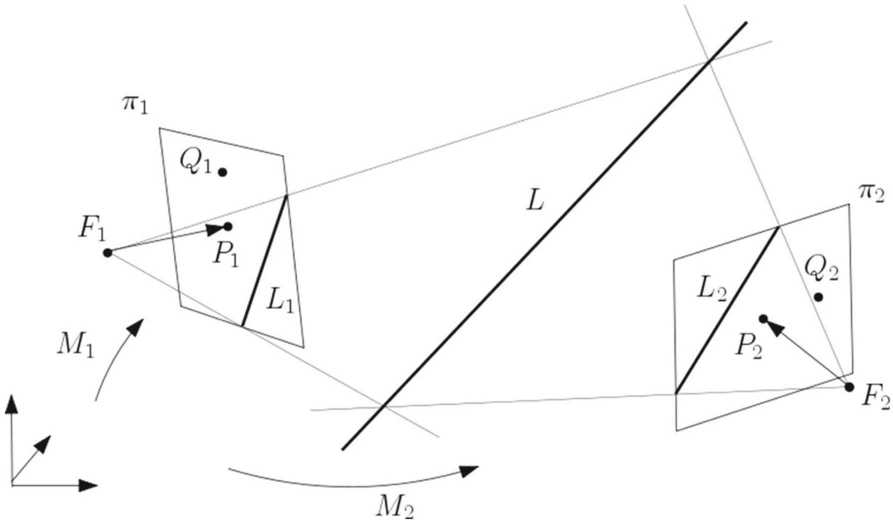


Figure 2. A line projected to two cameras

In the same way, one can obtain also the actual direction of the camera $M((F_0 - P_0) \wedge e_\infty)\tilde{M}$. For a concrete application one only needs to specify the initial position and the motor M.

Remark 3.1. In PGA, the camera position is also represented by two objects; the projective point F representing the camera center and line $P \wedge Q$ which lies in camera plane such that the line $P \wedge Q$ is orthogonal to line $F \wedge P$, i.e. $(F \wedge P) \cdot (P \wedge Q) = 0$. Similarly to the formula in CGA, the camera image plane is obtained as $\pi = P \wedge Q \wedge (F \wedge P)^*$.

4. Pose Estimation

Let us consider two cameras in a general position, given by two arbitrary motors M_1, M_2, and a line L which projects to conjugate lines L_1, L_2 in the two camera image planes π_1, π_2, see Fig. 2.

We show how easy it is, using CGA, to find the images L_1 and L_2 of L or to reconstruct L from those images. Indeed, given a line L in 3D, its image on a camera plane is given as the intersection of this plane with a plane spanned by L and the corresponding camera focus. In terms of the wedge product and the meet we thus have

$$L_k = (L \wedge F_k) \vee \pi_k, \quad k = 1, 2, \tag{3}$$

where the current positions F_k and π_k are computed from their initial position by (1) and (2), respectively, with M being either M_1 or M_2. Concerning the inverse problem, the real line is reconstructed from the given images as the intersection of the plane spanned by L_1 and F_1 with the plane spanned by L_2 and F_2. Thus its representation in CGA reads

$$L = (F_1 \wedge L_1) \vee (F_2 \wedge L_2). \tag{4}$$

Let us emphasize that formulas (3), (4) are valid for cameras in general positions based on arbitrary motors M_1 and M_2 and that the computations above are valid for both conformal geometric algebra and also for the projective geometric algebra.

Note that the conjugate lines L_1 and L_2 are considered as lines in 3D space. For applications we would rather need to express them in the 2D image plane. Therefore, we define maps $\iota_k : \mathbb{R}^2 \to \mathbb{R}^3$ which transform the xy-plane into π_k, $k = 1, 2$. Then a formula for L_k in the image plane geometric algebra is given by

$$L_k^{2\mathrm{D}} := \delta_k e_1 \wedge e_2 \wedge e_\infty + \cos(\alpha_k) e_1 \wedge e_\infty \wedge e_0 + \sin(\alpha_k) e_2 \wedge e_\infty \wedge e_0,$$

for a suitable planar Plücker coordinates based on distance from the origin δ_k and the slope $\tan(\alpha_k)$. These can be extracted from L_k as follows

$$\begin{aligned} \vec{L}_k &= (e_0 \wedge e_\infty) \cdot L_k^{2\mathrm{D}}, \\ (\delta_k)^* &= (L_k^{2\mathrm{D}} - \vec{L}_k \wedge e_0 \wedge e_\infty) \wedge e_3 \wedge e_0, \end{aligned}$$

where $\vec{L}_k \equiv \cos(\alpha_k) e_1 + \sin(\alpha_k) e_2$. Indeed, compute $(e_0 \wedge e_\infty) \cdot L_k^{2\mathrm{D}}$ as

$$\begin{aligned} &\langle (e_0 \wedge e_\infty) \cdot L_k^{2\mathrm{D}} \rangle_{3-2} \\ &\quad = \langle \delta_k (e_0 \wedge e_\infty)(e_1 \wedge e_2 \wedge e_\infty) + \cos(\alpha_k)(e_0 \wedge e_\infty)(e_1 \wedge e_\infty \wedge e_0) \\ &\qquad + \sin(\alpha_k)(e_0 \wedge e_\infty)(e_2 \wedge e_\infty \wedge e_0) \rangle_1 \\ &\quad = \langle \delta_k (-1 - e_0 e_\infty)(e_1 e_2 e_\infty) + \cos(\alpha_k)(-1 - e_0 e_\infty)(e_1(1 + e_\infty e_0)) \\ &\qquad + \sin(\alpha_k)(-1 - e_0 e_\infty)(e_2(1 + e_\infty e_0)) \rangle_1 \\ &\quad = \langle -\delta_k (e_1 e_2 e_\infty) + \cos(\alpha_k)(-e_1(1 + e_\infty e_0) - e_0 e_\infty e_1) \\ &\qquad + \sin(\alpha_k)(-e_2(1 + e_\infty e_0) - e_0 e_\infty e_2) \rangle_1 \\ &\quad = \langle -\delta_k (e_1 e_2 e_\infty) + \cos(\alpha_k)(-e_1 - e_1 e_\infty e_0 - e_0 e_\infty e_1) \\ &\qquad + \sin(\alpha_k)(-e_2 - e_2 e_\infty e_0 - e_0 e_\infty e_2) \rangle_1 \\ &\quad = -\cos(\alpha_k) e_1 - \sin(\alpha_k) e_2. \end{aligned}$$

In the very similar way compute $(L_k^{2\mathrm{D}} - \vec{L}_k \wedge e_0 \wedge e_\infty)$ as

$$\begin{aligned} &\delta_k e_1 \wedge e_2 \wedge e_\infty + \cos(\alpha_k) e_1 \wedge e_\infty \wedge e_0 + \sin(\alpha_k) e_2 \wedge e_\infty \wedge e_0 + (\cos(\alpha_k) e_1 \\ &\qquad + \sin(\alpha_k) e_2) \wedge e_0 \wedge e_\infty \\ &\quad = \delta_k e_1 \wedge e_2 \wedge e_\infty \end{aligned}$$

and finally,

$$(L_k^{2\mathrm{D}} - \vec{L}_k \wedge e_0 \wedge e_\infty) \wedge e_3 \wedge e_0 = \delta_k e_1 \wedge e_2 \wedge e_\infty \wedge e_3 \wedge e_0 = (\delta_k)^*.$$

Let us remark that in the inverse problem, we do not need to read off the Plücker coordinates directly. If we read off two points $[x_k^1, y_k^1]$, $[x_k^2, y_k^2]$ of the line L_k instead, then we get

$$\begin{aligned} L_k^{2\mathrm{D}} = (x_k^1 y_k^2 - x_k^2 y_k^1) e_1 \wedge e_2 \wedge e_\infty + (x_k^2 - x_k^1) e_1 \wedge e_\infty \wedge e_0 \\ + (y_k^2 - y_k^1) e_2 \wedge e_\infty \wedge e_0. \end{aligned}$$

Remark 4.1. In PGA, the geometric elements $F_1, F_2, L_1, L_2, L, \pi_1, \pi_2$ are represented differently (in a 4D space) but the fundamental formulae (3) and (4) have exactly the same form.

5. Models and the Problem

We will demonstrate our setup on two models; each of them is composed of two pinhole cameras but their relative position and motion freedom is different. The problem is to adapt the given system by changing free parameters (angles of rotations) so that the image of a given line in one camera or in both cameras (if possible) is in a specific position. Concretely, in the examples below we want to identify the projection of the line with the x-axis of the image coordinates. This problem in general is solved completely by formulas (3) and (4). An explicit solution is found by the following algorithm based on these formulas. We consider that the configuration of a given system depends on a set of parameters $\boldsymbol{\phi} = (\phi_1, \phi_2, \dots)$ and our aim is to find $\boldsymbol{\phi}$ such that the image of a line in the k-th camera coincides with a given reference line L_k^{ref} (x-axis in the examples below). Note that the algorithm has two parts. In the first one, we reconstruct the space line L from its images captured by the camera system in a configuration $\boldsymbol{\phi}^0$. The second part computes images of L in a general configuration $\boldsymbol{\phi}$.

- identify the system by specifying two motors M_1, M_2 and the map ι_k and their dependance on parameters $\boldsymbol{\phi}$,
- input two 2D images $L_1^{2\mathrm{D}}, L_2^{2\mathrm{D}}$ of a real line L captured on cameras in a positions described by parameters $\boldsymbol{\phi}^0$,
- compute $L_k(\boldsymbol{\phi}^0) = \iota_k(L_k^{2\mathrm{D}})$ for $k = 1, 2$,
- compute positions of camera centers for $k = 1, 2$ according to (1)
$$F_k(\boldsymbol{\phi}^0) = M_k(\boldsymbol{\phi}^0) F_0 \tilde{M}_k(\boldsymbol{\phi}^0),$$
- compute the real line L according to (4)
$$L = (F_1(\boldsymbol{\phi}^0) \wedge L_1(\boldsymbol{\phi}^0)) \vee (F_2(\boldsymbol{\phi}^0) \wedge L_2(\boldsymbol{\phi}^0))$$
- compute camera centers and image planes in a position $\boldsymbol{\phi}$ according to (1) and (2)
$$\begin{aligned} F_k(\boldsymbol{\phi}) &= M_k(\boldsymbol{\phi}) F_0 \tilde{M}_k(\boldsymbol{\phi}), \\ \pi_k(\boldsymbol{\phi}) &= M_k(\boldsymbol{\phi}) \pi_0 \tilde{M}_k(\boldsymbol{\phi}), \end{aligned}$$
- compute the images the line L in cameras in a position $\boldsymbol{\phi}$ according to (3)
$$L_k(\boldsymbol{\phi}) = (L \wedge F_k(\boldsymbol{\phi})) \vee \pi_k(\boldsymbol{\phi}),$$
- solve equation $L_k^{2\mathrm{D}}(\boldsymbol{\phi}) = L_k^{\mathrm{ref}}$ for $\boldsymbol{\phi}$.

For both models we consider that the camera initial position is in the Cartesian frame such that

$$\begin{aligned} F_0 &= c(0,0,0) = e_0, \\ P_0 &= c(0,0,f) = f e_3 + \frac{1}{2} f^2 e_\infty + e_0, \\ Q_0 &= c(0,1,f) = e_2 + f e_3 + \frac{1}{2}(f^2+1) e_\infty + e_0. \end{aligned}$$

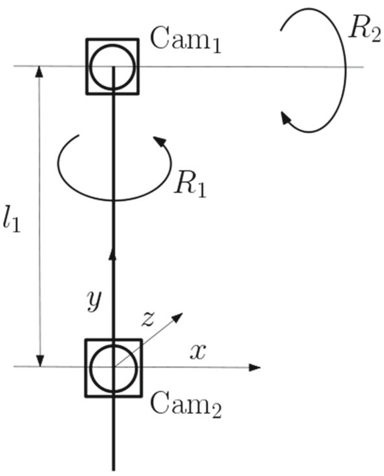


Figure 3. Scheme of the first model

Consequently, the initial camera axis $F_0 \wedge P_0 \wedge e_\infty = fe_3 \wedge e_\infty \wedge e_0$ is the z-axis and the initial image plane $\pi_0 = P_0 \wedge Q_0 \wedge (F_0 \wedge P_0 \wedge e_\infty)^*$ is given by

$$\pi_0 = -\frac{1}{2}f^2 e_1 \wedge e_2 \wedge e_3 \wedge e_\infty + \frac{1}{2}f e_1 \wedge e_2 \wedge e_\infty \wedge e_0.$$

The map $\iota_k : \mathbb{R}^2 \to \mathbb{R}^3$ maps xy-plane to π_0 (the translation about the focal distance f in the z-direction) and then to π_k (transformation M_k) and thus is given by conjugation by the motor $M_k \exp(-1/2 f e_3 \wedge e_\infty)$, $k = 1, 2$, i.e.

$$\iota_k(L_k^{2\mathrm{D}}) = M_k \left(1 - \frac{f}{2} e_3 \wedge e_\infty\right) L_k^{2\mathrm{D}} \left(1 + \frac{f}{2} e_3 \wedge e_\infty\right) \tilde{M}_k.$$

Since it is an orthogonal transformation in CGA, its inverse is given simply by a conjugation by the reversed motor. Now we have all what we need to solve the problem in CGA, except the specification of M_1 and M_2.

5.1. First Model

Let us consider a system of two cameras sharing one rotation axis such that the first of them can rotate also around a perpendicular axis. A precise scheme with an attached coordinate frame is displayed in Fig. 3.

The formulae for the motors can be read off the figure directly. Namely, we have

$$M_1 = R_2 R_1 T_1,$$
$$M_2 = R_1,$$

where the translation T_1 and the rotations R_1, R_2 are given by

$$\begin{aligned}
T_1 &= 1 - \frac{1}{2} l_1 e_2 \wedge e_\infty, \\
R_1 &= \cos\left(\frac{\phi_1}{2}\right) + \sin\left(\frac{\phi_1}{2}\right)(e_3 \wedge e_1), \\
R_2 &= \cos\left(\frac{\phi_2}{2}\right) + \sin\left(\frac{\phi_2}{2}\right)\ell_2
\end{aligned}$$

and where the axis ℓ_2 of the second rotation is

$$\ell_2 = R_1 T_1 (e_2 \wedge e_3) \tilde{T}_1 \tilde{R}_1.$$

Thus the appropriate motors are determined and a solution can be found according to the algorithm given above. In this particular example, it is possible to find a complete solution in symbolic form using CLIFFORD [1]. Namely, given Plücker coordinates $(m_1, m_2, m_3, d_1, d_2, d_3)$ of the world line L, its images in cameras are

$$\begin{aligned}
\vec{L}_1 &= \Big(m_2 \cos\phi_2 - (m_1 - l_1 d_3) \sin\phi_1 \sin\phi_2 + (m_3 + l_1 d_1) \cos\phi_1 \sin\phi_2\Big) e_1 \\
&\quad + \Big((m_1 - l_1 d_3) \cos\phi_1 + (m_3 + l_1 d_1) \sin\phi_1\Big) e_2 \\
\delta_1 &= f(m_1 - l_1 d_3) \sin\phi_1 \cos\phi_2 - f(m_3 + l_1 d_1) \cos\phi_1 \cos\phi_2 \\
\vec{L}_2 &= m_2 e_1 + (m_1 \cos\phi_1 + m_3 \cos\phi_1) e_2 \\
\delta_2 &= f m_1 \sin\phi_1 - f m_3 \cos\phi_1
\end{aligned}$$

On the other hand, given Plücker coordinates (x_1, y_1, δ_1) (x_2, y_2, δ_2) of L_1 and L_2, respectively, the real line L is reconstructed as

$$\begin{aligned}
m_1 &= l_1 x_2 y_1 f^2 \sin\phi_1 \cos\phi_2 - l_1 x_2 \delta_2 f \cos\phi_1 \cos\phi_2 + l_1 y_1 \delta_1 f \sin\phi_1 \sin\phi_2 \\
&\quad - l_1 \delta_1 \delta_2 \cos\phi_1 \sin\phi_2, \\
m_2 &= l_1 x_1 x_2 f^2 \cos\phi_2 + l_1 x_1 \delta_1 f \sin\phi_2, \\
m_3 &= l_1 x_2 y_1 f^2 \cos\phi_1 \cos\phi_2 + l_1 x_2 \delta_2 f \sin\phi_1 \cos\phi_2 + l_1 y_1 \delta_1 f \cos\phi_1 \sin\phi_2 \\
&\quad + l_1 \delta_1 \delta_2 \sin\phi_1 \sin\phi_2, \\
d_1 &= -y_2 \delta_1 f \sin\phi_1 \sin\phi_2 + x_2 y_1 f^2 \sin\phi_1 \cos\phi_2, \\
&\quad -(\delta_1 \delta_2 + x_2 y_1 f^2) \cos\phi_1 \sin\phi_2 + (x_1 \delta_1 f - x_2 \delta_2 f) \cos\phi_1 \cos\phi_2, \\
d_2 &= -y_2 \delta_2 f - x_2 y_1 f^2 \sin\phi_2 + y_1 \delta_1 f \cos\phi_2, \\
d_3 &= (\delta_1 \delta_2 + x_1 x_2) \sin\phi_1 \sin\phi_2 + (x_1 \delta_1 f - x_2 \delta_2 f) \sin\phi_1 \cos\phi_2, \\
&\quad - y_1 \delta_1 f \cos\phi_1 \sin\phi_2 - x_2 y_1 f^2 \cos\phi_1 \cos\phi_2.
\end{aligned}$$

Now we can solve our main equation

$$L_k^{\mathrm{2D}}(\boldsymbol{\phi}) = L_k^{\mathrm{ref}}$$

for $\boldsymbol{\phi}$ explicitly. To present a set of results we choose the kinematic parameters $\ell_1 = 1$ and $f = 0.045$. Furthermore, we assume that the initial camera positions are determined by the angles $\phi_1^0 = 0$ and $\phi_2^0 = -\frac{\pi}{8}$. If the Plücker coordinates coincide with this setting we conclude that the visualized line is placed in front of the system. We set the angle to be $\frac{\pi}{10}$, $\frac{\pi}{20}$ and $\frac{\pi}{100}$,

Table 2. The results where the image of a line in the camera coincides with x-axis

$\phi_{1,2}$	$d_{1,2}$	First model: r_1	r_2	Second model: r_1	r_2
$\pi/10$	0.1	−3.01	1.56	0.14	4.31
	0.01	−2.64	0.68	0.95	0.38
$\pi/20$	0.1	−3.08	1.55	0.07	1.15
	0.01	−2.88	0.63	0.61	0.27
$\pi/100$	0.1	−3.13	1.54	0.01	1.15
	0.01	−3.09	0.61	0.14	0.22

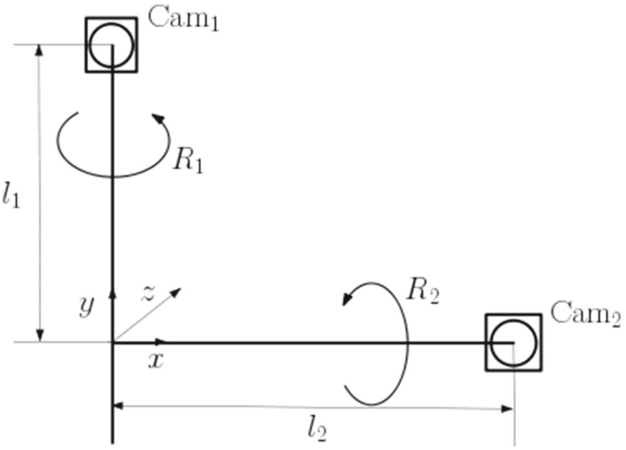


Figure 4. Scheme of the second model

respectively, and the appropriate distance to be 0.1 and 0.01. The numeric results are shown in columns three to four of Table 2.

The image within camera number one can be placed into Plücker plane coordinates $(1, 0, 0)$ by a combination of both rotations. The larger the object distance, the rotation changes become less important. Clearly, the data in the above table correspond to the fact that if $d_{1,2}$ is decreased, the visualized object is becomes more remote.

5.2. Second Model

Second model, see Fig. 4, is also based on two revolute joints, represented by rotations R_1, R_2 but the cameras do not lie on one axis.

In this case, the system can be described by the following set of motors.

$$M_1 = R_1T_1,$$
$$M_2 = R_2R_1T_2,$$

where the translations T_1, T_2 and the rotations R_1, R_2 are given by

$$\begin{aligned} T_1 &= 1 - \frac{1}{2} l_1 e_2 \wedge e_\infty, \\ T_2 &= 1 - \frac{1}{2} l_2 e_1 \wedge e_\infty, \\ R_1 &= \cos\left(\frac{\phi_1}{2}\right) + \sin\left(\frac{\phi_1}{2}\right)(e_3 \wedge e_1), \\ R_2 &= \cos\left(\frac{\phi_2}{2}\right) + \sin\left(\frac{\phi_2}{2}\right)\ell_2 \end{aligned}$$

and where the axis ℓ_2 of the second rotation is

$$\ell_2 = R_1(e_2 \wedge e_3)\tilde{R}_1.$$

Finally, we present an example given by the kinematic setting $\ell_1 = 1, \ell_2 = 0.5$ and $f = 0.045$. The initial configuration is given by $\phi_1^0 = \phi_2^0 = 0$ and the Plücker coordinates are the same as in the first model. The numeric results are shown in columns five to six of Table 2. The data in the above table are in agreement with the fact that decreasing $d_{1,2}$ correlates with increasing the distance of the visualized object.

6. Human-Like Vision

The models described above can be solved alternatively in PGA since all objects which occurred were flat, i.e. points, lines and planes, see Remarks 3.1 and 4.1. However, the CGA approach allows generalizations human-like vision. By this expression, we mean a binocular vision as above but with cameras which do not project on plane but rather on a sphere. Thus one case is similar to human vision but the relative position of "eyes" can vary and may depend on several parameters. For instance, the "eyes" can rotate as in the models above. In CGA, sphere is in principle the same object as plane and thus we can use the theory from Sects. 3, 4 and also the algorithm from Sect. 5 with minor changes. Instead of a camera plane π, we have a camera sphere σ but the transformation is the same as in (2), i.e.

$$\sigma = M\sigma_0\tilde{M}.$$

The role of F is the same and P defines the direction of an "eye". Instead of Q one needs more data to recover the whole image sphere σ. The basic formulas (3) and (4) are also still valid but as the images of L we get conjugate circles c_1, c_2 instead of conjugate lines L_1, L_2, i.e.

$$\begin{aligned} c_k &= (L \wedge F_k) \vee \sigma_k, \quad k = 1, 2, \\ L &= (F_1 \wedge c_1) \vee (F_2 \wedge c_2). \end{aligned}$$

Note that the images c_1, c_2, even in their initial position, are 3D objects. A possibility for how to get an equivalent 2D information is to take their stereographic projections, an operation which is easy with CGA. Using CGA one can also easily recover classical notions as the point of fixation or the horopter. Of course, the fixation does not need to exist in our general setting. But if it exists, it is obviously given by

$$\mathrm{Fix} = (F_1 \wedge P_1 \wedge e_\infty) \vee (F_2 \wedge P_2 \wedge e_\infty)$$

and the horopter circle reads

$$\mathrm{Hor} = F_1 \wedge F_2 \wedge Fix.$$

7. Conclusion

The motivation for this paper is given by a specific engineering application of binocular vision. More precisely, we consider two cameras attached to a mechanic manipulator on different axes. The whole construction provides several degrees of freedom. The initial setting reads the position of both cameras and the reference line projection of the observed object. The goal is to identify the reference line and center it on an arbitrary camera. Note that without the reference line is exact 3D position it is not possible to solve the problem, although this was not part of the setting.

We solved this problem for arbitrary manipulator kinematics by means of CGA objects and transformations. In Sect. 5 we introduced the algorithm for a particular kinematics choice and derived the explicit equation $L_k^{2\mathrm{D}}(\phi) = L_k^{ref}$, the solution of which is the desired configuration. We solved the problem for two specific configurations and included the presentation of several outputs. The final of the first example (Sect. 5.1) are presented in the form without the CGA symbolic which we find suitable for. Conformal algebra thus provides an effective system description. Its object oriented nature then allows to solve several problems simultaneously. The last step demands to solve the system of non linear equations. This should be done with special software that accepts the classical form of equations. Note that even the system transformation into the explicit form is accomplished with CGA manipulations.

Within the process we needed to describe the inner configuration of a pin-hole camera, which was done in Sect. 3. Furthermore, some unknown identities transforming the CGA line description into the Plücker coordinates were introduced, see Sect. 4.

References

[1] Ablamowicz R., FauserB.: CLIFFORD/Bigebra, a Maple package for Clifford (Co)algebra computations. http://www.math.tntech.edu/rafal/ (2015)

[2] Dorst, L., Fontijne, D., Mann, S.: Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry. Morgan Kaufmann Publishers Inc., Burlington (2007)

[3] Gonzalez-Jimenez, L., Carbajal-Espinosa, O., Loukianov, A., Bayro-Corrochano, E.: Robust pose control of robot manipulators using conformal geometric algebra. Adv. App. Clifford Algebras **24**(2), 533–552 (2014)

[4] Hildenbrand, D.: Foundations of Geometric Algebra Computing. Springer Science and Business Media, Berlin (2013)

[5] Hrdina, J., Návrat, A., Vašík, P.: 3-link robotic snake control based on CGA. Adv. Appl. Clifford Algebras **26**(3), 1069–1080 (2016)

[6] Lounesto, P.: Clifford Algebra and Spinors, 2nd edn. CUP, Cambridge (2006)

[7] Perwass, C.: Geometric Algebra with Applications in Engineering. Springer, Berlin (2009)

[8] Zamora-Esquivel, J., Bayro-Corrochano, E.: Kinematics and differential kinematics of binocular robot heads. Proceedings 2006 IEEE International Conference on Robotics and Automation, pp. 4130–4135 (2006)

Jaroslav Hrdina and Aleš Návrat
Faculty of Mechanical Engineering
Institute of Mathematics
Brno University of Technology
Brno
Czech Republic
e-mail: `hrdina@fme.vutbr.cz`

Aleš Návrat
e-mail: `navrat.a@fme.vutbr.cz`

Received: October 7, 2016.
Accepted: February 9, 2017.

Appendix 4

[24] Hrdina, J., Návrat, A., Vašík, P.: *Geometric Algebra for Conics*, Adv. Appl. Clifford Algebras (2018) 28: 66. https://doi.org/10.1007/s00006-018-0879-2

Adv. Appl. Clifford Algebras (2018) 28:66
© 2018 Springer International Publishing AG,
part of Springer Nature
0188-7009/030001-21
published online June 30, 2018
https://doi.org/10.1007/s00006-018-0879-2

Advances in
Applied Clifford Algebras

Geometric Algebra for Conics

Jaroslav Hrdina*, Aleš Návrat and Petr Vašík

Abstract. We present a particular geometric algebra together with such an embedding of two–dimensional Euclidean space that the algebra elements may be in the most efficient way interpreted as arbitrary conic sections. Consequently, in this setting we provide full description of the conic sections analysis, classification and their transformations. Examples that show the functionality and consistency are provided in Maple together with the source code.

Mathematics Subject Classification. Primary 15A66, Secondary 51N25.

Keywords. Conformal geometric algebra, Geometric algebra, Clifford algebra, Conic section.

1. Introduction

The contribution of geometric algebras in various applications is well known, see e.g. [4,6,7]. The fundamental idea is the algebraic representation of both geometric objects and their Euclidean transformations as single Clifford algebra elements where the actual transformation is realized by means of the algebra operation, see e.g. [5] for computational concepts. Let us recall the list of essential geometric algebras suitable for various Euclidean plane representations. By Euclidean plane representation we mean an embedding of $\mathbb{R}^2$ into $\mathbb{R}^{p,q}$ generating a geometric algebra $\mathbb{G}_{p,q}$ which is equivariant with respect to Euclidean transformations. The simplest case of the plane representation is $\mathbb{G}_2$ but it covers the rotations only, not translations. Its natural extension $\mathbb{G}_{3,1}$ contains not only Euclidean transformations but even conformal transformations (i.e. similarities) and yet the appropriate Clifford algebra remains non-degenerate in spite of the case of e.g. $\mathbb{G}_{2,0,1}$. Another contribution of

This article is part of the Topical Collection on Proceedings ICCA 11, Ghent, 2017, edited by Hennie De Schepper, Fred Brackx, Joris van der Jeugt, Frank Sommen, and Hendrik De Bie

The research was supported by the Czech Science Foundation under Grant no.: 17-21360S.

*Corresponding author.

Birkhäuser

$\mathbb{G}_{3,1}$ lies in the linearization of Euclidean metric and efficient computations with spheres. Our aim is to find and describe a suitable representation of the Euclidean plane that provides the same efficiency for calculations with arbitrary conic sections. We continue in C. Perwass's concept, [9], precisely that of $\mathbb{G}_{5,3}$ and completes its description by the full classification of the elements corresponding to the appropriate conic sections with full advantages of geometric algebra approach, e.g. presence of Euclidean transformations as algebra elements with natural and well defined action. To learn about different approaches see Sect. 7.

Thus, in Sect. 2 we define the Geometric Algebra for Conics (GAC) as Clifford algebra $\mathbb{G}_{5,3}$ together with the particular embedding of two-dimensional Euclidean space. Consequently, we provide the inner product representations of all geometric entities available in GAC and show that these correspond to all possible conic sections and their intersections (Sect. 3), and their outer product representations together with a precise discussion of duality (Sect. 4). In Sect. 5 we describe the essential GAC transformations. We stress that, apart from translations elaborated in [9], we describe rotations and scaling and thus we receive a representation of similarities in the plane which enables efficient conic sections manipulations. In Sect. 6 we show the Maple implementation of GAC and test the functionality on a list of examples.

2. Geometric Algebra for Conics

The idea of C. Perwass is to generalize the concept of (two-dimensional) conformal geometric algebra $\mathbb{G}_{3,1}$. In the usual basis $\bar{n}, e_1, e_2, n$, embedding of a plane in $\mathbb{G}_{3,1}$ is given by

$$(x, y) \mapsto \bar{n} + xe_1 + ye_2 + \frac{1}{2}(x^2 + y^2)n.$$

Let us recall that $\bar{n}, n$ are null-vectors which play the role of the origin and infinity, respectively, [9]. Hence the objects representable by vectors in $\mathbb{G}_{3,1}$ are linear combinations of $1, x, y, x^2 + y^2$, i.e. circles, lines, point pairs and points. If we want to cover also general conics, we need to add two terms: $\frac{1}{2}(x^2 - y^2)$ and xy. It turns out that we need two new infinities for that and also their two corresponding counterparts (Witt pairs), [8]. Thus the resulting dimension of the space generating the appropriate geometric algebra is eight.

Let $\mathbb{R}^{5,3}$ denote the eight-dimensional real coordinate space $\mathbb{R}^8$ equipped with a non-degenerate symmetric bilinear form of signature $(5, 3)$. The form defines Clifford algebra $\mathbb{G}_{5,3}$ and this is the Geometric Algebra for Conics in the algebraic sense. To add the geometric meaning we have to describe an embedding of the plane into $\mathbb{R}^{5,3}$. To do so, let us choose a basis of $\mathbb{R}^{5,3}$ such that the corresponding bilinear form is

$$B = \begin{pmatrix} 0 & 0 & -1_{3\times 3} \\ 0 & 1_{2\times 2} & 0 \\ -1_{3\times 3} & 0 & 0 \end{pmatrix}, \tag{1}$$

where $1_{2\times 2}$ and $1_{3\times 3}$ denote unit matrices of the displayed size. Analogously to CGA and to the notation in [9], we denote the corresponding basis elements as follows

$$\bar{n}_+, \bar{n}_-, \bar{n}_\times, e_1, e_2, n_+, n_-, n_\times.$$

The form of (1) suggests that the basis elements e_1, e_2 will play the usual role of standard basis of the plane while the null vectors $\bar{n}$, n will represent either the origin or the infinity. Note that there are three orthogonal 'origins' $\bar{n}$ and three corresponding orthogonal 'infinities' n. In terms of this basis, a point of the plane $\mathbf{x} \in \mathbb{R}^2$ defined by $\mathbf{x} = xe_1 + ye_2$ is embedded using the operator $C : \mathbb{R}^2 \to \mathcal{C} \subset \mathbb{R}^{5,3}$, which is defined by

$$C(x, y) = \bar{n}_+ + xe_1 + ye_2 + \frac{1}{2}(x^2 + y^2)n_+ + \frac{1}{2}(x^2 - y^2)n_- + xyn_\times. \quad (2)$$

The image $\mathcal{C}$ of the plane in $\mathbb{R}^{5,3}$ is an analogue of the conformal cone. In fact, it is a two-dimensional real projective variety determined by five homogenous polynomials of degree one and two.

Definition 2.1. Geometric Algebra for Conics (GAC) is the Clifford algebra $\mathbb{G}_{5,3}$ together with the embedding $\mathbb{R}^2 \to \mathbb{R}^{5,3}$ given by (2) in the basis determined by matrix (1).

Remark 2.2. The definition of GAC is analogous to the definition proposed by Perwass in [9]. The relation between the corresponding 'origins' and 'infinities' is $\bar{n}_{1,2} = \frac{1}{2}(\bar{n}_+ \pm \bar{n}_-)$ and $n_{1,2} = n_+ \pm n_-$, respectively.

Note that, up to the last two terms, the embedding (2) is the embedding of the plane into the two-dimensional conformal geometric algebra $\mathbb{G}_{3,1}$. In particular, it is evident that the scalar product of two embedded points is the same as in $\mathbb{G}_{3,1}$, i.e. for two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$ we have

$$C(\mathbf{x}) \cdot C(\mathbf{y}) = -\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2, \quad (3)$$

where the standard Euclidean norm is considered on the right hand side. In particular, each point is represented by a null vector.

Remark 2.3. The geometric algebra $\mathbb{G}_{3,1}$ is included in GAC. In terms of the usual basis, the inclusion is given by the identity on e_1, e_2 and by $\bar{n} \to \bar{n}_+$ and $n \to n_+$ on the null vectors. Also the two-dimensional version of the quadric geometric algebra (QGA) proposed in [10] is included in GAC. Here the inclusion is given by $e_0 \to \bar{n}_+$, $e_{\infty x} \to n_+ + n_-$ and $e_{\infty y} \to n_+ - n_-$.

Let us recall that the invertible algebra elements are called versors and they form a group, the Clifford group, and that conjugations with versors give transformations intrinsic to the algebra. Namely, if the conjugation with a $\mathbb{G}_{5,3}$ versor R preserves the 'cone' $\mathcal{C}$, i.e. for each $\mathbf{x} \in \mathbb{R}^2$ there exists such a point $\bar{\mathbf{x}} \in \mathbb{R}^2$ that

$$RC(\mathbf{x})\tilde{R} = C(\bar{\mathbf{x}}), \quad (4)$$

where $\tilde{R}$ is the reverse of R, then $\mathbf{x} \to \bar{\mathbf{x}}$ induces a transformation $\mathbb{R}^2 \to \mathbb{R}^2$ which is intrinsic to GAC. We will show in Sect. 5 that the conformal transformations are intrinsic to GAC.

Let us also recall the outer (wedge) product, inner product and the duality $A^* = AI^{-1}$. Henceforth we use the usual definitions as in [9]. Note that in GAC the pseudoscalar is given by

$$I = \bar{n}_+\bar{n}_-\bar{n}_\times e_1 e_2 n_+ n_- n_\times.$$

3. Inner Product Representation Of Conics

Let us recall the definition of inner product representation. An element $A_I \in \mathbb{G}_{5,3}$ is the inner product representation of a geometric entity A in the plane if and only if $A = \{\mathbf{x} \in \mathbb{R}^2 : C(\mathbf{x}) \cdot A_I = 0\}$. Hence, given a fixed geometric algebra, the representable objects can be found by examining the inner product of a vector and an embedded point. A general vector in the conic space $\mathbb{R}^{5,3}$ in terms of our basis is of the form

$$v = \bar{v}^+\bar{n}_+ + \bar{v}^-\bar{n}_- + \bar{v}^\times\bar{n}_\times + v^1 e_1 + v^2 e_2 + v^+ n_+ + v^- n_- + v^\times n_\times$$

and its inner product with an embedded point is then given by

$$C(x, y) \cdot v = -\frac{1}{2}(\bar{v}^+ + \bar{v}^-)x^2 - \bar{v}^\times xy - \frac{1}{2}(\bar{v}^+ - \bar{v}^-)y^2 + v^1 x + v^2 y - v^+,$$

i.e. by a general polynomial of degree two. Thus the objects representable in GAC are exactly conics. We also see that the two-dimensional subspace generated by infinities $n_-, n_\times$ is orthogonal to all embedded points. Hence a conic is uniquely represented (in homogeneous sense) by a vector in $\mathbb{R}^{5,3}$ modulo this subspace. This gives the desired dimension six. In other words, the inner representation of a conic in GAC can be defined as a vector

$$Q_I = \bar{v}^+\bar{n}_+ + \bar{v}^-\bar{n}_- + \bar{v}^\times\bar{n}_\times + v^1 e_1 + v^2 e_2 + v^+ n_+. \tag{5}$$

The classification of conics is well known. The non-degenerate conics are of three types, the ellipse, hyperbola, and parabola. Now, we present the vector form (5) appropriate to each conic type in the simplest case, i.e. an axes-aligned conic with its centre in the origin. The results may be verified easily by multiplying each vector by an embedded point which means the application of (1) and (2).

Example 3.1. In the canonical coordinate system, ellipse E_I and hyperbola H_I with semi-axis a, b and parabola P_I with semi-latis rectum p are represented by the following GAC vectors

$$E_I = (a^2 + b^2)\bar{n}_+ + (a^2 - b^2)\bar{n}_- - a^2 b^2 n_+, \tag{6}$$
$$H_I = (a^2 + b^2)\bar{n}_+ + (a^2 - b^2)\bar{n}_- + a^2 b^2 n_+, \tag{7}$$
$$P_I = \bar{n}_+ + \bar{n}_- + p e_2. \tag{8}$$

The following propositions specify the form of vector (5) for all types of conics in a general position depending on their internal parameters and the position and orientation in the plane.

Proposition 3.2. *An ellipse E with the semi-axes a, b centred in $(u, v) \in \mathbb{R}^2$ rotated by angle θ is in the GAC inner representation given by*

$$\begin{aligned} E_I = \bar{n}_+ &- (\alpha \cos 2\theta)\bar{n}_- - (\alpha \sin 2\theta)\bar{n}_\times \\ &+(u + u\alpha \cos 2\theta - v\alpha \sin 2\theta)e_1 + (v + v\alpha \cos 2\theta - u\alpha \sin 2\theta)e_2 \\ &+\tfrac{1}{2}\left(u^2 + v^2 - \beta - (u^2 - v^2)\alpha \cos 2\theta - 2uv\alpha \sin 2\theta\right) n_+, \end{aligned} \tag{9}$$

where

$$\alpha = \frac{a^2 - b^2}{a^2 + b^2}, \beta = \frac{2a^2b^2}{a^2 + b^2}.$$

A hyperbola H with the semi-axes a, b centred in $(u, v) \in \mathbb{R}^2$ rotated by angle θ is in the GAC inner representation given by the same expression but with

$$\alpha = \frac{a^2 + b^2}{a^2 - b^2}, \beta = \frac{-2a^2b^2}{a^2 - b^2}.$$

Proof. A direct computation gives

$$\begin{aligned} C(x, y) \cdot E_I = &- \tfrac{1}{2}(1 - \alpha \cos 2\theta)x^2 + \alpha \sin 2\theta xy - \tfrac{1}{2}(1 + \alpha \cos 2\theta)y^2 \\ &+ (u + u\alpha \cos 2\theta - v\alpha \sin 2\theta)x + (v + v\alpha \cos 2\theta - u\alpha \sin 2\theta)y \\ &- \tfrac{1}{2}\left(u^2 + v^2 - \beta - (u^2 - v^2)\alpha \cos 2\theta - 2uv\alpha \sin 2\theta\right) \\ = &- \frac{a^2b^2}{a^2 + b^2}\left(\frac{\bar{x}^2}{a^2} + \frac{\bar{y}^2}{b^2} - 1\right), \end{aligned}$$

where $\bar{x} = (x - u) \cos \theta + (y - v) \sin \theta$, $\bar{y} = -(x - u) \sin \theta + (y - v) \cos \theta$. The last equality follows from the double-angle formulae for trigonometric functions. Hence $(\bar{x}, \bar{y})$ are the canonical coordinates for the ellipse and these coordinates differ from the original ones by translation (u, v) and rotation by θ. The result for hyperbolas follows immediately since a hyperbola with semi axes a, b can be considered as an ellipse with a semi axis a and an imaginary semi axis b. □

Proposition 3.3. *A parabola P with the semi-latus rectum p centred in $(u, v) \in \mathbb{R}^2$ rotated by angle θ is in the GAC inner representation given by*

$$\begin{aligned} P_I = \bar{n}_+ &+ \cos 2\theta \bar{n}_- + \sin 2\theta \bar{n}_\times \\ &+(u + u \cos 2\theta + v \sin 2\theta - 2p \sin \theta)e_1 \\ &+(v - v \cos 2\theta + u \sin 2\theta + 2p \cos \theta)e_2 \\ &+\tfrac{1}{2}\left(u^2 + v^2 + (u^2 - v^2) \cos 2\theta + 2uv \sin 2\theta - 4pu \sin \theta + 4pv \cos \theta\right) n_+ \end{aligned} \tag{10}$$

Proof. A direct computation gives

$$\begin{aligned} C(x, y) \cdot P_I = &- \tfrac{1}{2}(1 + \cos 2\theta)x^2 - \sin 2\theta xy - \tfrac{1}{2}(1 - \cos 2\theta)y^2 \\ &+ (u + u \cos 2\theta + v \sin 2\theta - 2p \sin \theta)x \\ &+ (v - v \cos 2\theta + u \sin 2\theta + 2p \cos \theta)y \\ &- \tfrac{1}{2}\left(u^2+v^2+(u^2-v^2) \cos 2\theta+2uv \sin 2\theta-4pu \sin \theta+4pv \cos \theta\right) \\ =&2p\bar{y} - \bar{x}^2, \end{aligned}$$

where $\bar{x} = (x - u) \cos \theta + (y - v) \sin \theta$, $\bar{y} = -(x - u) \sin \theta + (y - v) \cos \theta$. □

Even though a circle is a special ellipse and thus its GAC representation is obtained from (9) for $\alpha = 0$, $\beta = \rho^2$, we give the GAC inner product representations of $\mathbb{G}_{3,1}$ elements as a separate proposition.

Proposition 3.4. *The GAC inner representations of $\mathbb{G}_{3,1}$ objects are given by the usual formulas. A circle C centred in (p_1, p_2) with radius ρ is given by*

$$C_I = \bar{n}_+ + p_1 e_1 + p_2 e_2 + \tfrac{1}{2}(p_1^2 + p_2^2) n_+ - \tfrac{1}{2}\rho^2 n_+$$

and a line L with unite normal (n_1, n_2) and a shift d from the origin is given by

$$L_I = n_1 e_1 + n_2 e_2 + d n_+.$$

Proof. One can directly compute again the inner product of C_I and L_I with $C(x, y)$ to prove the proposition. However, this is not necessary since by Remark 2.3 the algebra $\mathbb{G}_{3,1}$ is included in GAC and thus all $\mathbb{G}_{3,1}$ elements have the same form in GAC up to the subscript $+$. □

We complete the list of conic types by providing the algebra elements that represent two lines that intersect and two lines that do not intersect. Formulas for these degenerate conics can be easily derived from the non-degenerate ones by means of certain limits.

Proposition 3.5. *GAC inner representation of two parallel lines is given by (9) with coefficients $\alpha = -1$, $\beta = 2a^2$, where $2a$ is the distance between the lines. GAC inner representation of two intersecting lines, which are not perpendicular, is given by (9) with coefficients $\alpha = \frac{1+k^2}{1-k^2}$, $\beta = 0$, where k is the line derivation.*

Proof. Parallel lines are obtained from an ellipse by the limit $b \to \infty$. The intersecting lines are obtained from a hyperbola by setting $b = ka$ and then by taking the limit $a \to 0$. □

Remark 3.6. Two perpendicular lines cannot be expressed in the form (9) since the coefficient of $\bar{n}_+$ has to be zero. For this particular case we get the GAC inner representation of the form

$$\begin{aligned} LL_I^{\perp} = &-(\alpha \cos 2\theta)\bar{n}_- - (\sin 2\theta)\bar{n}_\times + (u \cos 2\theta - v \sin 2\theta) e_1 \\ &+ (v \cos 2\theta - u \sin 2\theta) e_2 - \tfrac{1}{2}\left((u^2 - v^2)\cos 2\theta + 2uv \sin 2\theta\right) n_+. \end{aligned}$$

It is well known that the type of a given unknown conic can be read off its matrix representation, which in our case for a conic given by vector (5) reads

$$Q = \begin{pmatrix} -\frac{1}{2}(\bar{v}^+ + \bar{v}^-) & -\frac{1}{2}\bar{v}^\times & \frac{1}{2}v^1 \\ -\frac{1}{2}\bar{v}^\times & -\frac{1}{2}(\bar{v}^+ - \bar{v}^-) & \frac{1}{2}v^2 \\ \frac{1}{2}v^1 & \frac{1}{2}v^2 & -v^+ \end{pmatrix}. \tag{11}$$

The entries of (11) can be easily computed by means of the inner product:

$$\begin{aligned}
q_{11} &= Q_I \cdot \tfrac{1}{2}(n_+ + n_-),\\
q_{22} &= Q_I \cdot \tfrac{1}{2}(n_+ - n_-),\\
q_{33} &= Q_I \cdot \bar{n}_+,\\
q_{12} = q_{21} &= Q_I \cdot \tfrac{1}{2}n_\times,\\
q_{13} = q_{31} &= Q_I \cdot \tfrac{1}{2}e_1,\\
q_{23} = q_{32} &= Q_I \cdot \tfrac{1}{2}e_2.
\end{aligned}$$

It is also well known how to determine the internal parameters of an unknown conic and its position and the orientation in the plane from the matrix (11). Hence all this can be determined from the GAC vector Q_I by means of the inner product.

The list of conics is complete though the list of the GAC objects is yet more extensive. More precisely, the objects that are representable in GAC are conics and their intersections, only. Intersection of two conics, hence the next basic objects, are point quadruplets. The properties of inner and outer product imply that intersections are given by the wedge product of inner representations. Indeed, given the inner representation of two different conics Q^1, Q^2 in GAC, we have

$$C(x,y) \cdot (Q_I^1 \wedge Q_I^2) = (C(x,y) \cdot Q_I^1)Q_I^2 - (C(x,y) \cdot Q_I^2)Q_I^1.$$

The vectors Q_I^1, Q_I^2 are linearly independent since we consider different conics and hence the above formula equals to zero if and only if $C(x,y) \cdot Q_I^1 = 0$ and $C(x,y) \cdot Q_I^2 = 0$. Hence the inner product representation of the intersection of two conics is given by

$$(Q^1 \cap Q^2)_I = Q_I^1 \wedge Q_I^2. \tag{12}$$

4. Outer Product Representation

Let us recall that the outer product representation is given as a null space of the wedge product. The duality between the outer and inner product implies

$$\{\mathbf{x} \in \mathbb{R}^2 : C(\mathbf{x}) \cdot A_I = 0\} = \{\mathbf{x} \in \mathbb{R}^2 : C(\mathbf{x}) \wedge A_I^* = 0\}.$$

Hence A_I^* is usually considered as the outer product representation of the entity A. However, A_I^* is always a multivector of the form $A_O \wedge \bar{n}_- \wedge \bar{n}_\times$ in GAC, where A_O is an element of grade five containing basis elements of the six-dimensional subspace (5) only. Thus it is convenient to consider rather A_O as the outer representation of A. In other words, A_O is the outer representation of an entity A if and only if

$$A = \{\mathbf{x} \in \mathbb{R}^2 : C(\mathbf{x}) \wedge A_O \wedge \bar{n}_- \wedge \bar{n}_\times = 0\}.$$

The duality between the outer and inner product representation of a conic then reads

$$A_O = (A_I \wedge n_- \wedge n_\times)^*, \tag{13}$$

$$A_I = (A_O \wedge \bar{n}_- \wedge \bar{n}_\times)^*. \tag{14}$$

Remark 4.1. Since GAC has 2^8 basis elements, the time load for the computation of the dual algebra element is enormous. For implementations it is more convenient to compute the dual representation using left contraction, i.e. inner product in this particular case. Namely, we define two new 'pseudoscalars'

$$I_{OI} = \bar{n}_+ \bar{n}_- \bar{n}_\times e_1 e_2 n_+, \tag{15}$$

$$I_{IO} = \bar{n}_+ e_1 e_2 n_+ n_- n_\times \tag{16}$$

and the duality between representations is given by the inner product with these elements, $A_O = A_I \cdot I_{IO}$ and $A_I = A_O \cdot I_{OI}$, depending on the direction.

This definition of the outer product representation allows a simple construction of conic sections by means of the outer product of points that lie on the conic section.

Proposition 4.2. *The outer product representation of a conic Q given by five points with GAC representatives $P_1, \ldots, P_5$ is*

$$Q_O = P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge P_5. \tag{17}$$

An axes-aligned conic Q^{al} given by four points with GAC representatives $P_1, \ldots, P_4$ is

$$Q_O^{al} = P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge n_\times. \tag{18}$$

A circle C given by three points with GAC representatives P_1, P_2, P_3 is

$$C_O = P_1 \wedge P_2 \wedge P_3 \wedge n_- \wedge n_\times. \tag{19}$$

A line L given by two points with GAC representatives P_1, P_2 is

$$L_O = P_1 \wedge P_2 \wedge n_+ \wedge n_- \wedge n_\times. \tag{20}$$

Proof. Let us start with a general conic. We prove an equivalent statement that $(Q_O \wedge \bar{n}_- \wedge \bar{n}_\times)^*$ is the inner representation of the conic Q defined by points $P_1, \ldots, P_5$. The duality between the inner and outer product implies

$$\bar{n}_{-,\times} \cdot (Q_O \wedge \bar{n}_- \wedge \bar{n}_\times)^* = (\bar{n}_{-,\times} \wedge Q_O \wedge \bar{n}_- \wedge \bar{n}_\times)^* = 0$$

and thus $(Q_O \wedge \bar{n}_- \wedge \bar{n}_\times)^*$ is a vector which contains neither n_- nor $n_\times$, i.e. it is an inner representation of a conic, see (5). Moreover, we have

$$P_i \cdot (Q_O \wedge \bar{n}_- \wedge \bar{n}_\times)^* = (P_i \wedge Q_O \wedge \bar{n}_- \wedge \bar{n}_\times)^* = 0$$

for each $i = 1, \ldots, 5$. In other words, all points P_i lie on Q and thus Q is the unique conic spanned by them. A similar argumentation holds for the axes-aligned conic Q^{al}, for the circle C, and also for the line L. Namely, $(Q_O^{al} \wedge \bar{n}_- \wedge \bar{n}_\times)^*$ is the inner representation of a conic with a zero coefficient at $\bar{n}_\times$. Hence its inner product with an embedded point $C(x, y)$ does not contain the mixed term xy and consequently the conic is aligned to the axes. The dual to C_O is a vector of type (5) which contains neither $\bar{n}_\times$ nor $\bar{n}_-$, therefore it represents a circle. Similarly, L_O is a vector of type (5) which contains none of $\bar{n}_\times$, $\bar{n}_-$, $\bar{n}_+$, therefore it represents a line. Let us remark that the form of outer representations of circles and lines can be also read off Remark 2.3. □

Remark 4.3. The above proposition also explains the choice of null vectors $n_+, n_-, n_\times$ that represent 'infinities'. An axes-aligned conic is a conic containing one infinity, a circle is a conic containing two infinities and a line is a conic containing all three infinities.

Remark 4.4. The dimension zero GAC objects are point quadruplets and they are given by wedge of the corresponding GAC points.

5. Euclidean Transformations and Scaling

The main advantage of GAC compared to models like $\mathbb{G}_6$ or QGA is that it is fully operational in the sense that it allows all Euclidean transformations, i.e. rotations and translations. But not just that, it also allows scaling in the sense of (4). Hence, like in the case of CGA (or $\mathbb{G}_{3,1}$), one obtains all conformal transformations. The following propositions specify the form of GAC versor for rotation (rotor), translation (translator), and scaling (scalor).

Proposition 5.1. *The rotor for a rotation around the origin by the angle* φ *is given by* $R = R_+(R_1 \wedge R_2)$, *where*

$$R_+ = \cos(\tfrac{\varphi}{2}) + \sin(\tfrac{\varphi}{2})e_1 \wedge e_2, \tag{21}$$

$$R_1 = \cos(\varphi) + \sin(\varphi)\bar{n}_\times \wedge n_-, \tag{22}$$

$$R_2 = \cos(\varphi) - \sin(\varphi)\bar{n}_- \wedge n_\times. \tag{23}$$

Proof. It is easy to see that R_+, R_-, $R_\times$ are invertible elements of unite length. Indeed, the reverse is obtained by changing the sign at the bivector part, and such an element is the inverse at the same time. Hence R is also a unit versor. Then observe that R_+ acts non-trivially only on the $\mathbb{G}_{3,1}$ part of the embedding while the action of $R_1 \wedge R_2$ on this part is trivial. We know that R_+ acts correctly since it has the form of the usual $\mathbb{G}_{3,1}$ rotor and hence it is sufficient to show that $R_1 \wedge R_2$ acts on the n_- and $n_\times$ part as a rotation. We compute

$$R_1 \wedge R_2 = \cos^2\theta + \sin\theta\cos\theta(\bar{n}_\times \wedge n_- - \bar{n}_- \wedge n_\times) + \sin^2\theta(\bar{n}_- \wedge \bar{n}_\times \wedge n_\times \wedge n_-)$$

and then

$$(R_1 \wedge R_2)n_-(\tilde{R}_2 \wedge \tilde{R}_1) = (\cos^2\theta - \sin^2\theta)n_- - (2\sin\theta\cos\theta)n_\times,$$

$$(R_1 \wedge R_2)n_\times(\tilde{R}_2 \wedge \tilde{R}_1) = (2\sin\theta\cos\theta)n_- + (\cos^2\theta - \sin^2\theta)n_\times.$$

Hence on the n_- and $n_\times$ part of $C(x, y)$, see (2), we get

$$\begin{aligned}
&(R_1 \wedge R_2)(\tfrac{1}{2}(x^2 - y^2)n_- + xyn_\times)(\tilde{R}_2 \wedge \tilde{R}_1) \\
&\quad = (\tfrac{1}{2}(x^2 - y^2)(\cos^2\theta - \sin^2\theta) + 2xy\sin\theta\cos\theta)n_- \\
&\qquad + (xy(\cos^2\theta - \sin^2\theta) - (x^2 - y^2)\sin\theta\cos\theta)n_\times \\
&\quad = \tfrac{1}{2}(\bar{x}^2 - \bar{y}^2)n_- + \bar{x}\bar{y}n_\times,
\end{aligned}$$

where $\bar{x} = x\cos\theta + y\sin\theta$, $\bar{y} = -x\sin\theta + y\cos\theta$ are the rotated coordinates. Putting all together we have

$$R_+(R_1 \wedge R_2)C(x,y)(\tilde{R}_2 \wedge \tilde{R}_1)\tilde{R}_+ = C(\bar{x},\bar{y}).$$

□

Proposition 5.2. *The translator is given by $T = T_+T_-T_\times$, where*

$$T_+ = 1 - \tfrac{1}{2}ue_1 \wedge n_+ \tag{24}$$
$$T_- = 1 - \tfrac{1}{2}ue_1 \wedge n_- + \tfrac{1}{4}u^2 n_+ \wedge n_- \tag{25}$$
$$T_\times = 1 - \tfrac{1}{2}ue_2 \wedge n_\times \tag{26}$$

for a translation in the direction e_1 around u. Similarly, for a translation in the direction e_2 around v one has

$$T_+ = 1 - \tfrac{1}{2}ve_2 \wedge n_+ \tag{27}$$
$$T_- = 1 + \tfrac{1}{2}ve_2 \wedge n_- - \tfrac{1}{4}v^2 n_+ \wedge n_- \tag{28}$$
$$T_\times = 1 - \tfrac{1}{2}ve_1 \wedge n_\times \tag{29}$$

Proof. We prove only the first part of the proposition since the proof of the second part is almost the same. First observe that T is a unit versor and that T_+ acts non-trivially only on the $\mathbb{G}_{3,1}$ part of embedded point, see (2), and since it has the same form as the $\mathbb{G}_{3,1}$ translator we know that it acts correctly on this part. For the next part of the translator we compute

$$T_- e_1 \tilde{T}_- = e_1 + un_-$$
$$T_- \bar{n}_+ \tilde{T}_- = \bar{n}_+ + \tfrac{1}{2}u^2 n_-$$

and the action on all other basis elements in $C(x,y)$ is trivial. Hence

$$T_- C(x,y)\tilde{T}_- = C(x,y) + \tfrac{1}{2}(u^2 + 2xu)n_-.$$

Similarly, we compute that $T_\times$ acts trivially on $C(x,y)$ up to $T_\times e_2 \tilde{T}_\times = e_2 + un_\times$. Therefore we have

$$T_- T_\times C(x,y)\tilde{T}_\times \tilde{T}_- = C(x,y) + \tfrac{1}{2}(u^2 + 2xu)n_- + uyn_\times.$$

Now it is easy to see that altogether we have

$$T_+T_-T_\times C(x,y)\tilde{T}_\times\tilde{T}_-\tilde{T}_+ = C(x+u,y).$$

□

Remark 5.3. The conic's inner representation form may be derived in a way different from Sect. 3. Indeed, the model covariancy may be used and vectors representing "zero position" conics in Example 3.1 may be sandwich-multiplied by versors for translation and rotation.

Proposition 5.4. *The scalor for a scaling by α is given by $S = S_+S_-S_\times$, where*

$$S_+ = \tfrac{\alpha+1}{2\sqrt{\alpha}} + \tfrac{\alpha-1}{2\sqrt{\alpha}}\bar{n}_+ \wedge n_+ \tag{30}$$
$$S_- = \tfrac{\alpha+1}{2\sqrt{\alpha}} + \tfrac{\alpha-1}{2\sqrt{\alpha}}\bar{n}_- \wedge n_- \tag{31}$$
$$S_\times = \tfrac{\alpha+1}{2\sqrt{\alpha}} + \tfrac{\alpha-1}{2\sqrt{\alpha}}\bar{n}_\times \wedge n_\times. \tag{32}$$

Proof. The part S_+ is the $\mathbb{G}_{3,1}$ versor for a scaling. Indeed, $S_+\tilde{S}_+ = 1$, and its action on $\mathcal{C}$ yields

$$\begin{aligned} S_+C(x,y)\tilde{S}_+ =& \tfrac{1}{\alpha}\bar{n}_+ + xe_1 + ye_2 + \tfrac{1}{2}\alpha(x^2+y^2) \\ &+ \tfrac{1}{2}(x^2-y^2) + xy). \end{aligned}$$

The parts S_- and $S_\times$ are also unit versors and they act by the multiplication by α on the n_- basis element and $n_\times$ basis element, respectively. Together we get

$$S_+S_-S_\times C(x,y)\tilde{S}_\times\tilde{S}_-\tilde{S}_+ = \tfrac{1}{\alpha}C(\alpha x, \alpha y).$$

□

Remark 5.5. Note that R_+ commutes with R_1 and R_2 but R_1 and R_2 do not commute. $T_\times$ commutes with T_+ and T_- but T_+ and T_- do not commute. Moreover, the translators in x and y direction do not commute in general but they, of course, commute on $\mathcal{C}$. All this is best checked on the Lie algebra level. The generators of the above operators are given as follows:

$$\begin{aligned} \text{rotation: } r &= \tfrac{1}{2}e_1 \wedge e_2 + \bar{n}_\times \wedge n_- + n_\times \wedge \bar{n}_-, \\ x-\text{translation: } t_1 &= -\tfrac{1}{2}e_1 \wedge n_+ - \tfrac{1}{2}e_1 \wedge n_- - \tfrac{1}{2}e_2 \wedge n_\times, \\ y-\text{translation: } t_2 &= -\tfrac{1}{2}e_2 \wedge n_+ + \tfrac{1}{2}e_2 \wedge n_- - \tfrac{1}{2}e_1 \wedge n_\times. \\ \text{scaling: } s &= \bar{n}_+ \wedge n_+ + \bar{n}_- \wedge n_- + \bar{n}_\times \wedge n_\times \end{aligned}$$

This is the full list of conformal transformations generators in GAC. However, the conic sections transform naturally under projective transformations. The group of projective transformations is of dimension eight. The remaining transformations are non-isotropic scalings and shears. The discussion of these transformations lies beyond the scope of this paper and will be a subject to further research.

Note that the versors from the above propositions are spinors and thus conformal transformations are represented by orthogonal transformations on $\mathbb{R}^{5,3}$ with respect to metric (1). And since conics and their intersections are represented by null spaces of inner products with points, the versors transform correctly not just points but also conics and their intersections, i.e. all GAC entities.

Remark 5.6. If we replace a spinor by an invertible element of grade one, we get a non-Euclidean transformation which we call a general reflection (or inversion) in GAC. We just mention that in the case that this element of grade one is the inner representation of an ellipse (or another GAC entity) we get an inversion in the ellipse (or in the given entity). We provide some illustrative examples at the end of the next section.

6. An Implementation in Maple

We use a Maple package CLIFFORD for computations in Clifford and Grassmann algebras, [1].

6.1. The Initialization of GAC

The initialization of the Clifford algebra $\mathbb{G}_{5,3}$ in CLIFFORD is given by prescribing the matrix of bilinear form B according to (1).

```
with(Clifford);
B := blockmatrix(3,3 , [ Matrix(3,3),Matrix(3,2),
-IdentityMatrix(3,3),Matrix(2,3),IdentityMatrix(2,2),
Matrix(2,3),-IdentityMatrix(3,3),Matrix(3,2),Matrix(3,3)]);
```

The package uses as a default the standard Grassmann basis in $\Lambda\mathbb{R}^8$, where $\mathbb{R}^8$ is spanned by the vectors called e_n for $1 \le n \le 8$. We make aliases in order to emphasize which basis elements play the role of origins, the role of Euclidean basis, and the role of infinities. Note that the notation differs from the former one, e_1 is called ex, $\bar{n}_+$ is called $e0p$, n_+ is called $einfp$ etc.

```
alias(e0p=e1,e0m=e2,e0k=e3,ex=e4,ey=e5,einfp=e6,
einfm=e7,einfk=e8);
```

Now the definition of GAC is finished by defining the embedding of the plane according to (2)

```
C:=proc (x,y)
e0p+x*ex+y*ey+1/2*(x^2+y^2)*einfp
+1/2*(x^2-y^2)*einfm+x*y*einfk;
end proc:
```

The duality between representations is implemented according to Remark 4.1 as follows

```
OtoI:= proc (A)
LC(A,e1 &w e2 &w e3 &w e4 &w e5 &w e6,B);
end proc:
ItoO:= proc (A)
LC(A,e1 &w e4 &w e5 &w e6 &w e7 &w e8,B);
end proc:
```

6.2. The Inner Product Representation of Conics

The inner product representation of conics in terms of their internal parameters and position and orientation in the plane is given exactly according to propositions in Sect. 3. For example, the procedure giving the GAC inner product representation of an ellipse reads

```
Ell:= proc(r,s,u,v,theta)
local be,al,th;
th:=2*theta;
al:=(r^2-s^2)/(r^2+s^2);
be:=2*r^2*s^2/(r^2+s^2);
e0p-al*cos(th)*e0m-al*sin(th)*e0k+(u-u*al*cos(th)
-v*al*sin(th))*ex+(v+v*al*cos(th)-u*al*sin(th))*ey
+1/2*(u^2+v^2-be-(u^2-v^2)*al*cos(th)-2*u*v*al*sin(th))*einfp;
end proc:
```

Example 6.1. Let us compute the inner product representation of an ellipse $E1$ with semi-axes of length 2 and 1, with the centre point $(2, 1)$, and an ellipse $E2$ with the same parameters but rotated by $\pi/6$ using the above procedure. The code

```
E1:=Ell(2,1,2,1,0):
E2:=Ell(2,1,2,1,Pi/6):
```

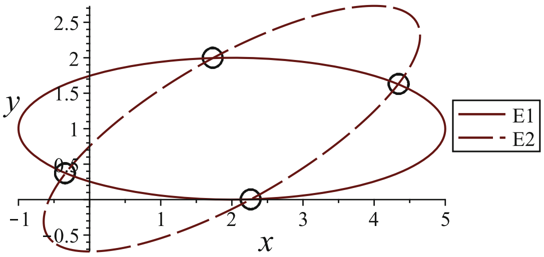


FIGURE 1. Two ellipses and their intersections

gives

$$E1 := e0p - (3/5) * e0m + (4/5) * ex + (8/5) * ey + (4/5) * einfp$$

$$E2 := e0p - (3/10) * e0m - (3/10) * sqrt(3) * e0k + (7/5 - (3/10) * sqrt(3)) * ex$$
$$+\ (13/10 - (3/5) * sqrt(3)) * ey + (1/2 * (5/2 - (6/5) * sqrt(3))) * einfp$$

To visualize the result we compute the inner product of these vectors and we plot the null points of the resulting expression. The inner product is computed by procedure LC from the CLIFFORD package.

```
eq1:=subs({Id=1},LC(C(x,y),E1,B))=0:
eq2:=subs({Id=1},LC(C(x,y),E2,B))=0:
Graph:=implicitplot([eq1,eq2],x=-3..5,y=-3..3,
scaling=constrained,gridrefine=3,legend=["E1","E2"],
linestyle=[solid,longdash]);
```

We also compute the intersections of these two ellipses according to (12) and we display the result in one graph, see Fig. 1.

```
LC(C(x,y),E1 &w E2,B):
sol:=map(allvalues,clisolve(
points:=[]:
for i from 1 to nops(sol1) do points:=[op(points),
[rhs(sol1[i][1]),rhs(sol1[i][2])]] end do:
GraphPoints:=pointplot(points,symbolsize=30,symbol=circle):
plots[display]({Graph,GraphPoints});
```

Example 6.2. Let us consider an example of two hyperbolas and two parabolas. Let H1 be a hyperbola with semi-axes 2 and 1 placed in the origin and rotated by $\pi/6$, and let H2 be the same hyperbola but translated to $(1, 2)$. Let P1 be a parabola with a semilatus rectum 1 shifted to $(3, 1)$ and rotated by $\pi/2$, and let P2 be a parabola in the same position but with the semilatus rectum equal to 2. Maple procedures for GAC inner product representations defined according to (9) and (10) then give

$$H1 := e0p - (5/6) * e0m - (5/6) * sqrt(3) * e0k + (4/3) * einfp$$

$$H2 := e0p - (5/6) * e0m - (5/6) * sqrt(3) * e0k + (1/6 - (5/3) * sqrt(3)) * ex$$
$$+ (11/3 - (5/6) * sqrt(3)) * ey + (1/2 * (61/6 - (10/3) * sqrt(3))) * einfp$$

$$P1 := e0p - e0m - 2 * ex + 2 * ey - 5 * einfp$$

$$P2 := e0p - e0m - 4 * ex + 2 * ey - 11 * einfp$$

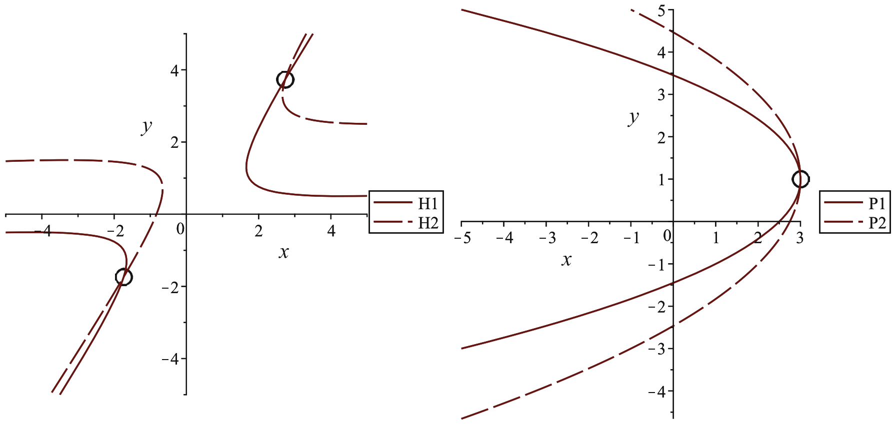


FIGURE 2. Two hyperbolas and parabolas and their intersections

Replacing E1, E2 in the above code by H1, H2 and P1, P2, respectively, we obtain graphs displayed in Figure 2.

6.3. The outer product representation of conics

The conics can be easily constructed from points lying on them by taking their wedge product according to Proposition 4.2.

Example 6.3. Let us compute a conic spanned by points [3,4], [−2,−1], [1,−2], [−4,2] and [−3,4]. Note that we move from the outer product representation to the inner product representation using the procedure OtoI defined above.

```
Conic:=C(3,4) &w C(-2,-1) &w C(1,-2) &w C(-4,2) &w C(-3,4);
ConicI:=OtoI(Conic);
```

This code gives

$$\begin{aligned} ConicI := & -630 * e0k - 3570 * e0p + 1050 * e0m - 6930 * ey \\ & - 2520 * ex + 20580 * einfp \end{aligned}$$

We compute the inner product of this vector with an embedded point, and we prove that we have got the right conic by plotting the result. Indeed, the following code provides the graph displayed in Fig. 3.

```
eq:=subs({Id=1},LC(C(x,y),CI,B))=0:
Gr1:=implicitplot(eq,x=-5..5,y=-5..5,scaling=constrained,
gridrefine=3):
Gr2:=pointplot([[3,4], [-2,-1], [1,-2], [-4,2], [-3,4]],
symbolsize=30,symbol=circle):
plots[display]({Gr1,Gr2});
```

Example 6.4. Let us take the same list of points as in the previous example and let us compute and plot the axes-aligned conic AI spanned by the first four points from the list, the circle CI spanned by the first three points and the line LI spanned by the first two points. According to Proposition 4.2, we have

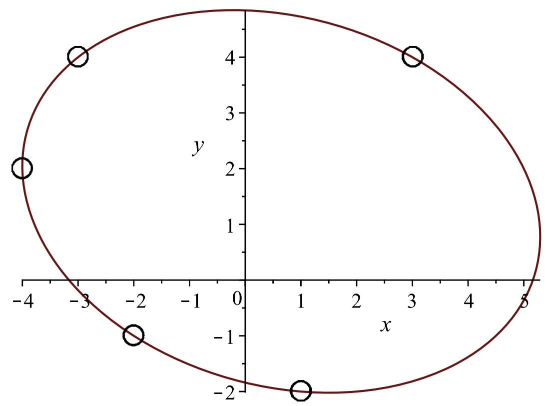


FIGURE 3. An ellipse spanned by five points

```
AI:=OtoI(C(3,4) &w C(-2,-1) &w C(1,-2) &w C(-4,2) &w einfk):
CI:=OtoI(C(3,4) &w C(-2,-1) &w C(1,-2) &w einfm &w einfk):
LI:=OtoI(C(3,4) &w C(-2,-1) &w einfp &w einfm &w einfk):
```

and this gives

$$AI := -80 * e0m + 50 * e0p + 75 * ey + 105 * ex - 290 * einfp$$
$$CI := 20 * e0p + 30 * ey + 10 * ex - 100 * einfp$$
$$LI := -5 * ey + 5 * ex - 5 * einfp$$

Computing inner products and plotting results, we obtain a graphical verification of correctness, see Fig. 4.

6.4. The transformations in GAC

The transformations are implemented as procedures according to propositions in Sect. 5. For example, the rotation is given by Proposition 5.1 as follows

```
Rotor:=proc (t)
local Rot1,Rot2,Rot3;
Rot1:=cos(t/2)+sin(t/2)*(ex &w ey);
Rot2:=cos(t)+sin(t)*(e0k &w einfm);
Rot3:=cos(t)-sin(t)*(e0m &w einfk);
Rot1 &c (Rot2 &w Rot3);
end proc:
rot:= proc (A,t)
Rotor(t) &c A &c Rotor(-t);
end proc:
```

The procedures defining versors for translations in the x and y direction, given by Proposition 5.2, are called T1 and T2, respectively, and the procedure defining the scalor given by Proposition 5.4 is called S. Then the procedures for translation and scaling read

```
trans:= proc(A,x,y)
T1(x) &c T2(y) &c A &c reversion(T2(y),B) &c reversion(T1(x),B);
end proc:
scale:= proc(A,t)
S(t) &c A &c reversion(S(t),B);
end proc:
```

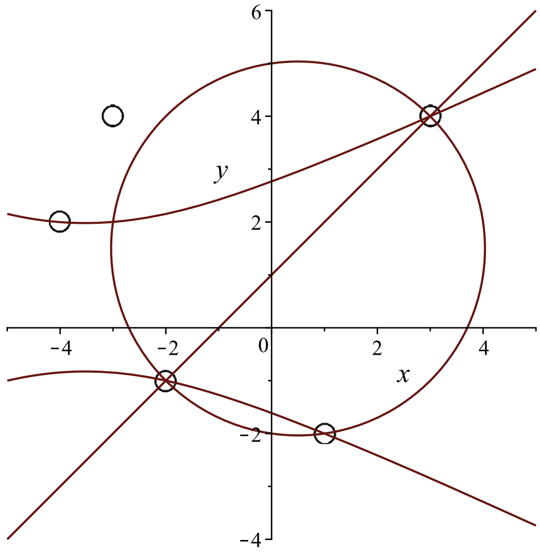


FIGURE 4. An axes-aligned conic, a circle, and a line spanned by given points

Example 6.5. Let us compute Euclidean transformations of an ellipse E with semi-axes 3 and 2 aligned to the axes with its center in the origin. First, we compute translation by $(1, 2)$, then rotation by $\pi/6$, and finally rotation by $\pi/6$ of the translated ellipse.

```
E:=Ell(3,2,0,0,0): Et:=trans(E,1,2):
Er:=rot(E,Pi/6):   Ert:=rot(Et,Pi/6):
```

We get

$$
\begin{aligned}
E &:= e0p - (5/13) * e0m - (36/13) * einfp \\
Et &:= e0p - (5/13) * e0m + (8/13) * ex + (36/13) * ey + (4/13) * einfp \\
&\quad - (32/13) * einfm + (36/13) * einfk \\
Er &:= e0p - (5/26) * e0m + (5/26) * sqrt(3) * e0k - (36/13) * einfp \\
Ert &:= e0p - (5/26) * e0m + (5/26) * sqrt(3) * e0k \\
&\quad + (2/13 * (9 + 2 * sqrt(3))) * ex \\
&\quad + (2/13 * (-2 + 9 * sqrt(3))) * ey + (4/13) * einfp \\
&\quad + (2/13 * (9 * sqrt(3) - 8)) * einfm + (2/13 * (9 + 8 * sqrt(3))) * einfk
\end{aligned}
$$

and the corresponding ellipses are displayed in Fig. 5.

Example 6.6. We also compute a rescaling of the rotated ellipse Er. Note that the scaling is given by conjugation with a versor given by Proposition 5.4. The code

```
Ers:=scale(Er,1.5):
```

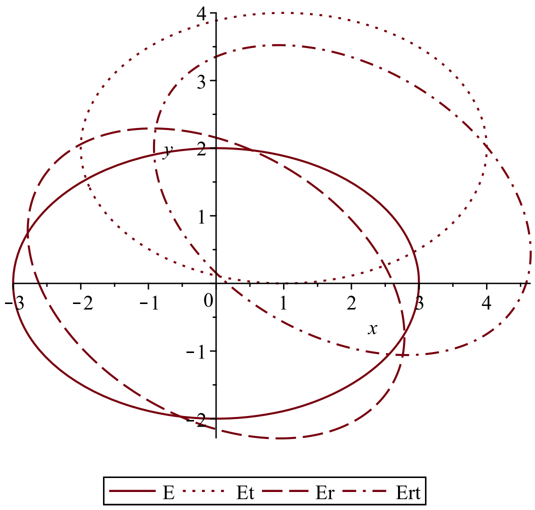


FIGURE 5. Euclidean transformations of an ellipse

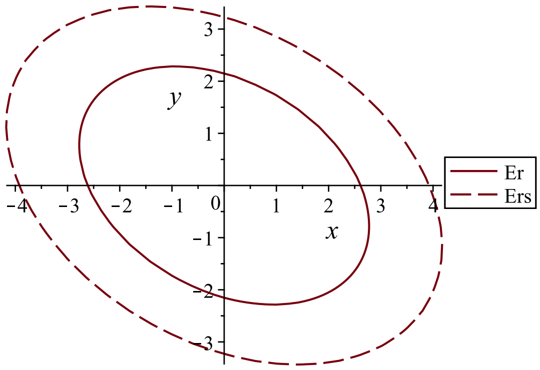


FIGURE 6. A scaling of an ellipse

gives

$$\begin{aligned}Ers :=& .589824 * e0p - .113427692 * e0m + .196462526 * e0k \\ & - 3.675057232 * einfp\end{aligned}$$

and the resulting ellipse is displayed in Fig. 6.

Remark 6.7. If we replace the above versors by an invertible vector, we get a transformation which is neither Euclidean nor conformal. Figure 7 shows what happens when the vector is a line and we act on an ellipse and when the vector is an ellipse and we act on a line, respectively.

Figure 8 shows what happens when the vector is a an ellipse and we act on a point. The inversion of an inner point with respect to the ellipse gives the outer ellipse and the inversion of an outer point gives the inner ellipse.

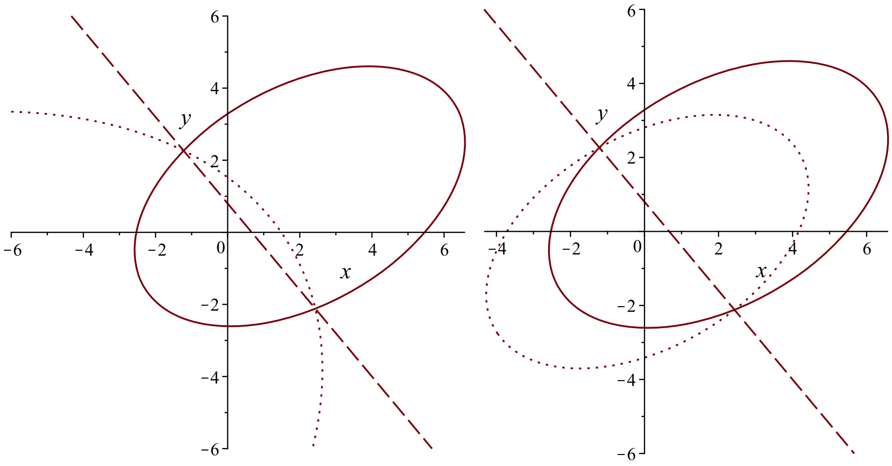


FIGURE 7. The inversion of a line in an ellipse and the ellipse in a line

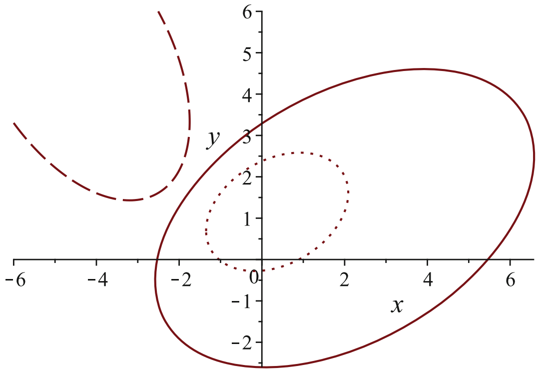


FIGURE 8. The inversion of inner and outer point in an ellipse

7. Comparison to Known Concepts

Nowadays, the corresponding theory is partially elaborated for $\mathbb{G}_{5,3}$ and $\mathbb{G}_6$ by Perwass in [9], for (two-dimensional version of) double conformal geometric algebra (DCGA) in [2,3] by Easter and Hitzer, and for (two-dimensional version of) quadric geometric algebra (QGA) in [10] by Zamora-Esquivel. While the concept of $\mathbb{G}_6 = \mathbb{G}_{6,0}$, [9], seems to be geometrically correct for conics, its main disadvantage is that it does not contain translations. Moreover, even rotations are obtained as algebraic operator in [9] with no clear natural explanation. The concepts of DCGA and QGA are two different rather algebraic extensions of CGA.

More precisely, QGA, [10], is a geometric algebra $\mathbb{G}_{6,3}$ together with an embedding of a Euclidean space $\mathbb{R}^3$ constructed as three independent stereographic projections for each axis separately. In the planar case, the algebra is $\mathbb{G}_{4,2}$ and the embedding of the plane is given by

$$c_Q(x,y) = e_{01} + e_{02} + xe_1 + ye_2 + \frac{1}{2}(x^2 e_{\infty 1} + y^2 e_{\infty 2}), \tag{33}$$

where the basis $e_{01}, e_{02}, e_1, e_2, e_{\infty 1}, e_{\infty 2}$ has the same meaning as in CGA but is (orthogonally) extended by one Witt pair of null vectors $e_{02}, e_{\infty 2}$. Following the considerations in CGA, the euclidean entities described in the inner product representation are exactly the algebraic varieties generated by the polynomials in $1, x, y, x^2, y^2$ and therefore just axes-aligned conics are well defined. Moreover, the operations that work correctly in QGA include translation, transposition, dilation, and intersection only. In general, rotation does not work correctly neither for conics nor points. Particularly, rotation of points must be performed in CGA after projection. Moreover, a conic rotated by an arbitrary angle cannot be represented by any known QGA entity.

Next, the planar concept of DCGA, [2,3], corresponds to the geometric algebra $\mathbb{G}_{6,2}$. It contains two subalgebras isomorphic to the conformal geometric algebra $\mathbb{G}_{3,1}$ which we denote by plus and minus sign superscript. Following CGA, let us choose a basis $e_0^{\pm}, e_1^{\pm}, e_2^{\pm}, e_\infty^{\pm}$ of the generating quadratic vector space $\mathbb{R}^{6,2} = \mathbb{R}^{3,1} \oplus \mathbb{R}^{3,1}$ in which the quadratic form has a block-diagonal form with blocks given as in CGA. Then a point (x, y) is mapped to a DCGA (double) point $P = P^+ \wedge P^-$ by

$$P^{\pm} = xe_1^{\pm} + ye_2^{\pm} + \frac{1}{2}(x^2 + y^2)e_\infty^{\pm} + e_0^{\pm}. \tag{34}$$

The key fact is the following. If $A \in \mathbb{G}_{3,1}^+$ and $B \in \mathbb{G}_{3,1}^-$ and $P = P^+ \wedge P^-$ is a (double) point then the properties of the inner product imply

$$(P^+ \wedge P^-) \cdot (A \wedge B) = -(P^+ \cdot A)(P^- \cdot B).$$

Hence the DCGA inner product representations are obtained as products of CGA representations and DCGA objects are algebraic varieties generated by polynomials which are linear combinations of 1, x, y, $x^2 + y^2$, xy, $x(x^2 + y^2)$, $y(x^2 + y^2)$, and $(x^2 + y^2)^2$. Since the map from the Euclidean space to DCGA is a product of two inverses to stereographic projections, and thus a conformal map, all conformal transformations work and they are represented by the same algebra elements as in CGA. Unfortunately DCGA does not satisfy a basic property, namely the uniquess of the representation. The correspondence between geometric objects and algebraic entities is ambiguous. Consequently, it lacks the duality between the inner and outer representation.

On the other hand, we continue in another C. Perwass's concept, [9], precisely that of $\mathbb{G}_{5,3}$ and finalize its description by the complete classification of the elements corresponding to the appropriate conic sections with full advantages of geometric algebra approach, e.g. presence of the duality and presence of Euclidean transformations as algebra elements with natural and well defined action (Table 1).

Table 1. The symbol $\odot$ stands for capable, $\times$ for incapable

	DCGA	QGA	GAC
dimension (for the plane)	8	6	8
representation of conics	$\odot$	$\odot$	$\odot$
IPNS - OPNS duality	$\times$	$\odot$	$\odot$
Euclidean transformations	$\odot$	$\times$	$\odot$

8. Conclusion

We described representation of the Euclidean plane that provides the same efficiency for calculations with arbitrary conic sections as is performed by CGA for spheres. More precisely, we defined the Geometric Algebra for Conics (GAC) as Clifford algebra $\mathbb{G}_{5,3}$ together with the embedding (2) of the plane. Consequently, we finalized the concept of C. Perwass by the complete description of the elements corresponding to the appropriate conic sections as well as to particular transformations. Particularly, we provided the equation of ellipse and hyperbola, (9), and parabola, (10), in the GAC inner product representation and a description of a general conic in the outer product representation, (4.2). Furthermore, we have found the exact GAC elements that represent the transformations, such as rotation, (5.1), translation, (5.2), and scaling, (5.4), which proves the correctness of the GAC concept. We conclude by the list of examples in Maple displayed together with the source code.

References

[1] Ablamowicz, R., Fauser, B.: CLIFFORD/Bigebra, a Maple package for Clifford (Co)algebra computations. http://www.math.tntech.edu/rafal/ (2015)

[2] Easter, R.B., Hitzer, E.: Double conformal geometric algebra. Adv. Appl. Clifford Algebras **27**, 2175 (2017). https://doi.org/10.1007/s00006-017-0784-0

[3] Easter, R.B., Hitzer, E.: Double conformal geometric algebra. Adv. Appl. Clifford Algebras **27**(3), 2175–2199 (2017)

[4] Gonzalez-Jimenez, L., Carbajal-Espinosa, O., Loukianov, A., Bayro-Corrochano, E.: Robust pose control of robot manipulators using conformal geometric algebra. Adv. Appl. Clifford Algebras **24**(2), 533–552 (2014)

[5] Hildenbrand, D.: Foundations of geometric algebra computing. Springer, New York (2013)

[6] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: Geometric algebras for uniform colour spaces. Math. Meth. Appl. Sci. (2017). https://doi.org/10.1002/mma.4489

[7] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: CGA-based robotic snake control. Adv. Appl. Clifford Algebr. **27**(1), 621–632 (2017)

[8] Lounesto, P.: Clifford algebra and spinors, 2nd edn. CUP, Cambridge (2006)

[9] Perwass, Ch.: Geometric algebra with applications in engineering. Springer, New York (2009)

[10] Zamora-Esquivel, J.: $\mathbb{G}_{6,3}$ geometric algebra; description and implementation. Adv. Appl. Clifford Algebras **24**(2), 493–514 (2014)

Jaroslav Hrdina, Aleš Návrat and Petr Vašík
Institute of Mathematics, Faculty of Mechanical Engineering
Brno University of Technology
Brno
Czech Republic
e-mail: `hrdina@fme.vutbr.cz`

Aleš Návrat
e-mail: `navrat.a@fme.vutbr.cz`

Petr Vašík
e-mail: `vasik@fme.vutbr.cz`

Received: January 31, 2018.
Accepted: June 12, 2018.

Appendix 5

[32] Hrdina, J., Návrat, A., Vašík, P.: *Quantum Computing Based on Complex Clifford Algebras*, Quantum Information Processing (2022) 21:310, 21:310 doi:10.1007/s11128-022-03648-w

Quantum Information Processing (2022) 21:310
https://doi.org/10.1007/s11128-022-03648-w

Quantum computing based on complex Clifford algebras

Jaroslav Hrdina[1] · Aleš Návrat[1] · Petr Vašík[1]

Received: 4 March 2022 / Accepted: 3 August 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

We propose to represent both n-qubits and quantum gates acting on them as elements in the complex Clifford algebra defined on a complex vector space of dimension $2n$. In this framework, the Dirac formalism can be realized in straightforward way. We demonstrate its functionality by performing quantum computations with several well known examples of quantum gates. We also compare our approach with representations that use real geometric algebras.

Keywords Quantum computing · Clifford algebras

Mathematics Subject Classification 68Q12 · 15A66

1 Introduction

Real geometric (Clifford) algebras (GA) may be understood as a generalisation of well known quaternions which is an alternative for matrix description of orthogonal transformations. Real geometric algebras have a wide range of applications in robotics [14, 17], image processing [7], numerical methods [18], etc. Among the main advantage of this approach we count the calculation speed, straightforward and geometrically oriented implementation and effective parallelisation [13, 16]. We stress that all these implementations are using Clifford's geometric algebra to represent specific orthogonal transformations.

The authors was supported by the Grant No. FSI-S-20-6187.

✉ Aleš Návrat
navrat.a@fme.vutbr.cz

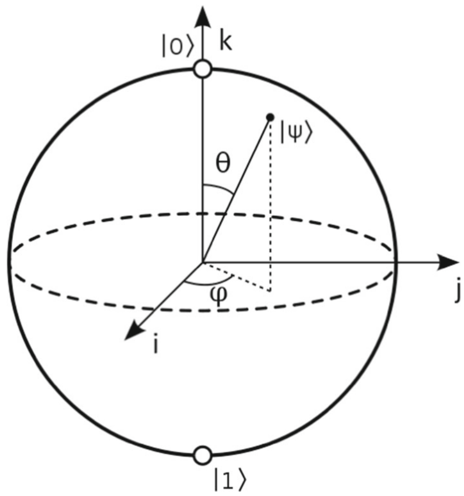
Jaroslav Hrdina
hrdina@fme.vutbr.cz

Petr Vašík
vasik@fme.vutbr.cz

1 Institute of Mathematics, Faculty of Mechanical Engineering, Brno University of Technology, Brno, Czech Republic

Published online: 17 September 2022 Springer

Fig. 1 Bloch sphere for a qubit represented in quaternions



Recently, an increasing number of papers adopt the apparat of real GA in the description, elaboration and analysis of Quantum Computing (QC) algorithms [1, 6, 8]. The basic idea for this lies in identification of a state qubit with a Bloch sphere together with the identification of qubit gates with the rotations of the sphere. Namely, it is well known that a normalized qubit can be written in terms of basis vectors $|0\rangle, |1\rangle$ as

$$|\psi\rangle = \cos(\theta/2)|0\rangle + (\cos\varphi + i\sin\varphi)\sin(\theta/2)|1\rangle, \tag{1.1}$$

where $0 \leq \theta \leq \pi$ and $0 \leq \varphi < 2\pi$ and that parameters θ, φ can be interpreted as spherical coordinates of a point on the unit sphere in $\mathbb{R}^3$, see Fig. 1.

The spirit of GA description is to represent a point on the sphere by the rotation that takes a fixed initial point to that point. Under the usual choices, see Fig. 1, the initial point is the north pole $(0, 0, 1)$ and the qubit (1.1) corresponds to the counterclockwise rotation by θ with respect to y axis composed with the clockwise rotation by φ with respect to z axis. In particular, the basis states $|0\rangle, |1\rangle$ are represented by the identity and rotation $\theta = \pi$ respectively. In the quaternionic description we have $(0, 0, 1) = \mathbf{k}$ and the rotations are represented by elements $\exp(-1/2\theta\mathbf{j}) = \cos(\theta/2) - \mathbf{j}\sin(\theta/2)$ and $\exp(1/2\varphi\mathbf{k}) = \cos(\varphi/2) + \mathbf{k}\sin(\varphi/2)$ respectively. Indeed, the Bloch sphere representation of qubit (1.1) is given by

$$e^{\frac{1}{2}\varphi\mathbf{k}}e^{-\frac{1}{2}\theta\mathbf{j}}\mathbf{k}e^{\frac{1}{2}\theta\mathbf{j}}e^{-\frac{1}{2}\varphi\mathbf{k}} = \cos\varphi\sin\theta\mathbf{i} + \sin\varphi\sin\theta\mathbf{j} + \cos\theta\mathbf{k}$$

In this sense the qubit state (1.1) is represented by quaternion $\exp(1/2\varphi\mathbf{k})\exp(-1/2\theta\mathbf{j})$. In particular, the basis states $|0\rangle, |1\rangle$ are represented by quaternions $|0\rangle = 1$ and $|1\rangle = \exp(-1/2\pi\mathbf{j}) = -\mathbf{j}$ respectively which in GA language is Eq. (4.3), see Sect. 4.1 for more details. In principle, this representation of qubits is based on exceptional isomorphisms of low-dimensional Lie groups $SU(2)$, $Spin(3)$ and $Sp(1)$. For higher dimensions, no similar natural identification of unitary and spin groups exists. Therefore, to realise the states of multi-qubit, it is necessary to use so-called correlators which increases algebraic abstraction and lacks the original elegance. Indeed, such approach is no competition to elegant Dirac formalism.

In our paper, we present an alternative to the qubit representation in the form of complex Clifford algebras which are indeed substantially more appropriate for QC representation by means of GA because they respect the complex nature of quantum theory. In our approach, Dirac formalism may be translated to complex GA of an appropriate dimension in rather straightforward way and, in spite of abstract symbolic Dirac formalism, all expressions are represented in a particular algebra and thus may be manipulated and implemented as algebra elements directly, without any need for matrix representation. In the sequel, we briefly recall the definition of real GA and provide a more detailed introduction to complex GA. In Sect. 3, we show the representation of qubits and multi-qubits in complex GA, their transformations (gates). We provide an explicit forms of elementary 1-gates and 2-gates. We also discuss the case of multi-gates, ie. gates obtained by a tensor product. In Sect. 4, we describe a qubit by means of real geometric algebra. More precisely, we compare a description known from literature, ie. the one based on the isomorphism of unitary group $SU(2)$ and spin group $Spin(3)$, with the real description following from our complex GA approach and an isomorphism $\mathbb{C}_2 \to \mathbb{G}_3$ of complex and real algebra.

2 Complex Clifford algebras

A Clifford algebra is a normed associative algebra that generalizes the complex numbers and the quaternions. Its elements may be split into Grassmann blades and the ones with grade one can be identified with the usual vectors. The geometric product of two vectors is a combination of the commutative inner product and the anti-commutative outer product. The scalars may be real or complex. In Sect. 3 we show that the complex Clifford algebra constructed over a quadratic space of even dimension can be efficiently used to represent quantum computing but we start with the real case.

2.1 Real Clifford algebras

The construction of the universal real Clifford algebra is well-known, for details see e.g. [9, 20]. We give only a brief description here. Let the real vector space $\mathbb{R}^m$ be endowed with a non-degenerate symmetric bilinear form B of signature (p, q), and let $(e_1, \ldots, e_m)$ be an associated orthonormal basis, i.e.

$$B(e_i, e_j) = \begin{cases} 1 & \text{if } i = j = 1, \ldots, p \\ -1 & \text{if } i = j = p+1, \ldots, m \\ 0 & \text{if } i \neq j \end{cases} \quad \text{where } 1 \leq i, j \leq m = p + q.$$

Let us recall that the Grassmann algebra is an associative algebra with the anti-symmetric outer product defined by the rule

$$e_i \wedge e_j + e_j \wedge e_i = 0 \ \text{ for } 1 \leq i, j \leq m.$$

The Grassmann blade of grade r is $e_A = e_{i_1} \wedge \cdots \wedge e_{i_r}$, where the multi-index A is a set of indices ordered in the natural way $1 \leq i_1 \leq \cdots \leq i_r \leq m$, and we put $e_\emptyset = 1$. Blades of orders $0 \leq r \leq m$ form the basis of the graded Grassmann algebra $\Lambda(\mathbb{R}^m)$. Next, we introduce the inner product

$$e_i \cdot e_j = B(e_i, e_j), \quad 1 \leq i, j \leq m,$$

leading to the so-called *geometric product* in the Clifford algebra

$$e_i e_j = e_i \cdot e_j + e_i \wedge e_j, \quad 1 \leq i, j \leq m,$$

The respective definitions of the inner, the outer and the geometric product are then extended to blades of the grade r as follows. For the inner product we put

$$e_j \cdot e_A = e_j \cdot (e_{i_1} \wedge \cdots \wedge e_{i_r}) = \sum_{k=1}^{r} (-1)^k B(j, i_k) e_{A \setminus \{i_k\}},$$

where $e_{A \setminus \{i_k\}}$ is the blade of grade $r - 1$ created by deleting e_{i_k} from e_A. This product is also called the left contraction in literature. For the outer product we have

$$e_j \wedge e_A = \begin{cases} e_j \wedge e_{i_1} \wedge \cdots \wedge e_{i_r} & \text{if } j \notin A \\ 0 & \text{if } j \in A \end{cases}$$

and for the geometric product we define

$$e_j e_A = e_j \cdot e_A + e_j \wedge e_A.$$

Finally, these definitions are linearly extended to the whole of the vector space $\Lambda(\mathbb{R}^m)$. Thus we get an associative algebra over this vector space, the so-called real Clifford algebra, denoted by $\mathbb{G}_{p,q} = \mathrm{Cl}(p, q, \mathbb{R})$. Note that this algebra is naturally graded; the grade zero and grade one elements are identified with $\mathbb{R}$ and $\mathbb{R}^m$ respectively. The projection operator $\mathbb{G}_{p,q} \to \Lambda^r(\mathbb{R}^m)$ will be denoted by $[\ ]_r$.

This grading define a $\mathbb{Z}_2$-grading of the Clifford algebra according to the parity of grades. Namely, the linear map $v \to -v$ on $\mathbb{R}^m$ extends to an automorphism α called the grade involution and decomposes $\mathbb{G}_{p,q}$ into positive and negative eigenspaces. The former is called the even subalgebra $\mathbb{G}^0_{p,q}$ and the latter is called the odd part $\mathbb{G}^1_{p,q}$. In addition to α, there are two important antiautomorphisms of real Clifford algebras. The first one is $\tilde{x}$ called the reverse or transpose operation and it is defined by extension of identity on $\mathbb{R}^m$ and by the antiautomorphism property $\widetilde{xy} = \tilde{y}\tilde{x}$. The second antiautomorphism is called the Clifford conjugation $\bar{x}$ and the operation is defined by composing α and the reverse

$$\bar{x} = \alpha(\tilde{x}) = \widetilde{\alpha(x)}.$$

2.2 The complexification

When allowing for complex coefficients, the same generators e_A produce by the same formulas the complex Clifford algebra which we denote by $\mathbb{C}_m = \mathrm{Cl}(m, \mathbb{C})$. Clearly, in the complex case no signature is involved, since each basis vector e_j may be multiplied by the imaginary unit i to change the sign of its square. Hence we may assume we start with the real Clifford algebra $\mathbb{G}_m$ with the inner product $e_i e_j = \delta_{ij}$, $1 \le i, j \le m$, and we construct the complex Clifford algebra as its complexification $\mathbb{C}_m := \mathbb{G}_m \oplus i\mathbb{G}_m$, i.e. any element $\varphi \in \mathbb{C}_m$ can be written as $\varphi = x + iy$, where $x, y \in \mathbb{G}_m$. The complex Clifford algebras for small m are well known; $\mathbb{C}_0$ are complex numbers itself, $\mathbb{C}_1$ is the algebra of bicomplex numbers and $\mathbb{C}_2$ is the algebra of biquaternions. More details on complex Clifford algebras one can find in the papers [2–4, 10].

The construction via the complexification of $\mathbb{G}_m$ leads to the definition an important antiautomorphism of $\mathbb{C}_m$, so-called Hermitian conjugation

$$\varphi^\dagger = (x + iy)^\dagger = \bar{x} - i\bar{y}, \tag{2.1}$$

where the bar notation stands for the Clifford conjugation in $\mathbb{G}_m$. Note that on the zero grade part of the complex Clifford algebra $\mathbb{C}_0 = \mathbb{C}$ it coincides with the usual complex conjugation. The elements satisfying $\varphi^\dagger = \varphi$ and $\varphi^\dagger = -\varphi$ will be called Hermitian and anti-Hermitian respectively. Hermitian conjugation is a very important anti-involution which is the Clifford analogue of the conjugate transpose in matrices. It leads to the definition of the Hermitian inner product on $\mathbb{C}_m$ given by

$$\langle \varphi | \psi \rangle = [\varphi^\dagger \psi]_0, \quad \varphi, \psi \in \mathbb{C}_m \tag{2.2}$$

where we recall that $[\ ]_0$ denotes the projection to the scalar part, i.e. the grade zero part. Indeed, it is easy to see that it is linear in the second slot and conjugate linear in the first slot; for each $z \in \mathbb{C}$ and $\varphi \in \mathbb{C}_m$ we have $(z\varphi)^\dagger = \varphi^\dagger z^\dagger = \bar{z}\varphi^\dagger$ since the Hermitian conjugation is anti-automorphism. The Hermitian symmetry of (2.2) follows from the involutivity of the Hermitian conjugation while its positive definiteness follows from the fact that

$$\langle \varphi | \varphi \rangle = [\varphi^\dagger \varphi]_0 = \sum_A \varphi_A^2,$$

where A is an arbitrary multiindex and φ_A is the coefficient at the Grassmann blade e_A, i.e. $\varphi = \sum_A \varphi_A e_A$. Let us discuss the last equality in more detail. For two multiindices $A = \{i_1, \ldots, i_r\}$, $B = \{k_1, \ldots, k_s\}$ the scalar projection $[\bar{e}_A e_B]_0$ is nonzero only if the grades are equal, i.e. $r = s$. Then by the definition of geometric product and the reverse operation we get $[\tilde{e}_A e_B]_0 = (e_{i_1} \cdot e_{k_1}) \cdots (e_{i_r} \cdot e_{k_r}) = \delta_{i_1 k_1} \cdots \delta_{i_r k_r} = \delta_{AB}$, whence by the linearity of the grade projection

$$[\varphi^\dagger \varphi]_0 = \sum_{A,B} \bar{\varphi}_A \varphi_B [\tilde{e}_A e_B]_0 = \sum_A \varphi_A^2.$$

2.3 Witt basis

Henceforth we assume the dimension of the generating vector space is even, i.e. $m = 2n$. In such a case the complexification of the Clifford algebra can be introduced by considering so-called complex structure, i.e. a specific orthogonal linear transformation $J : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ such that $J^2 = -1$, where 1 stands for the identity map. Namely, we choose J such that its action on the orthonormal basis $e_1, \dots, e_{2n}$ is given by $J(e_j) = -e_{j+n}$ and $J(e_{j+n}) = e_j$, $j = 1, \dots, n$. With J one may associate two projection operators which produce the main objects of the complex setting by acting on the orthonormal basis, so-called Witt basis elements $(f_j, f_j^\dagger)$. Namely, we define

$$f_j = \frac{1}{2}(1 + iJ)(e_j) = \frac{1}{2}(e_j - ie_{j+n}), \quad j = 1, \dots, n$$
$$f_j^\dagger = \frac{1}{2}(1 - iJ)(e_j) = \frac{1}{2}(e_j + ie_{j+n}), \quad j = 1, \dots, n$$

Note that it is not confusion of the notation since $f_j^\dagger$ indeed is the image of f_j under Hermitian conjugation (2.1). The Witt basis elements are isotropic with respect to the geometric product, i.e. for each $j = 1, \dots, n$ they satisfy $f_j^2 = 0$ and $f_j^{\dagger 2} = 0$. They also satisfy the Grassmann identities

$$f_j f_k + f_k f_j = f_j^\dagger f_k^\dagger + f_k^\dagger f_j^\dagger = 0, \quad j, k = 1, \dots, n \tag{2.3}$$

and the duality identities

$$f_j f_k^\dagger + f_k^\dagger f_j = \delta_{jk}, \quad j, k = 1, \dots, n \tag{2.4}$$

The Witt basis of the whole complex Clifford algebra $\mathbb{C}_{2n}$ is then obtained, similarly to the basis of the real Clifford algebra, by taking the 2^{2n} possible geometric products of Witt basis vectors, i.e. it is formed by elements

$$(f_1)^{i_1} (f_1^\dagger)^{j_1} \cdots (f_n)^{i_n} (f_n^\dagger)^{j_n}, \quad i_k, j_k \in \{0, 1\} \text{ for } k = 1, \dots, n \tag{2.5}$$

One can eventually use the Grassmann blades of Witt elements as the basis of $\mathbb{C}_{2n}$. The relation of the two basis can be deduced from the relation of the geometric product of Witt basis elements to the corresponding inner and outer product, for more details see [3, 20].

$$f_j f_k = f_j \cdot f_k + f_j \wedge f_k = f_j \wedge f_k$$

$$f_j^\dagger f_k^\dagger = f_j^\dagger \cdot f_k^\dagger + f_j^\dagger \wedge f_k^\dagger = f_j^\dagger \wedge f_k^\dagger$$
$$f_j f_k^\dagger = f_j \cdot f_k + f_j \wedge f_k^\dagger = \frac{1}{2}\delta_{jk} + f_j \wedge f_k^\dagger$$

2.4 Spinor spaces

In the language of Clifford algebras, spinor space is defined as a minimal left ideal of the complex Clifford algebra and is realized explicitly by means of a self-adjoint primitive idempotent. The realization of spinor space within the complex Clifford algebra $\mathbb{C}_{2n}$ can be constructed directly using the Witt basis as follows. We start by defining

$$I_j = f_j f_j^\dagger \text{ and } K_j = f_j^\dagger f_j, \quad j = 1, \ldots, n$$

Direct computations show that both I_j, K_j are mutually commuting self-adjoint idempotents. More precisely, for $j, k = 1, \ldots, n$ the following identities hold.

$$\begin{aligned} I_j^\dagger = I_j,\ I_j^2 = I_j \text{ and } K_j^\dagger = K_j,\ K_j^2 = K_j, \\ I_j I_k = I_k I_j,\ K_j K_k = K_k K_j, \\ I_j K_k = K_k I_j \text{ whenever } j \neq k, \text{ and } I_j K_j = K_j I_j = 0. \end{aligned}$$

Moreover, the duality relations (2.4) between Witt basis vectors imply that $I_j + K_j = 1$ for each $j = 1, \ldots, n$. Hence we get the resolution of the identity $1 = \prod_{j=1}^n (I_j + K_j)$. Consequently we get

$$\mathbb{C}_{2n} = \mathbb{C}_{2n} \prod_{j=1}^{n} (I_j + K_j) = \mathbb{C}_{2n} I_1 \cdots I_n \oplus \mathbb{C}_{2n} I_1 \cdots I_{n-1} K_1 \oplus \cdots \oplus \mathbb{C}_{2n} K_1 \cdots K_n,$$

a direct sum decomposition of the complex Clifford algebra into 2^n isomorphic realizations of the spinor space that are denoted according to the specific idempotent involved:

$$\mathbb{S}_{\{i_1 \ldots i_s\}\{k_1 \ldots k_t\}} = \mathbb{C}_{2n} I_{i_1} \ldots I_{i_s} K_{k_1} \ldots K_{k_t} \subset \mathbb{C}_{2n}, \tag{2.6}$$

where $s + t = n$ and the indices are pairwise different. Each such space has dimension 2^n and its basis is obtained by right multiplication of the basis of $\mathbb{C}_{2n}$ by the corresponding primitive idempotent $I_{i_1} \cdots I_{i_s} K_{k_1} \cdots K_{k_t}$. By the basic properties of the Witt basis elements (2.3) and (2.4) it is easy to see that this action is nonzero if and only if the element of $\mathbb{C}_{2n}$ actually lies in the Grassmann algebra generated by n-dimensional space $(f_{i_1}^\dagger, \ldots, f_{i_s}^\dagger, f_{k_1}, \ldots, f_{k_t})$, i.e. we may write

$$\mathbb{S}_{\{i_1 \cdots i_s\}\{k_1 \cdots k_t\}} = \Lambda(f_{i_1}^\dagger, \ldots, f_{i_s}^\dagger, f_{k_1}, \ldots, f_{k_t}) I_{i_1} \cdots I_{i_s} K_{k_1} \cdots K_{k_t}. \tag{2.7}$$

In terms of multiindices $A = \{i_1, \ldots, i_s\}$, $B = \{k_1, \ldots, k_t\}$ this spinor space can be written shortly as $\mathbb{S}_{AB}$. It is an easy observation that it has the structure of a Hilbert space of dimension 2^n due to the Hermitian product (2.2) and that the multiplication in $\mathbb{C}_{2n}$ makes each spinor space $\mathbb{S}_{AB}$ into a left $\mathbb{C}_{2n}$-module. Hence the elements of the complex Clifford algebra that keep the Hermitian product invariant define a representation of the corresponding unitary group on the spinor space. Namely let a $\lambda \in \mathbb{C}_{2n}$ act on two spinors $\varphi, \psi \in \mathbb{S}_{AB}$. Then we compute $\langle \lambda\varphi|\lambda\psi\rangle = [\varphi^\dagger\lambda^\dagger\lambda\psi]_0$ by definition and due to the antiautomorphism property of the Hermitian conjugation. Hence the elements of the complex Clifford algebra such that

$$\lambda^\dagger\lambda = 1 \tag{2.8}$$

holds keep the Hermitian product invariant and thus define a representation of the unitary group $U(2^n)$ on the spinor space $\mathbb{S}_{AB}$. These elements also satisfy $\lambda\lambda^\dagger = 1$ and will be called unitary elements of $\mathbb{C}_{2n}$ in analogy with unitary matrices. Let us remark that in the representation theory this representation of the unitary group is well known. It comes from the so called spin representation of the corresponding complex orthogonal group.

3 Quantum computing in complex Clifford algebras

The idea is to perform quantum computing in the Hilbert space defined by a complex Clifford algebra with Hermitian product defined by (2.2) instead of the classical realization of the Hilbert space on complex coordinate space with the standard Hermitian inner product. A quantum state is then represented by an element of a complex Clifford algebra lying in spinor space (2.6) and unitary transformations are then realized as elements (2.8) of the same algebra. The computation becomes especially efficient when using Witt basis of the complex Clifford algebra, see 2.3. However the mathematical framework described in the previous section allows for a direct application to general states and transformations of multiple qubits, for clarity we start with the description of the basic case of a single qubit.

3.1 A qubit and single qubit gates

A qubit will be represented by an element in the complex Clifford algebra $\mathbb{C}_2$ instead of its standard representation by a vector in the complex coordinate space $\mathbb{C}^2$. The Witt basis elements $(f, f^\dagger)$ satisfy the Grassmann and duality identities

$$f^2 = f^{\dagger 2} = 0, \; ff^\dagger + f^\dagger f = 1,$$

leading to $ff^\dagger = \frac{1}{2} + f \wedge f^\dagger$ and $[ff^\dagger]_0 = \frac{1}{2}$ in particular. The Witt basis vectors induce a basis of the complex Clifford algebra $\mathbb{C}_2$ of the form $(1, f, f^\dagger, ff^\dagger)$. In this algebra we have two primitive idempotents $I = ff^\dagger$ and $K = f^\dagger f$ that give rise to two isomorphic spinor spaces: $\mathbb{S} = \mathbb{C}_2 I = \Lambda(f^\dagger)I$ and $\bar{\mathbb{S}} = \mathbb{C}_2 K = \Lambda(f)K$. For representing the qubit we choose the former one, see Remark 3.1. Choosing the basis $(1, f^\dagger)$ of the Grassmann algebra $\Lambda(f^\dagger)$, we get the following basis of $\mathbb{S}$ that will represent the zero state and the one state of a qubit

$$\begin{aligned} |0\rangle &= I = ff^\dagger \\ |1\rangle &= f^\dagger I = f^\dagger ff^\dagger = (1 - ff^\dagger)f^\dagger = f^\dagger, \end{aligned} \tag{3.1}$$

whence the Clifford algebra representation of a qubit in a general superposition state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, for arbitrary complex numbers α, β, is given by

$$\psi = (\alpha + \beta f^\dagger)I = \alpha ff^\dagger + \beta f^\dagger \in \mathbb{S} \subset \mathbb{C}_2. \tag{3.2}$$

However the basis (3.1) of spinor space $\mathbb{S}$ is orthogonal with respect to the Hermitian product in Clifford algebra $\mathbb{C}_2$ defined by (2.2), it is not orthonormal since the length of the basis elements equals $1/2$ due to the spinorial nature of the representation. To make the basis orthonormal we will modify the Hermitian product in (2.2) by this factor, namely we will assume $\langle\varphi|\psi\rangle = 2[\varphi^\dagger\psi]_0$ for any spinors $\varphi, \psi \in \mathbb{C}_2$. Indeed, then we compute

$$\begin{aligned} \langle 0|1\rangle &= 2[ff^\dagger f^\dagger]_0 = 0, \\ \langle 0|0\rangle &= \langle 1|1\rangle = 2[ff^\dagger]_0 = 1. \end{aligned}$$

Remark 3.1 The choice of idempotent I is motivated by conventions in physics for creation and annihilation operators. Indeed, $f^\dagger$ is a realization of the abstract creation operator of the so-called CAR algebra and thus we want it to represent the qubit state $|1\rangle$ rather than $|0\rangle$.

A single qubit gate is represented by an unitary element in Clifford algebra $\mathbb{C}_2$ and it acts on a qubit in spinor space $\mathbb{S}$ by left multiplication. Obviously the identity gate is defined by $1 \in \mathbb{C}_2$ and serially wired gates are given by the product of the individual representatives in $\mathbb{C}_2$ due to the associativity of the Clifford product. For our choice of the basis of qubit states the commonly used quantum gates operating on a single qubit are represented in terms of the Witt basis as follows.

Proposition 3.2 *Representing the basic qubit states in the complex Clifford algebra* $\mathbb{C}_2$ *as* $|0\rangle = ff^\dagger$, $|1\rangle = f^\dagger$
we get representations of single qubit gates in $\mathbb{C}_2$

$$\begin{aligned} \textit{X-gate: } & \lambda_X = f^\dagger + f \\ \textit{Y-gate: } & \lambda_Y = if^\dagger - if \\ \textit{Z-gate: } & \lambda_Z = ff^\dagger - f^\dagger f \end{aligned}$$

Proof By the definition of Hermitian conjugation in Clifford algebra $\mathbb{C}_2$, the identification of basic qubit states $|0\rangle = ff^\dagger$, $|1\rangle = f^\dagger$ leads to the identification of their

Hermitian duals $\langle 0| = ff^\dagger$, $\langle 1| = f$. Then we get a representation of projection operators

$$|0\rangle\langle 0| = ff^\dagger ff^\dagger = ff^\dagger,\ |0\rangle\langle 1| = ff^\dagger f = f,\ |1\rangle\langle 0| = f^\dagger ff^\dagger = f^\dagger,\ |1\rangle\langle 1| = ff^\dagger,$$

where we used the Grassmann and duality identities for the Witt basis elements $f, f^\dagger$. The representations of single qubit gates from the proposition then follow by their definitions on basic qubit states. ◻

Remark 3.3 Using basis $1, e_1, e_2, e_1 \wedge e_2$ of Clifford algebra $\mathbb{C}_2$ generated by an orthonormal basis e_1, e_2 of $\mathbb{C}^2$ instead of the Witt basis the representation of single qubit gates X, Y and Z are given as

$$\lambda_X = e_1,\ \lambda_Y = -e_2,\ \lambda_Z = ie_1 \wedge e_2$$

Example 3.4 Let us discuss the some of these basic quantum gates in more detail. The X-gate is the quantum equivalent to of the NOT gate for classical computers, sometimes called a bit-flip as it maps the basis state $|0\rangle$ to $|1\rangle$ and vice versa. Hence we have

$$X = |1\rangle\langle 0| + |0\rangle\langle 1| = f^\dagger + f.$$

which is equal to e_1 by definition of the Witt basis elements. We can also compute directly the action of the corresponding element of $\mathbb{C}_2$ on a qubit basis in $\mathbb{S}$ to prove the correctness of the representation

$$\begin{aligned}\lambda_X|0\rangle &= (f^\dagger + f)ff^\dagger = f^\dagger ff^\dagger = f^\dagger = |1\rangle,\\ \lambda_X|1\rangle &= (f^\dagger + f)f^\dagger = ff^\dagger = |0\rangle.\end{aligned}$$

Similarly, for the phase-flip Z-gate we get $Z = |0\rangle\langle 0| - |1\rangle\langle 1| = ff^\dagger - f^\dagger f$ since it leaves the basis state $|0\rangle$ unchanged and maps $|1\rangle$ to $-|1\rangle$ and a general phase-shift gate $|1\rangle \mapsto e^{i\varphi}|1\rangle$ is given by $R_\varphi = ff^\dagger + e^{i\varphi}f^\dagger f$. The effect of a series circuit where X is put after Z can be described as a single gate represented by the Clifford product

$$XZ = (f^\dagger + f)(ff^\dagger - f^\dagger f) = f^\dagger - f.$$

It is also easy to check the involutivity of the single qubit gates X, Y and Z, e.g. for a serial composition of two X-gates we have

$$X^2 = (f^\dagger + f)(f^\dagger + f) = f^\dagger f + f^\dagger f = 1.$$

The representations of rotation operator gates can be obtained directly by computing exponentials of gates X, Y, Z in $\mathbb{C}_2$. Consequently one can get the formula for the

Hadamard gate, an important single qubit gate that we have not discussed yet. Namely, since $Y^2 = 1$, we compute

$$\begin{aligned} H &= X \exp(-iY\pi/4) = (f^\dagger + f)(\cos\tfrac{\pi}{4} + \sin\tfrac{\pi}{4}(f^\dagger - f)) \\ &= \tfrac{1}{\sqrt{2}}(ff^\dagger - f^\dagger f + f + f^\dagger), \end{aligned}$$

which is equal to $\frac{1}{\sqrt{2}}(e_1 + ie_1 \wedge e_2)$ in the orthonormal basis.

The Clifford algebra representations of basic single qubit gates X, Y and Z in Proposition 3.2 determine an explicit form of a unitary element in $\mathbb{C}_2$ representing a general single qubit gate in terms of the Witt basis.

Corollary 3.5 *Each single qubit gate operating on qubit* (3.2) *is represented by an element of the complex Clifford algebra* $\mathbb{C}_2$ *of a form*

$$\lambda = aff^\dagger + bf + cf^\dagger + df^\dagger f. \tag{3.3}$$

where $a, b, c, d \in \mathbb{C}$ *are complex numbers such that* $a^2+c^2 = b^2+d^2 = 1$, $\bar{b}a+\bar{d}c = 0$ *holds.*

Proof Can be deduced from the description of projection operators in the proof of Proposition 3.2 or directly by writing the condition on unitary elements $\lambda^\dagger\lambda = 1$ in terms of the Witt basis, see (2.8), as follows. An arbitrary element $\lambda \in \mathbb{C}_2$ can be written as in (3.3) for some complex numbers $a, b, c, d \in \mathbb{C}$ since the four-tuple $(ff^\dagger, f^\dagger, f, f^\dagger f)$ form a basis of $\mathbb{C}_2$. The right-hand side of the equation for unitary elements can be written as $ff^\dagger + f^\dagger f$ and for the left-hand side we compute

$$\begin{aligned} \lambda^\dagger\lambda &= (\bar{a}ff^\dagger + \bar{b}f^\dagger + \bar{c}f + \bar{d}f^\dagger f)(aff^\dagger + bf + cf^\dagger + df^\dagger f) \\ &= (a^2 + c^2)f^\dagger f + (\bar{b}a + \bar{d}c)f^\dagger + (\bar{a}b + \bar{c}d)f + (b^2 + d^2)ff^\dagger, \end{aligned}$$

where we used the definition of the Hermitian conjugation and its properties discussed in Sect. 2.2 and where we repeatedly used the duality and Grassmann identities for the Witt basis elements $f, f^\dagger$, see Sect. 2.3. The result follows by comparing the coefficients on both sides of the equation while noting that $\bar{a}b + \bar{c}d$ is the complex conjugate of $\bar{b}a + \bar{d}c$. □

Note that the condition on these coefficients can be expressed equivalently as the orthonormality of complex vectors (a, c) and (b, c) with respect to the standard Hermitian product on $\mathbb{C}^2$. Hence the coefficients define a 2×2 unitary matrix proving the equivalence between Clifford and matrix descriptions. Namely, each unitary element in $\mathbb{C}_2$ corresponds to a matrix in $U(2)$ as follows.

$$aff^\dagger + bf + cf^\dagger + df^\dagger f \leftrightarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{3.4}$$

Remark 3.6 Upon restriction to a normalized qubit $\langle\psi|\psi\rangle = 2[\psi^\dagger\psi]_0 = \alpha^2+\beta^2 = 1$, it is sufficient to consider gates from the special unitary group $SU(2)$, the connected component of $U(2)$. Such gates are represented by unitary matrices with unite determinant and that they can be written as displayed in (3.4) for $b = -\bar{c}$ and $d = \bar{a}$. Hence the elements in $\mathbb{C}_2$ representing subgroup $SU(2)$ are of a form $\lambda = aff^\dagger - \bar{c}f + cf^\dagger + \bar{a}f^\dagger f$.

3.2 Multiple qubits and multiple qubit gates

Following constructions in Sect. 2 the Hilbert space of states of a general n-qubit can be represented by a spinor space in the complex Clifford algebra $\mathbb{C}_{2n}$ and n-qubit gates as unitary elements in the same algebra. For the explicit description we choose Witt basis $(f_1, f_1^\dagger, \dots, f_n, f_n^\dagger)$ of complex coordinate space $\mathbb{C}^{2n}$ leading to the Witt basis of the Clifford algebra $\mathbb{C}_{2n}$ formed by 2^{2n} geometric products of these elements given by (2.5). In this way we get a realization of the standard Fock basis of spinor space [5, 11]. From the 2^n spinor spaces contained in the algebra we choose the spinor space $\mathbb{S}_n = \mathbb{C}_{2n}I$ defined by primitive idempotent

$$I = I_1 \cdots I_n = f_1 f_1^\dagger \cdots f_n f_n^\dagger \tag{3.5}$$

for modelling states of a n-qubit. This choice is motivated by its identification with the Grassmann algebra generated by “creation operators” $f_1^\dagger, \dots, f_n^\dagger$. Indeed, for such a realization of the spinor space of a n-qubit we have

$$\mathbb{S}_n = \mathbb{C}_{2n}I = \Lambda(f_1^\dagger, \dots, f_n^\dagger)I \tag{3.6}$$

since $f_j I = 0$ for each $j = 1, \dots, n$ by Grassmann and duality identities for the Witt basis elements. Similarly to the case of a single qubit, we multiply the Hermitian product (2.2) by a normalization factor 2^n that reflects the spinorial nature of our representation in order to get a simple formula for elements of unite norm. Namely, for two spinors $\varphi, \psi \in \mathbb{S}_n$ we set

$$\langle\varphi|\psi\rangle = 2^n[\varphi^\dagger\psi]_0. \tag{3.7}$$

With this choice of spinor space and Hermitian product the main results of Sect. 2 that we need for representing qubits and quantum gates in a complex Clifford algebra read as follows.

Proposition 3.7 *Spinor space $\mathbb{S}_n \subset \mathbb{C}_{2n}$ given by (3.6) together with Hermitian product (3.7) form a Hilbert space of dimension $N = 2^n$ with an orthonormal basis*

$$|i_1 \cdots i_n\rangle = (f_1^\dagger)^{i_1} \cdots (f_n^\dagger)^{i_n} I, \tag{3.8}$$

where $i_1, \dots, i_n \in \{0, 1\}$. Unitary transformations are given by left multiplications by unitary elements, i.e. elements of $\mathbb{C}_{2n}$ such that $\lambda^\dagger\lambda = 1$.

Proof The proposition follows from the constructions described in Sect. 2, we only need to check the orthonormality of basis elements (3.8). For $i_k, j_k \in \{0, 1\}$, where $k = 1, \dots, n$, the Hermitian product of two basis elements is given by

$$\langle j_1 \cdots j_n | i_1 \cdots i_n \rangle = 2^n [I^\dagger (f_n)^{j_n} \cdots (f_1)^{j_1} (f_1^\dagger)^{i_1} \cdots (f_n^\dagger)^{i_n} I]_0$$

by definition. Let us prove the orthogonality first. Assume $i_k = 0$ and $j_k = 1$ for some k. Since the Witt basis element f_k anti-commutes with elements f_ℓ and $f_\ell^\dagger$ for each $\ell \neq k$ by (2.3), the previous formula can be expressed in a form $\pm 2^n [I^\dagger \cdots f_k I]_0$ and it vanishes since $f_k I = 0$ by (2.3) and (2.4). Similarly, if $i_k = 1$ and $j_k = 0$, then the above formula for Hermitian product vanishes since it contains factor $I^\dagger f_k^\dagger = 0$. Hence the Hermitian product vanish if $i_k \neq j_k$ for some k and the the orthogonality is proven. To prove the normality we notice that $(f_k)^{i_k} (f_k^\dagger)^{i_k} = I_{i_k}$ is an idempotent commuting with all elements $f_\ell, f_\ell^\dagger$, where $\ell \neq k$, and all idempotents I_{i_ℓ}. Hence we get

$$\langle i_1 \cdots i_n | i_1 \cdots i_n \rangle = 2^n [I^\dagger (f_n)^{i_n} \cdots (f_1)^{i_1} (f_1^\dagger)^{i_1} \cdots (f_n^\dagger)^{i_n} I]_0 = 2^n [I^\dagger \cdots I_{i_\ell} \cdots I]_0,$$

where I_{i_ℓ} are idempotents such that $i_\ell = 1$. The primitive idempotent I satisfies $I^\dagger = I$ and it also satisfies $I_{i_\ell} I = I$ for all ℓ since it is given by product of all such commuting idempotents, namely $I = I_1 \cdots I_n$ by definition. So we compute

$$\langle i_1 \cdots i_n | i_1 \cdots i_n \rangle = 2^n [I]_0 = 2^n [I_1 \cdots I_n]_0 = 1,$$

where the last equality follows from the decomposition of idempotents into grade components. Namely, we have $I_k = 1/2 + f_k \wedge f_k^\dagger$ and the geometric product of $f_k \wedge f_k^\dagger$ with elements not containing f_k neither $f_k^\dagger$ either vanishes or yields an element of grade at least two. □

Remark 3.8 Note that we chose MSB bit numbering for n-qubits. The choice of the LSB bit numbering would lead to different but isomorphic representations in $\mathbb{C}_{2n}$.

However an explicit description of a general unitary element in $\mathbb{C}_{2n}$ representing a n-qubit gate similar to the description of a general single qubit gate given in Corollary 3.5 is possible, it is more sophisticated and thus not helpful. The same happens in matrix representation and it reflects the complexity of unitary group $U(N)$. On the other hand, given a specific n-qubit gate its representation in $\mathbb{C}_{2n}$ is obtained by rewriting the defining formula in terms of projection operators in Dirac formalism via identification of n-qubit states (3.8). To make it clear we elaborate some examples for $n = 2$ in more detail.

Example 3.9 In the case of a 2-qubit we work in Clifford algebra $\mathbb{C}_4$ of dimension is $2^4 = 16$. Using the Witt basis $f_1, f_1^\dagger, f_2, f_2^\dagger$ of $\mathbb{C}^4$ we define a primitive idempotent $I = f_1 f_1^\dagger f_2 f_2^\dagger$ which gives rise to spinor space $\mathbb{S}_2 = \mathbb{C}_4 I$ of dimension $2^2 = 4$ with an orthonormal basis

$$|00\rangle = I,\ |10\rangle = f_1^\dagger I,\ |01\rangle = f_2^\dagger I,\ |11\rangle = f_1^\dagger f_2^\dagger I$$

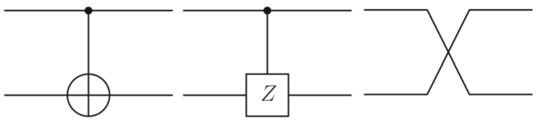


Fig. 2 2-qubit gates CNOT, CZ and SWAP respectively

Using this representation of basis states and the definition of Hermitian conjugation we can form 16 projection operators, e.g. $|00\rangle\langle 00| = II^\dagger = I$, $|00\rangle\langle 01| = II^\dagger f_2 = If_2 = f_1 f_1^\dagger f_2$, etc. Specific 2-qubit gates are then formed by a complex linear combinations of these elements in $\mathbb{C}_4$. We demonstrate the functionality of the spinor representation on 2-qubit gates known as CNOT, CZ and SWAP [21], see the diagrammatic descriptions of these gates in Fig. 2.

$$
\begin{aligned}
f_1 f_1^\dagger |00\rangle &= f_1 f_1^\dagger f_1 f_1^\dagger f_2 f_2^\dagger = I = |00\rangle,\\
f_1 f_1^\dagger |01\rangle &= f_1 f_1^\dagger f_2^\dagger f_1 f_1^\dagger f_1 f_1^\dagger = f_2^\dagger I = |01\rangle
\end{aligned}
$$

We systematically use Grassmann and duality identities for Witt basis elements, identity $II^\dagger = I$ in particular.

$$
\begin{aligned}
\lambda_{\mathrm{CNOT}} &= |00\rangle\langle 00| + |01\rangle\langle 01| + |11\rangle\langle 10| + |10\rangle\langle 11|\\
&= I + f_2^\dagger I f_2 + f_1^\dagger f_2^\dagger I f_1 + f_1^\dagger I f_2 f_1\\
&= f_1 f_1^\dagger f_2 f_2^\dagger + f_1 f_1^\dagger f_2^\dagger f_2 - f_1^\dagger f_1 f_2^\dagger - f_1^\dagger f_1 f_2\\
&= f_1 f_1^\dagger - f_1^\dagger f_1 (f_2^\dagger + f_2),\\
\lambda_{\mathrm{CZ}} &= |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10| - |11\rangle\langle 11|\\
&= f_1 f_1^\dagger + f_1^\dagger f_1 (f_2 f_2^\dagger - f_2^\dagger f_2),\\
\lambda_{\mathrm{SWAP}} &= |00\rangle\langle 00| + |11\rangle\langle 11| + |10\rangle\langle 01| + |01\rangle\langle 11|\\
&= f_1 f_1^\dagger f_2 f_2^\dagger + f_1^\dagger f_1 f_2^\dagger f_2 + f_1^\dagger f_2 - f_1 f_2^\dagger.
\end{aligned}
$$

3.3 Tensor product of gates

To describe effectively quantum logic circuits in the complex Clifford algebra, it remains to discuss representations of parallel quantum gates, i.e. representations of tensor product of gates. First of all we realize that the representation of tensor products of states is already determined by Proposition 3.7. Namely, a n-qubit $|i_1 \cdots i_n\rangle = |i_1\rangle \otimes \cdots \otimes |i_n\rangle$ is in the Clifford algebra represented by geometric product of representations of individual qubits. Hence a tensor product $|\varphi\rangle \otimes |\psi\rangle$ is represented by geometric product $\varphi\psi$, where the spinors φ, ψ are assumed to lie in disjoint vector spaces viewed as two orthogonal subspaces of their union. Now consider an action of a tensor product of gates $\lambda \otimes \mu$ given by unitary elements λ, μ of the Clifford algebra on such a state. The resulting state $\lambda\varphi \otimes \mu\psi$ is represented by $\lambda\varphi\mu\psi$ which is different from $\lambda\mu\varphi\psi$ in general due to the skew-symmetry of the geometric product. Namely, for two blades

e_A, e_B determined by disjoint multi-indices A, B we have

$$e_A e_B = (-1)^{|A||B|} e_B e_A.$$

and so the Clifford algebra has the structure of a superalgebra. Hence the geometric product does not represent the ordinary tensor product but it represents the super tensor product. It has the same structure as a vector space but with the multiplication rule determined by

$$(e_A e_B)(e_C e_D) = (-1)^{|B||C|}(e_A e_C)(e_B e_D) \tag{3.9}$$

on blades. Consequently, the geometric product identifies complex Clifford algebra for n-gates with the super tensor product of Clifford algebras for single qubit gates. The ordinary ungraded tensor product $\lambda_1 \otimes \cdots \otimes \lambda_n$ of gates λ_k from distinct copies of $\mathbb{C}_2$ is in Clifford algebra $\mathbb{C}_{2n}$ represented by $\lambda_1 \cdots \lambda_n$ only up to the sign. Although this sign depends on the n-qubit on which we act by (3.9) in general, it is completely determined by the set $\{\lambda_1, \ldots, \lambda_n\}$ in the case that λ_k for each $k = 1, \ldots, n$ is one of the basis elements of $\mathbb{C}_2$

$$\lambda_k \in \{f_k f_k^\dagger, f_k^\dagger f_k, f_k, f_k^\dagger\}. \tag{3.10}$$

Proposition 3.10 *A tensor product $\lambda_1 \otimes \cdots \otimes \lambda_n$, where $\lambda_k \in \{f_k f_k^\dagger, f_k^\dagger f_k, f_k, f_k^\dagger\}$ for each $k = 1, \ldots, n$, is represented by geometric product $(-1)^s \lambda_1 \cdots \lambda_n$, where the sign is determined by the cardinality of the sets S_i, such that $s = \sum_i |S_i|$, where*

$$S_i = \{\ell < i \ : \ \lambda_l = f_\ell \text{ or } \lambda_\ell = f_\ell^\dagger f_\ell\} \text{ in the case if } \lambda_i = f_i \text{ or } \lambda_i = f_i^\dagger. \tag{3.11}$$

Proof A n-qubit $\psi_1 \otimes \cdots \otimes \psi_n$ is represented by geometric product $\psi_1 \cdots \psi_n$ of mutually orthogonal spinors ψ_k by (3.8). Representation of a qubit obtained upon the action of $\lambda_1 \otimes \cdots \otimes \lambda_n$ on this n-qubit is given by

$$\lambda_1 \psi_1 \cdots \lambda_n \psi_n = (-1)^p \lambda_1 \cdots \lambda_n \psi_1 \cdots \psi_n$$

since λ_k, ψ_k are orthogonal to λ_ℓ, ψ_ℓ for $k \neq \ell$. Roughly speaking, the sign is determined by how many times we need to commute to get all elements λ_k to the left hand side. Thus it depends on spinors ψ_ℓ, $\ell < k$ which are combinations of components of grade one or two in general. Only the commuting with grade one components does change the sign. However the grade one spinors are multiples of $f_\ell^\dagger$ and they are annihilated by all elements of $\mathbb{C}_{2n}$ except elements $\lambda_\ell = f_\ell$ and $\lambda_\ell = f_\ell^\dagger f_\ell$ which act nontrivially. Hence the number of commutation steps that push λ_k to the left is equal to the number of such elements λ_ℓ, $\ell < k$. □

Example 3.11 Let us construct 2-qubit gates $X \otimes Y$ and $Y \otimes X$ according to Proposition 3.10. First we write these gates as a sum of tensor products of basis Witt basis elements and then for each such summand we compute the cardinality of set S giving the sign of the corresponding geometric product.

$$\begin{aligned} X \otimes Y &= i(f_1^\dagger \otimes f_2^\dagger - f_1^\dagger \otimes f_2 + f_1 \otimes f_2^\dagger - f_1 \otimes f_2) \\ &= i(f_1^\dagger f_2^\dagger - f_1^\dagger f_2 - f_1 f_2^\dagger + f_1 f_2) \\ Y \otimes X &= i(f_1^\dagger \otimes f_2^\dagger + f_1^\dagger \otimes f_2 - f_1 \otimes f_2^\dagger - f_1 \otimes f_2) \\ &= i(f_1^\dagger f_2^\dagger + f_1^\dagger f_2 + f_1 f_2^\dagger + f_1 f_2) \end{aligned}$$

Let us assume even more simple example of a X-gate with a parallel qubit without any gate. If the gate is acting on the first qubit we get a resulting 2-qubit gate $X \otimes \mathrm{id} = X_1$. However, acting on the second qubit we need to write the identity representation as $1 = f_1 f_1^\dagger + f_1^\dagger f_1$ since idempotent $K_1 = f_1^\dagger f_1$ makes the change of sign in contrast to idempotent $I_1 = f_1 f_1^\dagger$,

$$\mathrm{id} \otimes X = (I_1 + K_1) \otimes X_2 = I_1 X_2 - K_1 X_2 = f_1 f_1^\dagger (f_2^\dagger + f_2) - f_1^\dagger f_1 (f_2^\dagger + f_2).$$

The representations of controlled gates from Example 3.9 can be constructed from tensor product of single qubit gates as follows.

$$\begin{aligned} \lambda_{\mathrm{CNOT}} &= I_1 \otimes 1 + K_1 \otimes X_2 = I_1 - K_1 X_2 \\ \lambda_{\mathrm{CZ}} &= I_1 \otimes 1 + K_1 \otimes Z_2 = I_1 + K_1 Z_2 \end{aligned}$$

Example 3.12 The spinor space $\mathbb{S}_3$ representing states of 3-qubits has dimension $2^3 = 8$ in Clifford algebra $\mathbb{C}_6$ of dimension $2^6 = 64$. Using the primitive idempotent (3.5) and the orthonormal basis representation (3.8) in terms of the Witt basis the Toffoli gate, see [21], is represented by

$$\begin{aligned} \lambda_{\mathrm{CCNOT}} &= (I_1 \otimes I_2 + I_1 \otimes K_2 + I_2 \otimes K_1) \otimes \mathrm{id} + K_1 \otimes K_2 \otimes X_3 \\ &= I_1 I_2 + I_1 K_2 + I_2 K_1 + K_1 K_2 X_3 = 1 - K_1 K_2 + K_1 K_2 X_3 \\ &= 1 + f_1^\dagger f_1 f_2^\dagger f_2 (f_3 + f_3^\dagger - 1). \\ \lambda_{\mathrm{CSWAP}} &= I_1 \otimes \mathrm{id} + K_1 \otimes \lambda_{\mathrm{SWAP}} \\ &= I_1 + K_1 (I_2 I_3 + K_2 K_3 + f_2^\dagger f_3 - f_2 f_3^\dagger) \\ &= f_1 f_1^\dagger + f_1^\dagger f_1 (f_2 f_2^\dagger f_3 f_3^\dagger + f_2^\dagger f_2 f_3^\dagger f_3 + f_2^\dagger f_3 - f_2 f_3^\dagger) \end{aligned}$$

4 Quantum computing in real Clifford algebras

Accidental isomorphism can be used to formulate intrinsically complex quantum computing in a real framework. We show two ways how to see a qubit in real Clifford algebra $\mathbb{G}_3$, i.e the GA induced by the standard euclidean inner product of signature $(3, 0)$. The first approach appears in literature, see [6, 9, 19], and describes qubit states as even elements in this algebra or equivalently as unite quaternions. The second approach is new and follows from the complex representation of qubits described above. For the other known concepts see [12, 15, 22]. We also mention how to deal with multiple qubits and multiple qubit gates in the real case.

4.1 A quaternionic qubit

The transition from complex to real framework which appears in literature is based on the well known coincidental isomorphism of Lie algebras $\mathfrak{su}(2) \cong \mathfrak{so}(3)$, or more precisely, on the corresponding isomorphism of Lie groups

$$SU(2) \cong Spin(3), \tag{4.1}$$

and the isomorphism of these groups with the group of unite quaternions. We can easily describe these isomorphisms explicitly by realizing Lie algebra $\mathfrak{so}(3)$ as bivectors in Clifford algebra $\mathbb{G}_3$ and Lie group $Spin(3)$ as elements of even grade in $\mathbb{G}_3$. Namely, in terms of Pauli matrices the Lie algebra isomorphism can be defined by mapping $i_{\mathbb{C}}\sigma_k \mapsto i\sigma_k$, $k = 1, 2, 3$, where we denote the usual complex unite by $i_{\mathbb{C}}$ in order to distinguish from pseudoscalar $i = \sigma_1\sigma_2\sigma_3$ in $\mathbb{G}_3$, while σ_k on the right hand side is seen as a vector in $\mathbb{G}_3$ satisfying $\sigma_k^2 = 1$. Consequently, using the Einstein summation convention, we get a Lie group isomorphism (4.1) of a form

$$\begin{pmatrix} a^0 + a^3 i_{\mathbb{C}} & a^2 + a^1 i_{\mathbb{C}} \\ -a^2 + a^1 i_{\mathbb{C}} & a^0 - a^3 i_{\mathbb{C}} \end{pmatrix} \mapsto a^0 + a^1\sigma_2\sigma_3 + a^2\sigma_3\sigma_1 + a^3\sigma_1\sigma_2 = a^0 + a^k\sigma_k^* \tag{4.2}$$

where the coefficients $a^0, a^1, a^2, a^3 \in \mathbb{R}$ and $\sigma_k^* = \sigma_k i = i\sigma_k$ is the duality defined by pseudoscalar $i = \sigma_1\sigma_2\sigma_3$. Assigning the quaternionic unites to σ_k, $k = 1, 2, 3$, defines an isomorphism with unite quaternions. A general state of a qubit is identified with the first column of the matrix on left hand side, thus in the real Clifford algebra $\mathbb{G}_3$ is represented by

$$|\psi\rangle = \begin{pmatrix} a^0 + a^3 i_{\mathbb{C}} \\ -a^2 + a^1 i_{\mathbb{C}} \end{pmatrix} \leftrightarrow \psi = a^0 + a^k\sigma_k^*$$

In particular, the standard computational basis $(1, 0)$ and $(0, 1)$ in $\mathbb{C}^2$ is in the real Clifford algebra formulation represented by

$$|0\rangle = 1 \text{ and } |1\rangle = -i\sigma_2 = \sigma_1\sigma_3, \tag{4.3}$$

respectively. The identification (4.2) also determines explicit formulas for a Hermitian inner product and a representation of Pauli matrices on even subalgebra $\mathbb{G}_3^0$, namely for $\varphi, \psi \in \mathbb{G}_3^0$ we have

$$\langle\varphi|\psi\rangle = [\tilde{\varphi}\psi]_0 - [\tilde{\varphi}\psi\sigma_1\sigma_2]_0 i_{\mathbb{C}}, \tag{4.4}$$

$$\hat{\sigma}_k|\psi\rangle \leftrightarrow \sigma_k\psi\sigma_3. \tag{4.5}$$

These formulas can be explained by viewing the unitary group $SU(2)$ as $SO(4) \cap GL(2, \mathbb{C})$, i.e. as the group of orthogonal transformations with respect to a real scalar product of signature $(4, 0)$ commuting with an orthogonal complex structure. A choice of a scalar product and a complex structure on even elements $\mathbb{G}_3^0$ then defines an Hermitian inner product on this space by a standard construction and thus defines an isomorphism (4.1). In our case, the scalar product is given by $(\varphi, \psi) = [\tilde{\varphi}\psi]_0$ and the complex structure J is defined by $J\psi = \psi i\sigma_3 = \psi\sigma_1\sigma_2$. Indeed, for such a choice the Hermitian product (4.4) is constructed as

$$\langle\varphi|\psi\rangle = (\varphi, \psi) - (\varphi, J\psi)i_{\mathbb{C}}.$$

The action of Pauli matrices in $\mathbb{G}_3^0$ given by (4.5) keep the scalar product invariant and commutes with the complex structure and thus keeps this Hermitian product invariant.

Remark 4.1 This point of view also allows to see the freedom of quaternionic representation of qubits. Namely, choosing a different complex structure or modifying the scalar product on $\mathbb{G}_3^0$ would lead to an isomorphism (4.1) different from (4.2) leading to representations of computational basis, Hermitian product and Pauli matrices different from (4.3), (4.4) and (4.5).

The reality of this qubit representation implies that multiple qubits are represented in a quotient space defined by so called correlator. Namely, representing qubits in the real geometric algebra $\mathbb{G}_3^+$ the space of n-qubits is $\mathbb{G}_3^+ \otimes \cdots \otimes \mathbb{G}_3^+$ instead the tensor power of n copies of $\mathbb{C}^2$. However this is the complex tensor product according to axions of the quantum mechanics. If we want to have a fully real description, including the real tensor product, we need to identify complex structures $J_k = i\sigma_3^k = \sigma_1^k\sigma_2^k$ (representing the multiplication by complex unite) in all copies. This can be done by introducing the n-qubit correlator

$$E_n = \prod_{k=2}^{n} \frac{1}{2}(1 - i\sigma_3^1 i\sigma_3^k).$$

Indeed, this element satisfies $E_n J_k = E_n J_\ell$ for all $k, \ell = 1, \dots, n$ and thus it defines a quotient space $\mathbb{G}_3^+ \otimes \cdots \otimes \mathbb{G}_3^+/E_n$ with a complex structure $J_n = E_n J_k = E_n i\sigma_3^k$. Multivectors belonging to this space can be regarded as n-qubit states.

4.2 A real complex qubit

Another way how to describe states of a qubit by a real algebra is to transfer its complex representation described in Sect. 3.1 via the accidental isomorphism of real algebras

$$\mathbb{C}_2 \cong \mathbb{G}_3. \tag{4.6}$$

In order to obtain an explicit representation of a qubit we choose a concrete realization of this isomorphism. Namely, in terms of the Witt basis of $\mathbb{C}_2$ and an orthonormal basis

σ_k of $\mathbb{R}^3$ we consider the isomorphism given by mapping

$$\begin{aligned} 1 &\mapsto 1 & i_{\mathbb{C}} &\mapsto \sigma_1\sigma_2\sigma_3 \\ f &\mapsto \tfrac{1}{2}(\sigma_1 - \sigma_1\sigma_3) & i_{\mathbb{C}} f &\mapsto \tfrac{1}{2}(\sigma_2\sigma_3 - \sigma_2) \\ f^\dagger &\mapsto \tfrac{1}{2}(\sigma_1 + \sigma_1\sigma_3) & i_{\mathbb{C}} f^\dagger &\mapsto \tfrac{1}{2}(\sigma_2\sigma_3 + \sigma_2) \\ ff^\dagger &\mapsto \tfrac{1}{2}(1 + \sigma_3) & i_{\mathbb{C}} ff^\dagger &\mapsto \tfrac{1}{2}(\sigma_1\sigma_2 + \sigma_1\sigma_2\sigma_3) \end{aligned}$$

In particular, the complex unite $i_{\mathbb{C}}$ is mapped to trivector $\sigma_1\sigma_2\sigma_3 \in \mathbb{G}_3$ and the primitive idempotent $I = ff^\dagger \in \mathbb{C}_2$ is mapped to real idempotent

$$I_{\mathbb{R}} = \frac{1}{2}(1 + \sigma_3) \in \mathbb{G}_3.$$

Using this idempotent the equivalence between the classical description of a qubit as a complex vector and as an element of $\mathbb{G}_3$ based on this isomorphism reads

$$|\psi\rangle = \begin{pmatrix} a^0 + a^3 i_{\mathbb{C}} \\ a^1 + a^2 i_{\mathbb{C}} \end{pmatrix} \leftrightarrow \psi = (a^0 + a^1\sigma_1 + a^2\sigma_2 + a^3\sigma_1\sigma_2)I_{\mathbb{R}},$$

where $a^0, a^1, a^2, a^3 \in \mathbb{R}$. Note that, in contrast to the quaternionic representation described in the previous section, a qubit is represented by a multivector in $\mathbb{G}_3$ containing blades of both even and odd grades in this case. In particular, the computational basis is given by

$$|0\rangle = I_{\mathbb{R}} \text{ and } |1\rangle = \sigma_1 I_{\mathbb{R}}. \tag{4.7}$$

The Hermitian inner product on $\mathbb{G}_3$ is given by the transition of the Hermitian product on $\mathbb{C}_2$ given by (2.2) via isomorphism (4.6). Looking at the prescription of the isomorphism we see that the Hermitian conjugation of basis elements is mapped to the reverse of the corresponding images in $\mathbb{G}_3$. Hence we have a particularly simple formula for the Hermitian product in this case, namely for two qubits $\varphi, \psi \in \mathbb{G}_3$ we have

$$\langle\varphi|\psi\rangle = [\tilde{\varphi}\psi]_0. \tag{4.8}$$

Our formula for isomorphism (4.6) yields also a particularly simple formula for the representation of Pauli matrices, namely

$$\hat{\sigma}_k|\psi\rangle \leftrightarrow \sigma_k\psi. \tag{4.9}$$

Although this real representation of a qubit is quite elegant, the representation of multiple qubits is as complicated as in the case of the quaternionic qubit described in 4.1. Due to its reality we need to use a correlator to identify the multiplication by complex unite in each slot of the tensor product $\mathbb{G}_3 \otimes \cdots \otimes \mathbb{G}_3$. Since this is a common feature of all real description of qubits we believe that the right way is to use the complex GA as described in Sect. 3 above.

Data Availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

1. Alves, R., Hildenbrand, D., Hrdina, J., et al.: An online calculator for quantum computing operations based on geometric algebra. Adv. Appl. Clifford Algebras **32**, 4 (2022)
2. Brackx, F., De Schepper, H., Sommen, F.: The Hermitean Clifford analysis toolbox. Adv. Appl. Clifford Algebras **18**(3–4), 451–487 (2008)
3. Brackx, F., De Schepper, H., Souček, V.: On the structure of complex Clifford algebra. Adv. Appl. Clifford Algebras **21**, 477–492 (2011)
4. Budinich, M.: On complex representations of Clifford algebra. Adv. Appl. Clifford Algebras (2019). https://doi.org/10.1007/s00006-018-0930-3
5. Budinich, P., Trautman, A.: Fock space description of simple spinors. J. Math. Phys. **30**(9), 2125–2131 (1989)
6. Cafaro, C., Mancini, S.: A geometric algebra perspective on quantum computational gates and universality in quantum computing. Adv. Appl. Clifford Algebras **21**, 493–519 (2011)
7. De Keninck, S.: Non-parametric realtime rendering of subspace objects in arbitrary geometric algebras. Lect. Notes Comput. Sci. **11542**, 549–555 (2019)
8. de Lima Marquezino, F., Portugal, R., Lavor, C.: A Primer on Quantum Computing. Springer, Berlin (2019)
9. Doran, C., Gullans, S.R., Lasenby, A., Lasenby, J., Fitzgerald, W.: Geometric Algebra for Physicists. Cambridge University Press, Cambridge (2003)
10. Ferreira, M., Sommen, F.: Complex boosts: a Hermitian Clifford algebra approach. Adv. Appl. Clifford Algebras **23**, 339–362 (2013)
11. Fock, V.: Konfigurationsraum und zweite Quantelung. Z. Phys. **75**(9–10), 622–647 (1932). (**in German**)
12. Gregorič, M., Mankoč Borštnik, N.S.: Quantum gates and quantum algorithms with Clifford algebra technique. Int. J. Theor. Phys. **48**, 507–515 (2009)
13. Hadfield, H., Hildenbrand, D., Arsenovic, A.: Gajit: symbolic optimisation and JIT compilation of geometric algebra in Python with GAALOP and Numba. Lect. Notes Comput. Sci. **11542**, 499–510 (2019)
14. Hadfield, H., Wei, L., Lasenby, J.: The forward and inverse kinematics of a delta robot. Lect. Notes Comput. Sci. **12221**, 447–458 (2020)
15. Havel, T.F., Doran, Ch.J.: Geometric algebra in quantum information processing. In: Quantum Computation and Information, 81–100, Contemp. Math., 305. American Mathematical Society (2000)
16. Hildenbrand, D., Franchini, S., Gentile, A., Vassallo, G., Vitabile, S.: GAPPCO: an easy to configure geometric algebra coprocessor based on GAPP programs. Adv. Appl. Clifford Algebras **27**(3), 2115–2132 (2017)
17. Hildenbrand, D., Hrdina, J., Návrat, A., Vašík, P.: Local controllability of snake robots based on CRA, theory and practice. Adv. Appl. Clifford Algebras **30**(1), 1–21 (2020)
18. Hrdina, J., Návrat, A., Vašík, P.: Conic fitting in geometric algebra setting. Adv. Appl. Clifford Algebras **29**(4), 1–13 (2019)
19. Lasenby, A., et al.: 2-Spinors, twistors and supersymmetry in the spacetime algebra. In: Oziewicz, Z., et al. (eds.) Spinors, Twistors, Clifford Algebras and Quantum Deformations. Kluwer Academic, Dordrecht (1993)
20. Lounesto, P.: Clifford Algebra and Spinors, 2nd edn. Cambridge University Press, Cambridge (2006)
21. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information, 10th Anniversary Cambridge University Press, Cambridge (2010)

22. Somaroo, S.S., Cory, D.G., Havel, T.F.: Expressing the operations of quantum computing in multiparticle geometric algebra. Phys. Lett. A **240**(1–2), 1–7 (1998)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.