Journal of Mathematical Imaging and Vision 22: 27–48, 2005

Pose Estimation in Conformal Geometric Algebra Part I: The Stratification of Mathematical Spaces

BODO ROSENHAHN AND GERALD SOMMER
Cognitive Systems Group, Institute of Computer Science and Applied Mathematics, Christian-Albrechts-University of Kiel, D-24105 Kiel, Germany
bro@ks.informatik.uni-kiel.de
gs@ks.informatik.uni-kiel.de

Abstract. 2D-3D pose estimation means to estimate the relative position and orientation of a 3D object with respect to a reference camera system. This work has its main focus on the theoretical foundations of the 2D-3D pose estimation problem: We discuss the involved mathematical spaces and their interaction within higher order entities. To cope with the pose problem (how to compare 2D projective image features with 3D Euclidean object features), the principle we propose is to reconstruct image features (e.g. points or lines) to one dimensional higher entities (e.g. 3D projection rays or 3D reconstructed planes) and express constraints in the 3D space. It turns out that the stratification hierarchy [11] introduced by Faugeras is involved in the scenario. But since the stratification hierarchy is based on pure point concepts a new algebraic embedding is required when dealing with higher order entities. The conformal geometric algebra (CGA) [24] is well suited to solve this problem, since it subsumes the involved mathematical spaces. Operators are defined to switch entities between the algebras of the conformal space and its Euclidean and projective subspaces. This leads to another interpretation of the stratification hierarchy, which is not restricted to be based solely on point concepts. This work summarizes the theoretical foundations needed to deal with the pose problem. Therefore it contains mainly basics of Euclidean, projective and conformal geometry. Since especially conformal geometry is not well known in computer science, we recapitulate the mathematical concepts in some detail. We believe that this geometric model is useful also for many other computer vision tasks and has been ignored so far. Applications of these foundations are presented in Part II [36].

Keywords: 2D-3D pose estimation, stratification hierarchy, conformal geometric algebra

1. Introduction

In this work we are concerned with the theoretical foundations of an algorithmic approach for simultaneous 2D-3D pose estimation from correspondences of different entities. Pose estimation itself is a basic visual task [14], and several approaches for monocular pose estimation exist, which relate the position of a 3D object to a reference camera coordinate system [1, 22, 39, 43]. Nearly all papers concentrate on one specific type of correspondences. But many situations are conceivable in which a system has to gather information from different hints or has to consider different reliabilities of measurements. While from the first situation the necessity follows to relate the correspondences of quite different geometric entities, the second problem necessitates the use of weighted mixtures of correspondences. To cope algebraically with these combined informations, is in general very hard. For example, some algorithms assume point correspondences between 3D model and 2D image data and relate 3D points to 3D projection lines [39]. Other algorithms assume line correspondences and relate 3D lines to 3D (reconstructed) planes [21, 22]. Several algorithms use information of the image plane to relate points to entities like circles [23]. All these papers use different algebraic embeddings. Matrix, quaternion and dual-quaternion algebras can be found to describe the situations in different geometries (Euclidean, affine or projective) [11, 37, 38].

One work concerning the combination of different kinds of correspondences can be found in [20]. There only point and line correspondences are treated.

In [37, 41] we started to embed the pose estimation problem for point, line and plane correspondences in the kinematic framework. We continued in [32] by applying a conformal [24] embedding, which appears much more compact and natural. This enables us to formalize the monocular pose estimation problem for kinematic chains [31] and to extend it to circle and sphere concepts [35].

Our work is separated in two parts, Part I (this article) and Part II [36]. Part I deals with the foundations of the pose estimation problem and formalizes the pose scenario by using the language of geometric algebras. It turns out, that the conformal geometric algebra (CGA) provides a new model dealing with projective and kinematic geometry which is not based on point concepts leading to a new stratification hierarchy. In Part II we then continue with application of these foundations to the pose estimation problem of different corresponding entities.

The main attribute of this contribution is to give an overview of the geometric scenario for 2D-3D pose estimation and their algebraic embedding in conformal geometric algebra (CGA) [24]. The contribution is organized as follows: The second section describes the pose estimation scenario in the context of the stratification hierarchy. Then, geometric algebras are introduced. Therefore, we start with the algebra of the Euclidean space, continue with the algebra of the projective space and end up in the algebra of the conformal space which subsumes the former ones. In the fourth section, the relations of projective and conformal geometry will be developed. This will be used in the Part II to formalize the 2D-3D pose estimation problem in one algebraic context.

2. Foundations of the 2D-3D Pose Estimation Problem

This section introduces the foundations of the 2D-3D pose estimation problem. Therefore, the general scenario is explained firstly. Then the involved mathematical spaces are explained and thirdly, the main principles how to cope with the pose estimation problem are explained and discussed.

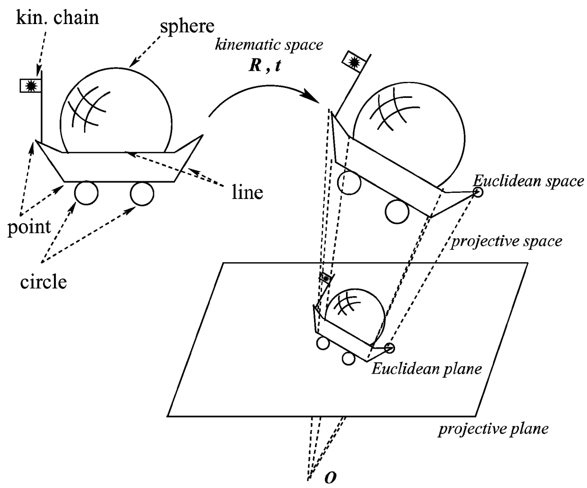


Figure 1. The scenario. The solid lines describe the assumptions: the camera model, the model of the object (consisting of points, lines, circles, spheres and kinematic chains) and corresponding extracted entities on the image plane. The dashed lines describe the pose of the model, which leads to the best fit of the object with the actually extracted entities.

2.1. *The Scenario of Pose Estimation*

In the scenario of Fig. 1 we describe the following situation: We assume 3D points, 3D lines, 3D spheres, 3D circles or kinematic chain segments as features or components of an object or reference model. Furthermore, we extract corresponding features in an image of a calibrated camera. The aim is to find the rotation $\mathcal{R}$ and translation $\mathcal{T}$ of the object, which lead to the best fit of the reference model with the actual extracted entities. One main question is, how to define a geometric error measure with respect to that. Though it is clear by intuition, a mathematical formalization is not easy and not unique. Comparing model features to image features leads to sets of constraint equations which have to be solved and model the involved geometry in an implicit manner.

The method how to establish the correspondences is out of the scope of this paper. The reader should consult e.g. [4] as an example to solve the matching problem in this context.

2.2. *The Stratification Hierarchy and Pose Estimation*

In the scenario of Fig. 1 four mathematical frameworks can be identified: The first one is the projective plane

$\mathcal{P}^2$ of a camera, embedded in the second framework, a 3D projective space $\mathcal{P}^3$. In this 3D projective space it is possible to project or reconstruct entities. The third one is the framework of kinematics. It contains the map of the *direct affine isometries* [12], which can be used to describe rigid body motions. A set of entities with the property that the distances between any two of them never vary is called a *rigid body*, and a transformation with the property of preserving distances during a continuous transformation is called a *rigid body motion*. A rigid body motion corresponds to the special Euclidean transformation group $SE(3)$. Although being a transformation by itself, it subsumes rotation and translation. To distinguish between two rigid body motions, a distance measure on the manifold has to be defined [7, 42]. But this is no simple task in general. Instead, the distance of two geometric entities in Euclidean space can be used to derive a measure of motions. This necessitates as a fourth framework the Euclidean space or Euclidean plane. The basic definitions of these spaces are the following [12]: The Euclidean space is a vector space V with a symmetric positive definite bilinear form (which induces a Euclidean norm). The kinematic space is an affine space with the group of rigid motions as special affine transformation. The projective space is the set of $(V\backslash\{0\})/\sim$ of equivalence classes with

$$\forall u, v \in V\backslash\{0\} : u \sim v \Leftrightarrow \exists\lambda \in \mathbb{R} : v = \lambda u.$$

Mathematically, a projective space $\mathcal{P}(V)$ is a set of equivalence classes of vectors in V. The spirit of projective geometry is to view an equivalence class $(u)_\sim$ as an *atomic* object, forgetting the internal structure of the equivalence class. For this reason, it is customary to call an equivalence class $a = (u)_\sim$ a *point* (the entire equivalence class $(u)_\sim$ is collapsed into a single object, viewed as a point).

The idea is to end up later in the Euclidean space. In that way it is possible to cope geometrically with the problem of noisy data and to evaluate the quality of the estimated pose. But since the Euclidean space is not well suited to describe projective geometry and kinematics, the aim is to transform the generated constraint equations only in the very last step in a distance measure of the Euclidean space. Before this step, we want to use the other spaces to represent partial problems in a suitable way. The above mentioned spaces of the pose estimation scenario are exactly the spaces of the stratification hierarchy which Faugeras introduced in 1995 [11]. The three main representations he is considering are the projective, affine and metric ones. All strata are involved in the 2D-3D pose estimation problem.

Table 1. Stratification of mathematical spaces.

Concept	Stratification				
Vector calculus	Euclidean	$\subseteq$	affine	$\subseteq$	projective
Geometric algebra	Euclidean	$\subseteq$	projective	$\subseteq$	conformal

In our approach, we are using geometric algebras instead of vector calculus to represent and handle different mathematical spaces of geometric meaning. The maximum sized algebra over a Euclidean space so far used by us is an algebra to handle conformal transformations [15]. A transformation is said to be conformal if it (locally) preserves angles. The conformal geometric algebra (CGA) contains the algebras for projective and Euclidean geometry as subalgebras, thus leading to another formalization of the stratification hierarchy, we propose in this contribution. Table 1 shows the different stratification hierarchies. The stratification hierarchy proposed by Faugeras has its roots in the vector space concepts and assumes points as the represented basic geometric entities. All other geometric entities are derived as subspaces of point sets without having an own algebraic existence. Well known is the homogeneous extension to express a Euclidean space as affine space and to use the homogeneous component for distinction between points and directions in the affine space. The projective space as a set of equivalence classes is directly built on the homogeneous vector space concepts. So this way to stratify the vision space is clearly motivated by the underlying point concepts of the vector spaces.

In geometric algebras instead, we do have besides point concepts so-called multivector concepts to model geometry. In the next section we will explain why it is necessary also to extend geometric algebras to homogeneous models. But this leads to a different stratification of the spaces since this stratification is not based on pure point concepts any more. Instead, the new stratification concept contains algebras for the Euclidean, projective and conformal space.

2.3. *Principles of Solving the Pose Estimation Problem*

The main problem of 2D-3D pose estimation is how to compare 3D Euclidean object features with 2D projective image data. There are two strategies for comparison: On the one hand it is possible to project the transformed entity in the image plane and to compare it with the extracted image data. This leads to a comparison in the projective plane or Euclidean plane, respectively. The second possibility is to projectively reconstruct the object features from the image data and to compare the (by one dimension higher) entities with the 3D object features. Both approaches have advantages and disadvantages. Here we want to discuss a few properties of both strategies: To enable comparisons in the first strategy, the projected object features have to be scaled in their homogeneous component. This leads to fractions with the unknown transformation in both, the numerator and the denominator. The equations are not linear any more and are not easy to solve numerically. Though the equations can also be expressed as projective linear system of equations, the problem is then to lose a distance measure and to risk bad conditioned equations. To avoid such problems, orthographic projections (see e.g. [6]) are used, but then the camera model is not perspective any more. Since the second strategy uses projective reconstructed data, this problem does not occur there. But the problem is that the distance measures in the 3D space is different to those in the image plane: Though the distance of two image points may be constant, the distance of two 3D projectively reconstructed points varies with the distance of the points to the optical center of the camera. This necessitates for degenerate situations[1] that the (from the image and object features generated) constraint equations must be adapted with respect to the projective depth. Table 2 summarizes the main principles of solving the pose problem in an implicit manner.

In our approach (similar to [43]) we projectively reconstruct the 3D data from image data and compare the one dimensional higher entities (their projective equivalence classes) with the 3D object features. There are three main arguments why we decided for the second strategy which is based on the stratification concepts above: Firstly, we want to describe the constraints as simply as possible and want to gain real-time performance. For this, the projectively reconstructed data are easier to handle in the 3D kinematic space than the projected data in the 2D projective space. The second advantage of the approach is that the error measures are formalized in the 3D Euclidean space and are directly connected to a spatial distance measure. This is in contrast to other approaches, where the minimization of estimating errors of the rigid body motion has to be computed directly on the manifold of the geometric transformation [7, 42]. The third argument is that the depth dependence of the 3D constraints can be adapted in each situation. As will be later shown (see Part II) our constraints can be scaled, and therefore transformed in depth-depending constraints comparable to the situation observed in the 2D image plane.

Table 2. Principles of formalizing constraints for the pose problem.

Constraint	Linear	Geometric distance measure	Full perspective
2D Euclidean	no	yes	yes
Orthographic projective	yes	yes	no
Full projective	yes	no	yes
3D kinematic	yes	yes	yes

Since CGA can be used to formalize kinematics and since it contains the algebras for projective and Euclidean geometry as subalgebras, it is well suited to be used in this context. Therefore, the whole scenario is formalized in CGA: That are the entities, the kinematic chains, the transformations of the entities and the constraints for collinearity, coplanarity and tangentiality of the involved entities.

3. Introduction to Geometric Algebras

What we currently call *geometric algebra* [15] is tightly related to Clifford algebra. Both in fact represent families of algebras which depend on both the chosen vector spaces the algebras are derived from and the chosen kind of product defining the special algebra. A nice historic introduction of Clifford's contribution of inventing a geometric extension of the real number system to such which provides a complete algebraic representation of directed numbers can be found in [44].

Clifford (or geometric) algebras have the properties of dense symbolic representations of higher order entities and of linear operations acting on those, coupled with strong under-pinned mathematical concepts. It is nice that many geometric concepts, which are often introduced separately in special algebras are unified in geometric algebras. So the concepts of duality in projective geometry, Lie algebras and Lie groups, incidence algebra, Plücker representations of lines, complex numbers, quaternions and dual quaternions can all be found in suitable geometric algebras. In geometric algebras there are strong relations between algebraic and geometric entities. Furthermore, both the object concepts and the operations acting on those are represented in one unique mathematical language.

We will now continue with a general introduction to geometric algebras and will proceed with algebras to model the Euclidean, projective and conformal space. A more extended introduction into geometric algebras can be found in [9, 10, 15, 16, 18, 19, 40]. See also the courses on web, e.g. [8, 29].

In general, a geometric algebra $\mathcal{G}_{p,q,r}$ is a linear space of dimension $2^n, n = p+q+r$, with a subspace structure, called blades, to represent so-called multivectors as higher grade algebraic entities in comparison to vectors of a vector space as first grade entities, or scalars as grade zero entities. A geometric algebra $\mathcal{G}_{p,q,r}$ results in a constructive way from a vector space $\mathbb{R}^{p,q,r}$, endowed with the signature (p, q, r), by application of a geometric product. The geometric product of two multivectors $\boldsymbol{A}$ and $\boldsymbol{B}$ is denoted as $\boldsymbol{AB}$. The geometric product consists of an outer $(\wedge)$ and an inner $(\cdot)$ product, whose roles are to increase or to decrease the order of the algebraic entities, respectively.

To be more detailed, we define the geometric product of a geometric algebra $\mathcal{G}_{p,q,r}$ for two basis vectors $\mathbf{e}_i$ and $\mathbf{e}_j$ as

$$\mathbf{e}_i\mathbf{e}_j = \begin{cases} 1 & \text{for } i = j \in \{1, \ldots, p\} \\ -1 & \text{for } i = j \in \{p+1, \ldots, p+q\} \\ 0 & \text{for } i = j \in \{p+q+1, \ldots, n\} \\ \mathbf{e}_{ij} = \mathbf{e}_i \wedge \mathbf{e}_j = -\mathbf{e}_j \wedge \mathbf{e}_i & \\ \quad \text{for } i \neq j & \end{cases} \tag{3.1}$$

A vector space with signature (p, q, r), $q \neq 0$, $r \neq 0$, is called pseudo-Euclidean. If $r \neq 0$, then its metric is degenerate. Although the dual-quaternions, which have some importance in kinematics, are isomorphic to a degenerate geometric algebra, see [2, 3], we will in the following only consider non-degenerate geometric algebras $\mathcal{G}_{p,q}$ where $r = 0$. Besides, we will write $\mathcal{G}_n$ if $q = 0$, that is, there is a Euclidean metric.

The inner $(\cdot)$ and outer $(\wedge)$ products of two vectors $\boldsymbol{u}, \boldsymbol{v} \in \langle\mathcal{G}_{p,q}\rangle_1 \equiv \mathbb{R}^{p+q}$ are defined as

$$\boldsymbol{u} \cdot \boldsymbol{v} := \frac{1}{2}(\boldsymbol{uv} + \boldsymbol{vu}), \tag{3.2}$$

$$\boldsymbol{u} \wedge \boldsymbol{v} := \frac{1}{2}(\boldsymbol{uv} - \boldsymbol{vu}). \tag{3.3}$$

Here $\alpha = \boldsymbol{u} \cdot \boldsymbol{v}$ represents a scalar, which is of grade zero, i.e. $\alpha \in \langle\mathcal{G}_{p,q}\rangle_0$ with $\langle\cdot\rangle_s$ is the operator to separate the grade-s entities of the linear space $\mathcal{G}_{p,q}$. Besides $\boldsymbol{B} = \boldsymbol{u} \wedge \boldsymbol{v}$ represents a bivector, i.e. $\boldsymbol{B} \in \langle\mathcal{G}_{p,q}\rangle_2$.

As extension, the inner product of an r-blade $\boldsymbol{u}_1 \wedge \ldots \wedge \boldsymbol{u}_r$ with an s-blade $\boldsymbol{v}_1 \wedge \ldots \wedge \boldsymbol{v}_s$ can be defined recursively by

$$\begin{aligned}&(\boldsymbol{u}_1 \wedge \ldots \wedge \boldsymbol{u}_r) \cdot (\boldsymbol{v}_1 \wedge \ldots \wedge \boldsymbol{v}_s) \\ &= \begin{cases} ((\boldsymbol{u}_1 \wedge \ldots \wedge \boldsymbol{u}_r) \cdot \boldsymbol{v}_1) \cdot (\boldsymbol{v}_2 \wedge \ldots \wedge \boldsymbol{v}_s) & \text{if } r \geq s \\ (\boldsymbol{u}_1 \wedge \ldots \wedge \boldsymbol{u}_{r-1}) \cdot (\boldsymbol{u}_r \cdot (\boldsymbol{v}_1 \wedge \ldots \wedge \boldsymbol{v}_s)) & \text{if } r < s, \end{cases}\end{aligned} \tag{3.4}$$

with

$$\begin{aligned}&(\boldsymbol{u}_1 \wedge \ldots \wedge \boldsymbol{u}_r) \cdot \boldsymbol{v}_1 \\ &\quad = \sum_{i=1}^{r}(-1)^{r-i}\boldsymbol{u}_1 \wedge \ldots \wedge \boldsymbol{u}_{i-1} \wedge (\boldsymbol{u}_i \cdot \boldsymbol{v}_1) \\ &\quad \wedge \boldsymbol{u}_{i+1} \wedge \ldots \wedge \boldsymbol{u}_r,\end{aligned} \tag{3.5}$$

$$\begin{aligned}&\boldsymbol{u}_r \cdot (\boldsymbol{v}_1 \wedge \ldots \wedge \boldsymbol{v}_s) \\ &\quad = \sum_{i=1}^{s}(-1)^{i-1}\boldsymbol{v}_1 \wedge \ldots \wedge \boldsymbol{v}_{i-1} \wedge (\boldsymbol{u}_r \cdot \boldsymbol{v}_i) \\ &\quad \wedge \boldsymbol{v}_{i+1} \wedge \ldots \wedge \boldsymbol{v}_s.\end{aligned} \tag{3.6}$$

We will make this more explicit in the next subsections.

For two blades $\boldsymbol{A}_{\langle r\rangle}$ and $\boldsymbol{B}_{\langle s\rangle}$ with non zero grade r and $s \in \mathbb{N}$ the inner and outer product can also be expressed as

$$\boldsymbol{A}_{\langle r\rangle} \cdot \boldsymbol{B}_{\langle s\rangle} = \langle\boldsymbol{AB}\rangle_{|r-s|} \tag{3.7}$$

and

$$\boldsymbol{A}_{\langle r\rangle} \wedge \boldsymbol{B}_{\langle s\rangle} = \langle\boldsymbol{AB}\rangle_{r+s}, \tag{3.8}$$

with the following additional rules:

1. If $r = 0$ or $s = 0$, the inner product is zero.
2. If $r + s > n$, the outer product is zero.

The blades of highest grade are n-blades, called *pseudoscalars* $\boldsymbol{P}$. Pseudoscalars differ from each other by a nonzero scalar only. There exist two unit n-blades, called the *unit pseudoscalars* $\pm\boldsymbol{I}$. The unit pseudoscalars are often indexed by the generating vector spaces of the geometric algebras, for example $\boldsymbol{I}_E$, $\boldsymbol{I}_P$ and $\boldsymbol{I}_C$ represent the unit pseudoscalars of the algebras for the Euclidean, projective and conformal space, respectively.

The magnitude $[\boldsymbol{P}]$ of a pseudo-scalar $\boldsymbol{P}$ is a scalar. It will be called *bracket* of $\boldsymbol{P}$ and is defined by

$$[\boldsymbol{P}] := \boldsymbol{P}\boldsymbol{I}^{-1}. \tag{3.9}$$

For the bracket determined by n vectors, we write

$$\begin{aligned}[\boldsymbol{v}_1 \ldots \boldsymbol{v}_n] &= [\boldsymbol{v}_1 \wedge \ldots \wedge \boldsymbol{v}_n] \\ &= (\boldsymbol{v}_1 \wedge \ldots \wedge \boldsymbol{v}_n)\boldsymbol{I}^{-1}.\end{aligned} \tag{3.10}$$

This can also be taken as a definition of a determinant, well known from matrix calculus. We define the *dual* $\boldsymbol{X}^\star$ of an r-blade $\boldsymbol{X}$ by

$$\boldsymbol{X}^\star := \boldsymbol{X}\boldsymbol{I}^{-1}. \tag{3.11}$$

It follows, that the dual of an r-blade is an $(n-r)$-blade.

The *reverse* $\widetilde{\boldsymbol{A}_{\langle s\rangle}}$ of an s-blade $\boldsymbol{A}_{\langle s\rangle} = \boldsymbol{a}_1 \wedge \ldots \wedge \boldsymbol{a}_s$ is defined as the reverse outer product of the vectors $\boldsymbol{a}_i$,

$$\begin{aligned}\widetilde{\boldsymbol{A}_{\langle s\rangle}} &= (\boldsymbol{a}_1 \wedge \boldsymbol{a}_2 \wedge \ldots \wedge \boldsymbol{a}_{s-1} \wedge \boldsymbol{a}_s)^\sim \\ &:= \boldsymbol{a}_s \wedge \boldsymbol{a}_{s-1} \wedge \ldots \wedge \boldsymbol{a}_2 \wedge \boldsymbol{a}_1.\end{aligned} \tag{3.12}$$

The *join* $\boldsymbol{A} \dot{\wedge} \boldsymbol{B}$ is the pseudoscalar of the space given by the sum of spaces spanned by $\boldsymbol{A}$ and $\boldsymbol{B}$.

For blades $\boldsymbol{A}$ and $\boldsymbol{B}$ the *dual shuffle* product $\boldsymbol{A} \vee \boldsymbol{B}$ is defined by the DeMorgan rule

$$(\boldsymbol{A} \vee \boldsymbol{B})^\star := \boldsymbol{A}^\star \dot{\wedge} \boldsymbol{B}^\star. \tag{3.13}$$

For blades $\boldsymbol{A}$ and $\boldsymbol{B}$ it is possible to use the join to express *meet* operations: Let be $\boldsymbol{A}$ and $\boldsymbol{B}$ two arbitrary blades and let $\boldsymbol{J} = \boldsymbol{A} \dot{\wedge} \boldsymbol{B}$, then

$$(\boldsymbol{A} \vee \boldsymbol{B}) := \left(\boldsymbol{A}\boldsymbol{J}^{-1} \wedge \boldsymbol{B}\boldsymbol{J}^{-1}\right)\boldsymbol{J}. \tag{3.14}$$

The meet $\vee$, also called the shuffle product, is the common factor of $\boldsymbol{A}$ and $\boldsymbol{B}$ with the highest grade. The meet will be used in Section 3.2 for incidence estimation of points, lines and planes.

For further computations, we also use both the commutator $\underline{\times}$ and the anticommutator $\overline{\times}$ product for any two multivectors,

$$\begin{aligned}\boldsymbol{A}\boldsymbol{B} &= \frac{1}{2}(\boldsymbol{A}\boldsymbol{B} + \boldsymbol{B}\boldsymbol{A}) + \frac{1}{2}(\boldsymbol{A}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{A}) \\ &=: \boldsymbol{A} \overline{\times} \boldsymbol{B} + \boldsymbol{A} \underline{\times} \boldsymbol{B}.\end{aligned} \tag{3.15}$$

The reader should consult [27] to become more familiar with the commutator and anticommutator product. Their role is to separate the symmetric part of the geometric product from the antisymmetric one.

Now we will proceed to introduce the algebras for the Euclidean, projective and conformal spaces.

3.1. The Euclidean Geometric Algebra

The algebra $\mathcal{G}_3$, which is derived from $\mathbb{R}^3$, i.e. $n = p = 3$, is the smallest and simplest one, we want to introduce here. This algebra is suitable to represent entities and operations in the 3D Euclidean space. Therefore, we call it EGA as abbreviation for Euclidean geometric algebra. We start with the three orthonormal basis vectors $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ of the 3D Euclidean space. The geometric algebra of the 3D Euclidean space consists of $2^3 = 8$ basis vectors,

$$\mathcal{G}_3 = span\{1, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_{23}, \mathbf{e}_{31}, \mathbf{e}_{12}, \mathbf{e}_{123} = \boldsymbol{I}_E\}. \tag{3.16}$$

The elements $\mathbf{e}_{ij} = \mathbf{e}_i\mathbf{e}_j = \mathbf{e}_i \wedge \mathbf{e}_j$ are the unit bivectors and the element $\mathbf{e}_{123} = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3 = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3 = \boldsymbol{I}_E$ is a trivector, called Euclidean unit pseudo-scalar, which squares to -1 and commutes with scalars, vectors and bivectors. To make more clear the above introduced rules of the geometric product, we will formulate the geometric product of two vectors as an example:

$$\begin{aligned}\boldsymbol{u}\boldsymbol{v} &= (u_1\mathbf{e}_1 + u_2\mathbf{e}_2 + u_3\mathbf{e}_3)(v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + v_3\mathbf{e}_3) \\ &= u_1\mathbf{e}_1(v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + v_3\mathbf{e}_3) \\ &\quad + u_2\mathbf{e}_2(v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + v_3\mathbf{e}_3) \\ &\quad + u_3\mathbf{e}_3(v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + v_3\mathbf{e}_3) \\ &= u_1v_1 + u_2v_2 + u_3v_3 + (u_1v_2 - u_2v_1)\mathbf{e}_{12} \\ &\quad + (u_3v_1 - u_1v_3)\mathbf{e}_{31} + (u_2v_3 - u_3v_2)\mathbf{e}_{23} \\ &= \boldsymbol{u} \cdot \boldsymbol{v} + \boldsymbol{u} \wedge \boldsymbol{v}.\end{aligned} \tag{3.17}$$

Thus, the geometric product of two vectors leads to a scalar, representing the inner product of the two vectors (corresponding to the scalar product of these vectors in vector calculus), and a bivector, representing the outer product of two vectors. The bivector corresponds to the dual of the vector which results from the cross product ($\times$) of two vectors (in vector calculus). The inner product of a bivector ($\boldsymbol{a} \wedge \boldsymbol{b}$) with a vector $\boldsymbol{c}$ leads to another vector,

$$\begin{aligned}(\boldsymbol{a} \wedge \boldsymbol{b}) \cdot \boldsymbol{c} &\overset{3.5}{=} -(\boldsymbol{a} \cdot \boldsymbol{c}) \wedge \boldsymbol{b} + \boldsymbol{a} \wedge (\boldsymbol{b} \cdot \boldsymbol{c}) \\ &= -(\boldsymbol{a} \cdot \boldsymbol{c})\boldsymbol{b} + (\boldsymbol{b} \cdot \boldsymbol{c})\boldsymbol{a},\end{aligned} \tag{3.18}$$

and thus, we get the equivalent formulation of the famous cross product rule for the 3D case,[2]

$$(\boldsymbol{a} \times \boldsymbol{b}) \times \boldsymbol{c} = \langle \boldsymbol{a}, \boldsymbol{c} \rangle \boldsymbol{b} - \langle \boldsymbol{b}, \boldsymbol{c} \rangle \boldsymbol{a}. \tag{3.19}$$

The inner product of two bivectors leads to a scalar,

$$\begin{aligned}(\boldsymbol{a} \wedge \boldsymbol{b}) \cdot (\boldsymbol{c} \wedge \boldsymbol{d}) &\overset{3.4}{=} ((\boldsymbol{a} \wedge \boldsymbol{b}) \cdot \boldsymbol{c}) \cdot \boldsymbol{d} \\ &\overset{3.18}{=} -((\boldsymbol{a} \cdot \boldsymbol{c})\boldsymbol{b} + (\boldsymbol{b} \cdot \boldsymbol{c})\boldsymbol{a}) \cdot \boldsymbol{d} \\ &= -(\boldsymbol{a} \cdot \boldsymbol{c})(\boldsymbol{b} \cdot \boldsymbol{d}) + (\boldsymbol{b} \cdot \boldsymbol{c})(\boldsymbol{a} \cdot \boldsymbol{d}),\end{aligned} \tag{3.20}$$

and we get (for the 3D case) the *Lagrange identity* for the cross products of 3D vectors,

$$\langle (\boldsymbol{a} \times \boldsymbol{b}), (\boldsymbol{c} \times \boldsymbol{d}) \rangle = \langle \boldsymbol{a}, \boldsymbol{c} \rangle \langle \boldsymbol{b}, \boldsymbol{d} \rangle - \langle \boldsymbol{b}, \boldsymbol{c} \rangle \langle \boldsymbol{a}, \boldsymbol{d} \rangle. \tag{3.21}$$

Note that the outer product is more general than the cross product, since it can be applied to spaces of any dimension and of any signature.

3.1.1. Representation of Points, Lines and Planes in the Euclidean Geometric Algebra. Points, lines and planes of the 3D space can all be modeled in the algebra $\mathcal{G}_3$. A point, representing a position in the 3D space, can simply be expressed by a linear combination of the three basis vectors,

$$\boldsymbol{u} = u_1\mathbf{e}_1 + u_2\mathbf{e}_2 + u_3\mathbf{e}_3. \tag{3.22}$$

A line can be represented as an inhomogeneous multivector by using a vector $\boldsymbol{r}$ for the direction and a bivector $\boldsymbol{m}$ containing the moment, as outer product of a point $\boldsymbol{x}$ on the line and the direction $\boldsymbol{r}$ of the line [5],

$$\begin{aligned}\boldsymbol{l} &= \boldsymbol{r} + \boldsymbol{x} \wedge \boldsymbol{r} \\ &= \boldsymbol{r} + \boldsymbol{m}.\end{aligned} \tag{3.23}$$

Incidence of a point with a line can be expressed by the kernel of a function $\mathcal{F}_{XL}$,

$$\begin{aligned}\boldsymbol{p} \in \boldsymbol{l} &\Leftrightarrow \mathcal{F}_{XL}(\boldsymbol{p}, \boldsymbol{l}) = 0 \\ &\Leftrightarrow (\boldsymbol{p} \wedge \boldsymbol{r}) - \boldsymbol{m} = 0.\end{aligned} \tag{3.24}$$

A plane can be represented by an entity one grade higher then the line. In terms of the Hesse distance d from the origin to the plane (coded by the Euclidean pseudo-scalar) and the unit bivector direction $\boldsymbol{n}$ from the origin to the plane, a plane is defined by

$$\boldsymbol{p} = \boldsymbol{n} + \boldsymbol{I}_E d. \tag{3.25}$$

Thus, a plane is an inhomogeneous multivector, consisting of a bivector and a trivector. The incidence of a point with a plane can be expressed in the following way,

$$\begin{aligned}\boldsymbol{x} \in \boldsymbol{p} &\Leftrightarrow \mathcal{F}_{XP}(\boldsymbol{x}, \boldsymbol{p}) = 0 \\ &\Leftrightarrow (\boldsymbol{x} \wedge \boldsymbol{n}) - \boldsymbol{I}_E d = 0.\end{aligned} \tag{3.26}$$

If we compare the representations of these three entities in EGA, we recognize, that those of lines and planes are more complicated than that of points. Also the constraint equations expressing the incidence relation are not compact or simple. This has its reason in the fact, that so far no origin of the vector space is modeled within the geometric algebra. In vector calculus this can formally be done by introducing an additional (or *homogeneous*) coordinate. Such an extension will also be done in Section 3.2 for modeling the projective space in a Clifford algebra.

3.1.2. Rotations and Translations in the Euclidean Space. Multiplication of the three basis vectors $\mathbf{e}_i$ with $\boldsymbol{I}_E$ results in the three basis bivectors $\boldsymbol{I}_E\mathbf{e}_i$. These bivectors rotate vectors in their own plane by 90°, e.g. $(\boldsymbol{I}_E\mathbf{e}_3)\mathbf{e}_2 = \mathbf{e}_{123}\mathbf{e}_3\mathbf{e}_2 = \mathbf{e}_{12}\mathbf{e}_2 = \mathbf{e}_1$, or $(\boldsymbol{I}_E\mathbf{e}_1)\mathbf{e}_2 = \mathbf{e}_{123}\mathbf{e}_1\mathbf{e}_2 = \mathbf{e}_{23}\mathbf{e}_2 = -\mathbf{e}_1$, etc. Note, since the basis vectors are orthonormal, it is equivalent to write $\mathbf{e}_{ij} = \mathbf{e}_i \wedge \mathbf{e}_j$ for $i \neq j$. The basis bivectors square to -1, and so they can easily be identified with the unit vectors $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ of the quaternion algebra $\mathbb{H}$ with the famous Hamilton relations $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$. We have the isomorphy $\mathcal{G}_3^+ \simeq \mathbb{H}$ with $\mathcal{G}_3^+$ as the even-grade subalgebra of $\mathcal{G}_3$.

The bivectors of the geometric algebra can be used to represent rotations of points in the 3D space. A *rotor* $\boldsymbol{R}$ is an even grade element of the algebra $\mathcal{G}_3$ which satisfies $\boldsymbol{R}\widetilde{\boldsymbol{R}} = 1$. Since the even grade elements of $\mathcal{G}_3$ are scalars and bivectors, a rotor $\boldsymbol{R}$ and its reverse $\widetilde{\boldsymbol{R}}$ is given by

$$\boldsymbol{R} = \underbrace{u_0}_{\text{scalar}} + \underbrace{u_1\mathbf{e}_{23} + u_2\mathbf{e}_{31} + u_3\mathbf{e}_{12}}_{\text{bivectors}}, \tag{3.27}$$

$$\widetilde{\boldsymbol{R}} = \underbrace{u_0}_{\text{scalar}} - \underbrace{u_1\mathbf{e}_{23} - u_2\mathbf{e}_{31} - u_3\mathbf{e}_{12}}_{\text{bivectors}}. \tag{3.28}$$

If we use the Euler representation of a rotor,

$$\begin{aligned} \boldsymbol{R} &= \exp\left(-\frac{\theta}{2}\boldsymbol{n}\right) \\ &= \cos\left(\frac{\theta}{2}\right) - \boldsymbol{n}\sin\left(\frac{\theta}{2}\right), \end{aligned} \tag{3.29}$$

it takes on geometric significance. Here $\boldsymbol{n}$ is a unit bivector representing the plane of the rotation (its dual $\boldsymbol{n}^\star$ corresponds to the rotation axis) and $\theta \in \mathbb{R}$ is representing the amount of rotation. The rotation of a point, represented by its vector $\boldsymbol{x}$, can be carried out by multiplying the rotor $\boldsymbol{R}$ from the left and its reverse from the right to the point $\boldsymbol{x}$,

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x}\widetilde{\boldsymbol{R}}. \tag{3.30}$$

Such a multiplication is also called *versor product* and the bivector $\boldsymbol{R}$ is the *versor* of this versor product. A rotor is representing the group $SO(3)$ in EGA. Thus, the operation concatenates according to a left-sided product $\boldsymbol{R} = \boldsymbol{R}_2\boldsymbol{R}_1$ yielding a new rotor. From this follows

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x}\widetilde{\boldsymbol{R}} = (\boldsymbol{R}_2\boldsymbol{R}_1)\,\boldsymbol{x}\left(\widetilde{\boldsymbol{R}_1}\widetilde{\boldsymbol{R}_2}\right). \tag{3.31}$$

In contrast to rotation matrices of $\mathbb{R}^3$, rotors are working not only on points, but for all types of geometric objects, and are defined independent on their grade and the dimension of the space.

The exponential function of multivectors $\boldsymbol{m}$ can also be expressed via its series expression,

$$\exp(\boldsymbol{m}) = \sum_{k=0}^{\infty} \frac{\boldsymbol{m}^k}{k!}. \tag{3.32}$$

In contrast to rotations, there exists no multiplicative way to formalize translation in the Euclidean geometric algebra. The only possibility is to express translations in an additive way, e.g., a point $\boldsymbol{x}$ is translated with a translation vector $\boldsymbol{t}$, by

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}. \tag{3.33}$$

This results from the well-known fact that translations in $\mathbb{R}^3 = \langle \mathcal{G}_3 \rangle_1$ constitute the additive group $\mathbb{R}^3$. Therefore, composite translations follow the rule $\boldsymbol{t} = \boldsymbol{t}_1 + \boldsymbol{t}_2$. Another problem concerns the linearity of both operations. A rotation, $\mathcal{R}$, is a linear operation. Let be $\boldsymbol{x}$ and $\boldsymbol{y}$ any multivectors of $\mathcal{G}_3$, then $\mathcal{R}\{\boldsymbol{x}+\boldsymbol{y}\} = \mathcal{R}\{\boldsymbol{x}\}+\mathcal{R}\{\boldsymbol{y}\}$. But translation behaves not linear. For two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, representing points of $\langle \mathcal{G}_3 \rangle_1$, it follows $\mathcal{T}\{x + y\} \neq \mathcal{T}\{x\} + \mathcal{T}\{y\}$.

These different behaviors cause problems in representing the rigid motion of an object in EGA as linear operation. A rigid motion in Euclidean space is a mapping $\mathcal{D} : \mathbb{R}^3 \to \mathbb{R}^3$ which preserves distances between points and angles between vectors. In general, the movement of a rigid body, that is a rigid displacement, may include both rotation and translation in the following way. Let be $\boldsymbol{x}', \boldsymbol{x} \in \langle \mathcal{G}_3 \rangle_1$, then

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x}\widetilde{\boldsymbol{R}} + \boldsymbol{t}. \tag{3.34}$$

A spatial rigid displacement $\mathcal{D} = (\boldsymbol{R}, \boldsymbol{t})$ belongs to the special Euclidean group $SE(3) = \mathbb{R}^3 \times SO(3)$. Thus a composite displacement $\mathcal{D} = \mathcal{D}_2\mathcal{D}_1$ exists with $\mathcal{D} = (\boldsymbol{R}, \boldsymbol{t}) = (\boldsymbol{R}_2, \boldsymbol{t}_2)(\boldsymbol{R}_1, \boldsymbol{t}_1) = (\boldsymbol{R}_2\boldsymbol{R}_1, \boldsymbol{R}_2\boldsymbol{t}_1 + \boldsymbol{t}_2)$. But regrettably, because of the non-linear behavior of the translation, the displacement is no linear operation in $\mathcal{G}_3$, neither for points nor for other entities. Fortunately, there are other algebraic embeddings which result in linearization with respects to points or other entities. While so far either point or line based transformations for rigid displacements have been distinguished [30], we will introduce in this paper a third category which is based on spheres, see Section 3.3.

3.2. *The Projective Geometric Algebra*

By using homogeneous coordinates we increase the dimension of the vector space by one and the corresponding algebra is of dimension $2^4 = 16$. The elements we gain are now scalars, vectors, bivectors, trivectors and the pseudoscalar. To model 3D projective geometry in a geometric algebra four basis vectors are needed. The signature of the derived vector space will be unimportant, therefore it is free to choose. We will introduce the geometric algebra $\mathcal{G}_{3,1}$ to represent the projective space. Here the additional basis vector $\mathbf{e}_-$ denotes the homogeneous component. Because $\mathbf{e}_-^2 = -1$, this basis vector induces a Minkowski metric. The algebra $\mathcal{G}_{3,1}$ contains the following elements,

$$\begin{aligned} \mathcal{G}_{3,1} = span\{1,\quad & \mathbf{e}_-, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \quad \mathbf{e}_{23}, \mathbf{e}_{31}, \mathbf{e}_{12}, \\ & \mathbf{e}_{-1}, \mathbf{e}_{-2}, \mathbf{e}_{-3}, \quad \mathbf{e}_{123}, \mathbf{e}_{-23}, \mathbf{e}_{-31}, \mathbf{e}_{-12}, \\ & \mathbf{e}_{-123} = \boldsymbol{I}_P\}. \end{aligned} \tag{3.35}$$

Note that e.g. $\mathbf{e}_{-123} \equiv \mathbf{e}_- \wedge \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$ and $\mathbf{e}_{-123}^2 = -1$.

3.2.1. Representation of Points, Lines and Planes in the Projective Geometric Algebra. In contrast to the Euclidean geometric algebra $\mathcal{G}_3$, in the projective geometric algebra $\mathcal{G}_{3,1}$ (PGA) we can simply represent points, lines and planes as r-blades, i.e. homogeneous multivectors of grade r. In that case the previously mentioned duality operator is of special importance since it transforms geometric entities to their duals.

A point can be represented by a 1-blade. The basis vector $\mathbf{e}_-$ represents the homogeneous component of the point. Thus, the point $\boldsymbol{x}$ given in $\mathcal{G}_3$, can be represented in $\mathcal{G}_{3,1}$ by

$$\boldsymbol{X} = \boldsymbol{x} + \mathbf{e}_-. \tag{3.36}$$

Since $\boldsymbol{X} \wedge \boldsymbol{X} = 0$, so also

$$\boldsymbol{X} \wedge \lambda \boldsymbol{X} = 0 \qquad \forall \lambda \in \mathbb{R}\backslash\{0\}. \tag{3.37}$$

For this reason the outer product is used to define the equivalence class of points in the projective space. All vectors $\boldsymbol{X}$ represent a point $\boldsymbol{A}$ if $\boldsymbol{A} \wedge \boldsymbol{X} = 0$. This means, that the so-called *outer product null space* defines the incidence of two entities, similar to [15].

A line can be represented by the outer product of two points, leading to a 2-blade,

$$\begin{aligned} \boldsymbol{L} &= \boldsymbol{X}_1 \wedge \boldsymbol{X}_2 \\ &= (\boldsymbol{x}_1 + \mathbf{e}_-) \wedge (\boldsymbol{x}_2 + \mathbf{e}_-) \\ &= \boldsymbol{x}_1 \wedge \boldsymbol{x}_2 + (\boldsymbol{x}_1 - \boldsymbol{x}_2)\mathbf{e}_- \\ &= \boldsymbol{m} + \boldsymbol{r}\mathbf{e}_-. \end{aligned} \tag{3.38}$$

The line $\boldsymbol{L}$ contains the moment $\boldsymbol{m}$ and direction $\boldsymbol{r}$. Therefore, it corresponds directly to the Plücker representation [5]. Being a 2-blade, the line contains 6 bivector components.

A plane can be represented by the outer product of three points, leading to a 3-blade

$$\begin{aligned} \boldsymbol{P} &= \boldsymbol{X}_1 \wedge \boldsymbol{X}_2 \wedge \boldsymbol{X}_3 \\ &= (\boldsymbol{x}_1 + \mathbf{e}_-) \wedge (\boldsymbol{x}_2 + \mathbf{e}_-) \wedge (\boldsymbol{x}_3 + \mathbf{e}_-) \\ &= \boldsymbol{x}_1 \wedge \boldsymbol{x}_2 \wedge \boldsymbol{x}_3 + (\boldsymbol{x}_1 - \boldsymbol{x}_2) \wedge (\boldsymbol{x}_1 - \boldsymbol{x}_3)\mathbf{e}_- \\ &= d\boldsymbol{I}_E + \boldsymbol{n}\mathbf{e}_-. \end{aligned} \tag{3.39}$$

This representation corresponds to the Hesse description of planes, formalizing a plane by the normal $\boldsymbol{n}$ (as bivector) of the plane, and the Hesse distance d of the plane to the origin.

As can be seen, the generation of the higher order entities is much more natural than in the algebra of the Euclidean space because it results from the incidence algebra of points.

The outer product of two blades is non-vanishing iff their supports have zero intersection. This can be used to prove an incidence relation [19], e.g. a point $\boldsymbol{X}$ is on a line $\boldsymbol{L}$ iff

$$\boldsymbol{X} \wedge \boldsymbol{L} = 0. \tag{3.40}$$

For blades $\boldsymbol{A}$ and $\boldsymbol{B}$ we use the previous defined shuffle product and the join, to express the *meet* operations: Let be $\boldsymbol{A}$ and $\boldsymbol{B}$ two arbitrary blades and let $\boldsymbol{J} = \boldsymbol{A} \dot{\wedge} \boldsymbol{B}$, then the meet can be written as

$$(\boldsymbol{A} \vee \boldsymbol{B}) = \left(\boldsymbol{A}\boldsymbol{J}^{-1} \wedge \boldsymbol{B}\boldsymbol{J}^{-1}\right)\boldsymbol{J}. \tag{3.41}$$

The meet is the common factor of $\boldsymbol{A}$ and $\boldsymbol{B}$ with highest grade. The meet defines a generalized intersection operation.

Note that the incidence operations always lead to entities in the projective space. To re-transform e.g. a projective point to a Euclidean point the *projective split* [15] has to be applied.

The advantage of the algebra $\mathcal{G}_{3,1}$ for the projective space, in comparison to the algebra $\mathcal{G}_3$ for the Euclidean space is that the representation of the entities is much more natural and provided by the subspace concepts. From this results a nice formulation of the duality concept in projective geometry and compact descriptions of joins and meets of subspaces, just by applying a suitable operator.

In PGA projective transformations can be expressed. These transformations are more general than Euclidean transformations, since they include also other transformations like scaling or shearing. Since we are only interested in Euclidean transformations, we have to restrict the projective transformations in a second processing step. So we need an algebraic embedding which enables the restriction of the transformations on a Euclidean transformation in a better way. The common used algebra so far is the *dual quaternion* algebra, which is isomorphic to the motor algebra $\mathcal{G}^+_{3,0,1}$ [3]. But since it contains null spaces, the duality concepts of projective geometry cannot be applied any more.[3] The aim is now, to proceed to the conformal algebra, which can handle these problems. One important property of the conformal geometric algebra is that it is non-degenerate, but contains an artificially generated null space. The algebra for projective geometry is furthermore a subset of this (extended) algebra. Since the null space is artificially generated, it is possible to switch between null spaces and non-null spaces, an important fact for the next sections.

3.3. *The Conformal Geometric Algebra*

We use the *conformal geometric algebra* [18, 24] to model the geometry of our scenario for pose estimation. The use of the conformal geometric algebra is motivated by introducing stereographic projections [12].

3.3.1. Stereographic Projection. The idea behind conformal geometry is to interpret points as *stereographically projected* points. Simply speaking, a stereographic projection is one way to make a flat map of the earth. Taking the earth as a 3D sphere, any map must distort shapes or sizes to some degree. The rule for a stereographic projection has a nice geometric description and is visualized for the 1D case in Fig. 2: Think of the earth as a transparent sphere, intersected on the equator by an *equatorial plane*. Now imagine a light bulb at the *north pole* $\boldsymbol{n}$, which shines through the sphere. Each point on the sphere casts a shadow on the paper, and that is where it is drawn on the map. A visualization for the 2D case is shown in Fig. 5. Before introducing a formalization in terms of geometric algebra, we want to repeat the basic formulas for projecting points in space on the sphere and vice versa,

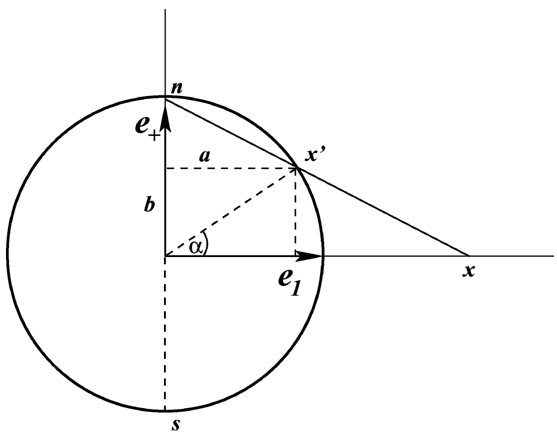


Figure 2. Visualization of a stereographic projection for the 1D case: Points on the circle are projected onto the line. Note that the north pole $\boldsymbol{n}$ projects to the points at infinity, and the south pole $\boldsymbol{s}$ projects to the origin.

e.g. given in [26]. To simplify the calculations, we will restrict ourselves to the 1-D case, as shown in Fig. 2. We assume two orthonormal basis vectors $\{\mathbf{e}_1, \mathbf{e}_+\}$ and assume the radius of the circle as $\rho = 1$. Note that $\mathbf{e}_+$ is an additional vector to the one-dimensional vector space spanned by $\mathbf{e}_1$ with $\mathbf{e}_+^2 = \mathbf{e}_1^2 = 1$.

To project a point $\boldsymbol{x}' = a\mathbf{e}_1 + b\mathbf{e}_+$ on the sphere onto the $\mathbf{e}_1$-axis, the interception theorems can be applied to obtain

$$\boldsymbol{x} = \left(\frac{a}{1-b}\right)\mathbf{e}_1 + 0\mathbf{e}_+. \tag{3.42}$$

To project a point $x\mathbf{e}_1$ ($x \in \mathbb{R}$) onto the circle we have to estimate the appropriate factors $a, b \in [0, \ldots, 1]$. The vector $\boldsymbol{x}'$ can be expressed as

$$\begin{aligned} \boldsymbol{x}' &= a\mathbf{e}_1 + b\mathbf{e}_+ \\ &= \frac{2x}{x^2+1}\mathbf{e}_1 + \frac{x^2-1}{x^2+1}\mathbf{e}_+, \end{aligned} \tag{3.43}$$

and using homogeneous coordinates this leads to a homogeneous representation of the point on the circle as

$$\boldsymbol{x}' = x\mathbf{e}_1 + \frac{1}{2}(x^2-1)\mathbf{e}_+ + \frac{1}{2}(x^2+1)\mathbf{e}_3. \tag{3.44}$$

Thus, the vector $\boldsymbol{x}$ is mapped to

$$\boldsymbol{x} \Rightarrow \boldsymbol{x}' = a\mathbf{e}_1 + b\mathbf{e}_+ + \mathbf{e}_3. \tag{3.45}$$

We define $\mathbf{e}_3$ to have a negative signature, and therefore replace $\mathbf{e}_3$ with $\mathbf{e}_-$, whereby $\mathbf{e}_-^2 = -1$. This has the advantage that in addition to using a homogeneous representation of points, we are also working in a Minkowski space. Euclidean points, stereographically projected onto the circle in Fig. 2, are then represented by the set of null vectors in our new space. That is, we have the mapping

$$\boldsymbol{x} \Rightarrow \boldsymbol{x}' = a\mathbf{e}_1 + b\mathbf{e}_+ + \mathbf{e}_-, \tag{3.46}$$

with

$$(\boldsymbol{x}')^2 = a^2 + b^2 - 1 = 0 \tag{3.47}$$

since (a, b) are the coordinates of a point on the unit circle. Note that each point in Euclidean space is in fact represented by a line of null vectors in the new space: the scaled versions of the null vector on the unit sphere. In [24] it is shown that the conformal group of n-dimensional Euclidean space $\mathbb{R}^n$ is isomorphic to the

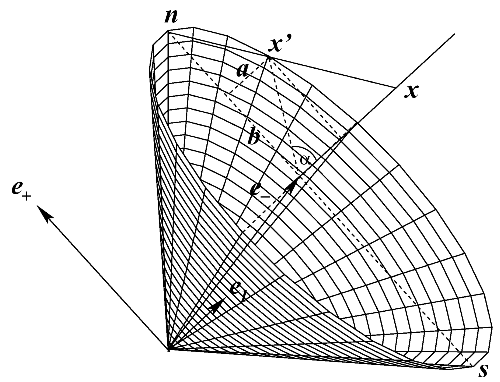


Figure 3. Visualization of the homogeneous model for stereographic projections for the 1D case. All stereographic projected points are on a cone, which is a null-cone in the Minkowski space. Note that in comparison to Fig. 2, the coordinate axes are rotated and perspective drawn.

Lorentz group of $\mathbb{R}^{n+1,1}$. Furthermore, the geometric algebra $\mathcal{G}_{n+1,1}$ of $\mathbb{R}^{n+1,1}$ has a spinor representation of the Lorentz group. Therefore, any conformal transformation of n-dimensional Euclidean space is represented by a spinor in $\mathcal{G}_{n+1,1}$, the conformal geometric algebra. Figure 3 visualizes the homogeneous model for stereographic projections for the 1D case.

Substituting the expressions for a and b from Eq. (3.43) into Eq. (3.46), we get

$$\boldsymbol{x}' = x\mathbf{e}_1 + \frac{1}{2}(x^2 - 1)\mathbf{e}_+ + \frac{1}{2}(x^2 + 1)\mathbf{e}_-. \quad (3.48)$$

This homogeneous representation of a point is used as point representation in the conformal geometric algebra. We will show this in the next section. Note that the stereographic projection from a plane leads to points on a sphere. Therefore, we can use (special) rotations on this sphere to model e.g. translations in the world or rigid body motion as coupled rotation/translation. Since we also use a homogeneous embedding, we have furthermore the possibility to model projective geometry.

3.3.2. Definition of the Conformal Geometric Algebra. To introduce CGA we follow [24] and start with a *Minkowski plane*, $\mathcal{G}_{1,1}$, whose vector space $\mathbb{R}^{1,1}$ has the orthonormal basis $\{\mathbf{e}_+, \mathbf{e}_-\}$, defined by the properties

$$\mathbf{e}_+^2 = 1 \quad \mathbf{e}_-^2 = -1 \quad \mathbf{e}_+ \cdot \mathbf{e}_- = 0. \quad (3.49)$$

In addition, a *null basis* can now be introduced by the vectors

$$\mathbf{e}_0 := \frac{1}{2}(\mathbf{e}_- - \mathbf{e}_+) \quad \text{and} \quad \mathbf{e} := \mathbf{e}_- + \mathbf{e}_+. \quad (3.50)$$

These vectors can be interpreted as the origin, $\mathbf{e}_0$, of the coordinate system and the point at infinity, $\mathbf{e}$, respectively. Note that this is in consistency with Fig. 3: $\mathbf{e}_0$ corresponds to the south pole and $\mathbf{e}$ corresponds to the north pole in homogeneous coordinates. Furthermore, we define

$$\boldsymbol{E} := \mathbf{e} \wedge \mathbf{e}_0 = \mathbf{e}_+ \wedge \mathbf{e}_-.$$

For these elements the following straightforwardly proved properties can be summarized as

$$\begin{array}{lll} \mathbf{e}_0^2 = \mathbf{e}^2 = 0 & \mathbf{e} \cdot \mathbf{e}_0 = -1 & \boldsymbol{E} = \mathbf{e}_+\mathbf{e}_- \\ \boldsymbol{E}^2 = 1 & \boldsymbol{E}\mathbf{e} = -\mathbf{e} & \boldsymbol{E}\mathbf{e}_0 = \mathbf{e}_0 \\ \mathbf{e}_+\boldsymbol{E} = \mathbf{e}_- & \mathbf{e}_-\boldsymbol{E} = \mathbf{e}_+ & \mathbf{e}_+\mathbf{e} = \boldsymbol{E} + 1 \\ \mathbf{e}_-\mathbf{e} = -(\boldsymbol{E} + 1) & \mathbf{e} \wedge \mathbf{e}_- = \boldsymbol{E} & \mathbf{e}_+ \cdot \mathbf{e} = 1. \end{array} \quad (3.51)$$

The role of the Minkowski plane is to generate null vectors, and so to extend a Euclidean vector space $\mathbb{R}^n$ to $\mathbb{R}^{n+1,1} = \mathbb{R}^n \oplus \mathbb{R}^{1,1}$ and, thus, resulting in the conformal geometric algebra $\mathcal{G}_{n+1,1}$. The conformal vector space derived from $\mathbb{R}^3$ is denoted as $\mathbb{R}^{4,1}$. A basis is given by $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_+, \mathbf{e}_-\}$. The corresponding algebra $\mathcal{G}_{4,1}$ contains $2^5 = 32$ elements. We denote the conformal unit pseudoscalar as

$$\boldsymbol{I}_C = \mathbf{e}_{+-123} = \boldsymbol{E}\boldsymbol{I}_E. \quad (3.52)$$

In this algebra we consider points of the so-called null cone, which fulfill the properties

$$\{\underline{\boldsymbol{X}} \in \mathbb{R}^{4,1} \mid \underline{\boldsymbol{X}}^2 = 0, \underline{\boldsymbol{X}} \cdot \mathbf{e} = -1\}. \quad (3.53)$$

The points of the null cone are related to those of the Euclidean space by

$$\underline{\boldsymbol{X}} = \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2\mathbf{e} + \mathbf{e}_0. \quad (3.54)$$

Evaluating $\underline{\boldsymbol{X}}$ leads to

$$\begin{aligned} \underline{\boldsymbol{X}} &= \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2\mathbf{e} + \mathbf{e}_0 \\ &= \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2(\mathbf{e}_+ + \mathbf{e}_-) + \frac{1}{2}(\mathbf{e}_- - \mathbf{e}_+) \\ &= \boldsymbol{x} + \left(\frac{1}{2}\boldsymbol{x}^2 - \frac{1}{2}\right)\mathbf{e}_+ + \left(\frac{1}{2}\boldsymbol{x}^2 + \frac{1}{2}\right)\mathbf{e}_-. \end{aligned} \quad (3.55)$$

This is exactly the homogeneous representation of a stereographic projected point, given in (3.48). The basis vectors $\{\mathbf{e}, \mathbf{e}_0\}$ only allow for a more compact representation of vectors than when using $\{\mathbf{e}_+, \mathbf{e}_-\}$.

We will now analyze new characteristic properties of the points, and so of the generated entities from these *points*.

3.3.3. Geometric Entities in Conformal Geometric Algebra. The use of a certain geometric algebra induces an involved metric and therewith a *basis geometric entity* from which the other entities are derived. In $\mathcal{G}_3$, the algebra of the Euclidean space, the basis entities are points, and lines and planes are formulated as certain sets of points. In the motor algebra $\mathcal{G}^+_{3,0,1}$, an algebra to model kinematics [3], the basis entities are lines, expressed in terms of the Plücker coordinates [5], and points and planes are written in these terms. In conformal geometric algebra, $\mathcal{G}_{4,1}$, the spheres are the basis entities [26] from which the other entities are derived. It turns out that the above introduced point representation is nothing more than a degenerate sphere.

To introduce primitive geometric entities in CGA we will start by introducing the representation of spheres in CGA. Then we will proceed to the other entities. A more detailed introduction can be found in [24].

There is no direct way to describe spheres as compact entities in $\mathcal{G}_3$. The only possibility to define them is given by formulating a constraint equation. The equation for a point, $\boldsymbol{x} \in \mathcal{G}_3$, on a sphere with center $\boldsymbol{p} \in \mathcal{G}_3$ and radius $\rho \in \mathbb{R}$, $\rho \geq 0$, can be written as

$$\begin{aligned} &(\boldsymbol{x} - \boldsymbol{p})^2 = \rho^2 \\ \Leftrightarrow\ &\boldsymbol{x}^2 - (\boldsymbol{x}\boldsymbol{p} + \boldsymbol{p}\boldsymbol{x}) + \boldsymbol{p}^2 = \rho^2. \end{aligned} \tag{3.56}$$

The basis entities of the 3D conformal space are spheres $\underline{\boldsymbol{S}}$, containing the center $\boldsymbol{p}$ and the radius ρ, $\underline{\boldsymbol{S}} = \boldsymbol{p} + \frac{1}{2}(\boldsymbol{p}^2 - \rho^2)\mathbf{e} + \mathbf{e}_0$. The point $\underline{\boldsymbol{X}} = \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2\mathbf{e} + \mathbf{e}_0$ is nothing more than a degenerate sphere with radius $\rho = 0$, which can easily be seen from the representation of a sphere. In $\mathcal{G}_{4,1}$ Eq. (3.56) can therefore be represented more compact:

$$\begin{aligned} &(\boldsymbol{x} - \boldsymbol{p})^2 = \rho^2 \\ \Leftrightarrow\ &\underline{\boldsymbol{X}} \cdot \underline{\boldsymbol{S}} = 0. \end{aligned} \tag{3.57}$$

This can easily be verified,

$$\begin{aligned} \underline{\boldsymbol{X}} \cdot \underline{\boldsymbol{S}} &= \left(\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2\mathbf{e} + \mathbf{e}_0\right) \cdot \left(\boldsymbol{p} + \frac{1}{2}(\boldsymbol{p}^2 - \rho^2)\mathbf{e} + \mathbf{e}_0\right) \\ &= -\frac{1}{2}(\boldsymbol{x}^2 + \boldsymbol{p}^2 - \rho^2) + \boldsymbol{x} \cdot \boldsymbol{p} \\ &= -\frac{1}{2}((\boldsymbol{x} - \boldsymbol{p})^2 - \rho^2). \end{aligned} \tag{3.58}$$

The dual form for a sphere is $\underline{\boldsymbol{S}}^\star$. The advantage of the dual form is that $\underline{\boldsymbol{S}}^\star$ can be calculated directly from points on the sphere: For four points on the sphere, $\underline{\boldsymbol{S}}^\star$ can be written as

$$\underline{\boldsymbol{S}}^\star = \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}} \wedge \underline{\boldsymbol{C}} \wedge \underline{\boldsymbol{D}}, \tag{3.59}$$

and a point $\underline{\boldsymbol{X}}$ is on a sphere $\underline{\boldsymbol{S}}$ iff $\underline{\boldsymbol{X}} \wedge \underline{\boldsymbol{S}}^\star = 0$. Note: To test incidence of a point with an entity can be expressed by the *inner product null-space* or *outer product null-space*, dependent on the representation or dual representation of the entity. This follows from the easy relationship (see e.g. [15])

$$\begin{aligned} &\underline{\boldsymbol{X}} \cdot \underline{\boldsymbol{S}} = 0 \\ \Leftrightarrow\ &\underline{\boldsymbol{X}} \wedge \underline{\boldsymbol{S}}^\star = 0. \end{aligned} \tag{3.60}$$

So far we have introduced the description of the first two entities, points and spheres.

Geometrically, a circle $\underline{\boldsymbol{Z}}$ can be described by the intersection of two spheres. This means:

$$\underline{\boldsymbol{X}} \in \underline{\boldsymbol{Z}} \Leftrightarrow \underline{\boldsymbol{X}} \in \underline{\boldsymbol{S}_1} \quad \text{and} \quad \underline{\boldsymbol{X}} \in \underline{\boldsymbol{S}_2}. \tag{3.61}$$

Since $\underline{\boldsymbol{S}_1}$ and $\underline{\boldsymbol{S}_2}$ can be assumed as linear independent, we can write

$$\begin{aligned} &\underline{\boldsymbol{X}} \in \underline{\boldsymbol{Z}} \\ \Leftrightarrow\ &(\underline{\boldsymbol{X}} \cdot \underline{\boldsymbol{S}_1})\underline{\boldsymbol{S}_2} - (\underline{\boldsymbol{X}} \cdot \underline{\boldsymbol{S}_2})\underline{\boldsymbol{S}_1} = 0 \\ \Leftrightarrow\ &\underline{\boldsymbol{X}} \cdot \underbrace{(\underline{\boldsymbol{S}_1} \wedge \underline{\boldsymbol{S}_2})}_{\underline{Z}} = 0 \\ \Leftrightarrow\ &\underline{\boldsymbol{X}} \cdot \underline{\boldsymbol{Z}} = 0 \end{aligned} \tag{3.62}$$

This means that algebraically a circle can be expressed as the intersection of two spheres. Figure 4 visualizes the generation of a circle as intersection of two spheres. The intersection of the circle with a third sphere leads to a point pair.

In the dual form circles are geometrically defined by three points on it,

$$\underline{\boldsymbol{Z}}^\star = \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}} \wedge \underline{\boldsymbol{C}}. \tag{3.63}$$

Table 3. The entities and their dual representations in CGA.

Entity	Representation	Grade	Dual representation	Grade
Sphere	$\underline{\boldsymbol{S}} = \boldsymbol{p} + \frac{1}{2}(\boldsymbol{p}^2 - \rho^2)\mathbf{e} + \mathbf{e}_0$	1	$\underline{\boldsymbol{S}}^\star = \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}} \wedge \underline{\boldsymbol{C}} \wedge \underline{\boldsymbol{D}}$	4
Point	$\underline{\boldsymbol{X}} = \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2\mathbf{e} + \mathbf{e}_0$	1	$\underline{\boldsymbol{X}}^\star = (-\boldsymbol{E}\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^2\mathbf{e} + \mathbf{e}_0)\boldsymbol{I}_E$	4
Plane	$\underline{\boldsymbol{P}} = \boldsymbol{n}\boldsymbol{I}_E - d\mathbf{e}$	1	$\underline{\boldsymbol{P}}^\star = \mathbf{e} \wedge \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}} \wedge \underline{\boldsymbol{C}}$	4
	$\boldsymbol{n} = (\boldsymbol{a} - \boldsymbol{b}) \wedge (\boldsymbol{a} - \boldsymbol{c})$			
	$d = (\boldsymbol{a} \wedge \boldsymbol{b} \wedge \boldsymbol{c})\boldsymbol{I}_E$			
Line	$\underline{\boldsymbol{L}} = \boldsymbol{r}\boldsymbol{I}_E + \mathbf{e}\boldsymbol{m}\boldsymbol{I}_E$	2	$\underline{\boldsymbol{L}}^\star = \mathbf{e} \wedge \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}}$	3
	$\boldsymbol{r} = \boldsymbol{a} - \boldsymbol{b}$			
	$\boldsymbol{m} = \boldsymbol{a} \wedge \boldsymbol{b}$			
Circle	$\underline{\boldsymbol{Z}} = \underline{\boldsymbol{S}_1} \wedge \underline{\boldsymbol{S}_2}$	2	$\underline{\boldsymbol{Z}}^\star = \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}} \wedge \underline{\boldsymbol{C}}$	3
	$\underline{\boldsymbol{P}_z} = \underline{\boldsymbol{Z}} \cdot \mathbf{e}, \underline{\boldsymbol{L}_z^\star} = \underline{\boldsymbol{Z}} \wedge \mathbf{e}$			
	$\underline{\boldsymbol{P}_z} = \underline{\boldsymbol{P}_z} \vee \underline{\boldsymbol{L}_z}, \rho = \dfrac{\underline{\boldsymbol{Z}}^2}{(\mathbf{e} \wedge \underline{\boldsymbol{Z}})^2}$			
Point pair	$\underline{\boldsymbol{PP}} = \underline{\boldsymbol{S}_1} \wedge \underline{\boldsymbol{S}_2} \wedge \underline{\boldsymbol{S}_3}$	3	$\underline{\boldsymbol{PP}}^\star = \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}}, \underline{\boldsymbol{X}}^\star = \mathbf{e} \wedge \underline{\boldsymbol{X}}$	2

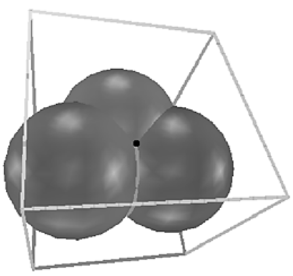


Figure 4. A circle can be expressed as intersection of two spheres. Intersecting the circle with a third sphere leads to two points (only one of these two points is visible).

Evaluating the outer products of three points leads to

$$\underline{\boldsymbol{Z}}^\star = \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}} \wedge \underline{\boldsymbol{C}} = A + A^-\mathbf{e} + A^+\mathbf{e}_0 + A^\pm\boldsymbol{E}, \tag{3.64}$$

with

$$\begin{aligned} A &= \boldsymbol{a} \wedge \boldsymbol{b} \wedge \boldsymbol{c} \\ A^- &= \frac{1}{2}(\boldsymbol{c}^2(\boldsymbol{a} \wedge \boldsymbol{b}) - \boldsymbol{b}^2(\boldsymbol{a} \wedge \boldsymbol{c}) + \boldsymbol{a}^2(\boldsymbol{b} \wedge \boldsymbol{c})) \\ A^+ &= \boldsymbol{a} \wedge \boldsymbol{b} + \boldsymbol{b} \wedge \boldsymbol{c} - \boldsymbol{a} \wedge \boldsymbol{c} \\ A^\pm &= \frac{1}{2}(\boldsymbol{a}(\boldsymbol{b}^2 - \boldsymbol{c}^2) + \boldsymbol{b}(\boldsymbol{c}^2 - \boldsymbol{a}^2) + \boldsymbol{c}(\boldsymbol{a}^2 - \boldsymbol{b}^2)). \end{aligned}$$

The dual form of lines are represented by the outer product of two points on the line and the point at infinity (see [26]), $\underline{\boldsymbol{L}}^\star = \mathbf{e} \wedge \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}}$. Since the outer product of three points determines a circle [24], the line can be interpreted as a circle passing through the point at infinity.

Similar to lines, dual planes can then be defined by the outer product of three points on the plane and the point at infinity, $\underline{\boldsymbol{P}}^\star = \mathbf{e} \wedge \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}} \wedge \underline{\boldsymbol{C}}$. A plane is a degenerate sphere, containing the point at infinity.

An overview of the definitions of the entities, their dual representations and their grades are given in Table 3. Since the outer product of 3 spheres leads to a point pair, it is a 2-blade in its dual space. Using the point at infinity leads to another representation of a pure point $\underline{\boldsymbol{X}}^\star = \mathbf{e} \wedge \underline{\boldsymbol{X}}$ in the dual space.

The dual lines and planes are given, similar to $\mathcal{G}_{3,1}$, by the Plücker coordinates of lines (direction $\boldsymbol{r}$ and moment $\boldsymbol{m}$) and the Hesse formulation (normal $\boldsymbol{n}$ and directed distance $d\boldsymbol{I}_E$) of planes, respectively.

The entities have now the following grades: points, spheres and planes are 1-blades, and lines and circles are 2-blades. Due to the fact that lines and planes are mostly generated by points on these entities, we will work with the dual representations of lines and planes in the next sections.

3.3.4. Conformal Transformations. In CGA, any conformal transformation can be expressed in the form

$$\sigma\underline{\boldsymbol{X}'} = G\underline{\boldsymbol{X}}G^{-1}, \tag{3.65}$$

where G is a versor and σ a scalar. Since the null cone is invariant under G, i.e. $(\underline{\boldsymbol{X}'})^2 = \underline{\boldsymbol{X}}^2 = 0$, we have to apply a scale factor σ to ensure $\underline{\boldsymbol{X}'} \cdot \mathbf{e} = \underline{\boldsymbol{X}} \cdot \mathbf{e} = -1$.

Table 4. Table of conformal transformations, the versors and scaling parameters.

Type	$G(\boldsymbol{x})$ on $\mathbb{R}^n$	Versor in $\mathbb{R}_{n+1,1}$	σ
Reflection	$-\boldsymbol{nxn} + 2\boldsymbol{n}\delta$	$\boldsymbol{V} = \boldsymbol{n} + \mathbf{e}\delta$	1
Inversion	$\frac{\rho^2}{\boldsymbol{x}-\boldsymbol{c}} + \boldsymbol{c}$	$\boldsymbol{V} = \boldsymbol{c} - \frac{1}{2}\rho^2\mathbf{e}$	$\left(\frac{\boldsymbol{x}-\boldsymbol{c}}{\rho}\right)^2$
Rotation	$\boldsymbol{RxR}^{-1}$	$\boldsymbol{R} = \exp\left(-\frac{\theta}{2}\boldsymbol{n}\right)$	1
Translation	$\boldsymbol{x} - \boldsymbol{t}$	$\boldsymbol{T}_t = 1 + \frac{1}{2}\boldsymbol{t}\mathbf{e}$	1
Transversion	$\frac{\boldsymbol{x}-\boldsymbol{x}^2\boldsymbol{t}}{\sigma(\boldsymbol{x})}$	$\boldsymbol{K}_t = 1 + \boldsymbol{t}\mathbf{e}_0$	$1 - 2\boldsymbol{t}\cdot\boldsymbol{x} + \boldsymbol{x}^2\boldsymbol{t}^2$
Dilation	$\lambda\boldsymbol{x}$	$\boldsymbol{D}_\lambda = \exp\left(-\frac{1}{2}\boldsymbol{E}(\ln\lambda)\right)$	λ^{-1}
Involution	$\boldsymbol{x}^\star = -\boldsymbol{x}$	$\boldsymbol{E}$	-1

Table 4, taken from [24], summarizes the conformal transformations. The first row shows the type of operation performed with the versor product. The second row shows as example the result of a transformation acting on a point. The third row shows the versor, which has to be applied and the last row shows the scaling parameter which is (sometimes) needed, to result in a homogeneous point and ensure the scaling $\underline{\boldsymbol{X}}' \cdot \mathbf{e} = \underline{\boldsymbol{X}} \cdot \mathbf{e} = -1$. As we see, any conformal transformation covers several more simple geometric transformations. In Table 4, a reflection is expressed with respect to a hyperplane with unit normal $\boldsymbol{n}$ and signed distance δ. The inversion is expressed for a circle of radius ρ centered at point $\boldsymbol{c}$. A transversion can be written down as an inversion followed by a translation and another inversion. The other transformations are self-explanatory. More explanations of the conformal group can also be found in [13, 26]. It is shown in e.g. [16], that in $\mathcal{G}_3$ the rotations are generated by reflections. Similarly one can ask, what does a reflection mean for the stereographic projected point. This is visualized in Fig. 5 for the 2D case: A reflection of a point $\underline{\boldsymbol{A}}$ on the sphere with respect to the 2D plane leads to a new point $\underline{\boldsymbol{B}}$ on the sphere, which corresponds to the inverse $\boldsymbol{B}$ of the point

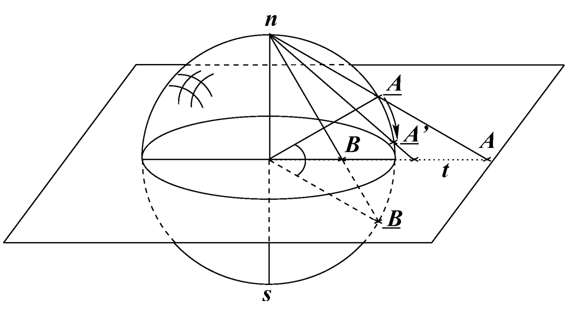


Figure 5. Visualization of an inversion and translation for a stereographic projected point in the 2D case.

$\boldsymbol{A}$ on the 2D plane. This means, that the basic operation in $\mathcal{G}_{4,1}$ is an inversion, and the other operations are derived from it. In Fig. 5 it is also shown, what a translation $\boldsymbol{t}$ of a point $\boldsymbol{A}$ on the 2D plane means for a corresponding point $\underline{\boldsymbol{A}}$ on the sphere. A translation $\boldsymbol{t}$ corresponds to a special rotation $\underline{\boldsymbol{A}} \rightarrow \underline{\boldsymbol{A}'}$. It is also easy to imagine that a rotation of a point in the 2D plane is exactly the same for its stereographically projected point on the sphere. This means, that a rotation can be calculated in the same manner as in $\mathcal{G}_2$ or $\mathcal{G}_3$ and a translation is a special rotation in $\mathcal{G}_{3,1}$ or $\mathcal{G}_{4,1}$, respectively. This is the reason why kinematics can be described in this model in a linear manner.

We will now concentrate on expressing rotations and translations in CGA.

3.3.5. Rigid Motions in Conformal Geometric Algebra. This section concerns the formulation of *rigid body motions* in CGA. As mentioned previously, a rigid body motion corresponds to the Euclidean transformation group $SE(3)$. Although being a transformation by itself, it subsumes rotation and translation. To describe a Euclidean transformation in a linear manner makes it necessary to have access on a multiplicative coupling of rotation and translation. Since the conformal transformation contains the Euclidean transformation, we can use the conformal group to express rigid body motions. Note: Though the conformal group is more general than the Euclidean group, for our pose estimation scenario it is sufficient to concentrate only on this subset of transformations.

So far, we can use rotors as elements of $\mathcal{G}_3^+$ to formalize pure rotation, but indeed it is not possible to describe general rigid body motions in this algebra in a multiplicative manner. As well as in $\mathcal{G}_3$ (see Section 3.1.2), rotations in $\mathcal{G}_{4,1}$ are represented by rotors, $\boldsymbol{R} = \exp(-\frac{\theta}{2}\boldsymbol{l})$. The components of the rotor

$\boldsymbol{R} \in \mathcal{G}_{4,1}^{+}$ are, similar to Section 3.1.2, the unit bivector $\boldsymbol{l}$ which represents the dual of the rotation axis, and the angle θ, which represents the amount of the rotation. The rotation of an entity can be performed just by multiplying the entity from the left with the rotor $\boldsymbol{R}$ and from the right with its reverse $\widetilde{\boldsymbol{R}}$. E.g., a rotation of a point can be written as $\underline{\boldsymbol{X}}' = \boldsymbol{R}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{R}}$. That this also holds for any blade (and thus for lines, planes, circles, spheres, etc.) can be seen from the easy relation

$$\begin{aligned}&\boldsymbol{R}(\underline{\boldsymbol{X}}_1 \wedge \underline{\boldsymbol{X}}_2 \wedge \dots \underline{\boldsymbol{X}}_n)\widetilde{\boldsymbol{R}}\\ &\quad = (\boldsymbol{R}\underline{\boldsymbol{X}}_1\widetilde{\boldsymbol{R}}) \wedge (\boldsymbol{R}\underline{\boldsymbol{X}}_2\widetilde{\boldsymbol{R}}) \wedge \dots (\boldsymbol{R}\underline{\boldsymbol{X}}_n\widetilde{\boldsymbol{R}}).\end{aligned} \tag{3.66}$$

If we want to translate an entity with respect to a translation vector $\boldsymbol{t} \in \langle \mathcal{G}_3 \rangle_1$, we can use a so called *translator*, $\boldsymbol{T} \in \mathcal{G}_{4,1}^{+}$, $\boldsymbol{T} = (1 + \frac{\mathbf{e}\boldsymbol{t}}{2}) = \exp(\frac{\mathbf{e}\boldsymbol{t}}{2})$. As mentioned previously, a translator is a special rotor and given in a null space since $\mathbf{e}^2 = 0$. Similar to rotations we can translate entities by multiplying the entity from the left with the translator $\boldsymbol{T}$ and with its reverse $\widetilde{\boldsymbol{T}}$ from the right,

$$\underline{\boldsymbol{X}}' = \boldsymbol{T}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{T}}. \tag{3.67}$$

To express a rigid body motion, we can apply rotors and translators consecutively. We denote such an operator,

$$\boldsymbol{M} = \boldsymbol{T}\boldsymbol{R}, \tag{3.68}$$

it is a special even-grade multivector, as a motor, which is an abbreviation of "moment and vector" [2]. The rigid body motion of e.g. a point $\underline{\boldsymbol{X}}$ can be written as

$$\underline{\boldsymbol{X}}' = \boldsymbol{M}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{M}}, \tag{3.69}$$

see also [17].

This formalization of a rigid displacement can not only be applied to points or lines (see [30]), but to all entities, contained in Table 3. Furthermore, the transformation rule is the same for all entities of Table 3. This is in contrast to a former definition of motors in the frame of motor algebra [2, 3], the algebra $\mathcal{G}_{3,0,1}^{+}$, which is formulating kinematics in a space composed of lines and which is isomorphic to the dual quaternion algebra. Although Eq. (3.68) is a valid definition of a motor in both the motor algebra and CGA, its behavior with respect to different entities is quite different. Compared with the motor algebra, in CGA we do not have to make any sign changes, depending on the entity, where the motor has to act on. This makes several case decisions in the previous formalizations of kinematics unnecessary and thus, the calculations will become more easy. The reason for that increased symmetry of a motor action lies in our chosen algebraic embedding.

3.3.6. Twist and Screw Transformations. Now follows a further definition of a motor in CGA based on the so-called *twists*. Every rigid body motion can be expressed as a twist or screw motion [25], which is a rotation around a line in space (in general not passing through the origin)[4] combined with a translation along this line. In CGA it is possible to use the rotors and translators to express screw motions in space. We will start with the formalization of general rotations and then continue with screw motions. It will turn out that a general rotation is a special case of a screw motion and its generator is directly connected to the representation of a 3D line.

To model a rotation of a point $\underline{\boldsymbol{X}}$ around an arbitrary line $\underline{\boldsymbol{L}}$ in the space, the general idea is to translate the point $\underline{\boldsymbol{X}}$ with the distance vector between the line $\underline{\boldsymbol{L}}$ and the origin, to perform a rotation and to translate the transformed point back. So a motor $\boldsymbol{M} \in \mathcal{G}_{4,1}^{+}$ describing a general rotation has the form

$$\boldsymbol{M} = \boldsymbol{T}\boldsymbol{R}\widetilde{\boldsymbol{T}}, \tag{3.70}$$

denoting the inverse translation, rotation and back translation, respectively. Using the exponential form of the translator and rotor leads to[5]

$$\begin{aligned}\boldsymbol{M} &= \boldsymbol{T}\boldsymbol{R}\widetilde{\boldsymbol{T}}\\ &= \exp\left(\frac{\mathbf{e}\boldsymbol{t}}{2}\right)\exp\left(-\frac{\theta}{2}\boldsymbol{l}\right)\exp\left(-\frac{\mathbf{e}\boldsymbol{t}}{2}\right)\\ &= \left(1 + \frac{\mathbf{e}\boldsymbol{t}}{2}\right)\exp\left(-\frac{\theta}{2}\boldsymbol{l}\right)\left(1 - \frac{\mathbf{e}\boldsymbol{t}}{2}\right)\\ &= \exp\left(\left(1 + \frac{\mathbf{e}\boldsymbol{t}}{2}\right)\left(-\frac{\theta}{2}\boldsymbol{l}\right)\left(1 - \frac{\mathbf{e}\boldsymbol{t}}{2}\right)\right)\\ &= \exp\left(-\frac{\theta}{2}(\boldsymbol{l} + \mathbf{e}(\boldsymbol{t} \cdot \boldsymbol{l}))\right).\end{aligned} \tag{3.71}$$

This formulation corresponds to the one for a general rotation given in [24]. Merely an exponential representation of the motor is used since then it is more easy to calculate its derivative.

It is interesting to mention that the exponential part of the motor $\boldsymbol{M} = \boldsymbol{T}\boldsymbol{R}\widetilde{\boldsymbol{T}}$ consists directly of the line components to rotate the entities around. To show this property, firstly the description of a dual line $\underline{\boldsymbol{L}}^{\star}$ is recalled,

$$\begin{aligned}\underline{\boldsymbol{L}}^{\star} &= \mathbf{e} \wedge \underline{\boldsymbol{A}} \wedge \underline{\boldsymbol{B}} \\ &= \mathbf{e}(\boldsymbol{a} \wedge \boldsymbol{b}) + (\boldsymbol{b} - \boldsymbol{a})\boldsymbol{E}.\end{aligned} \tag{3.72}$$

Using the unit direction $\boldsymbol{n}$ and the plumb point $\boldsymbol{t}$ of the origin to the line leads to the line representation

$$\underline{\boldsymbol{L}}^{\star} = \mathbf{e}(\boldsymbol{t} \wedge \boldsymbol{n}) + \boldsymbol{n}\boldsymbol{E}. \tag{3.73}$$

Multiplying the dual line $\underline{\boldsymbol{L}}^{\star}$ with $\boldsymbol{I}_C$ (from the right) results in

$$\begin{aligned}(\mathbf{e}(\boldsymbol{t} \wedge \boldsymbol{n}) + \boldsymbol{n}\boldsymbol{E})\boldsymbol{I}_C &= (\mathbf{e}(\boldsymbol{t} \wedge \boldsymbol{n})\boldsymbol{E}\boldsymbol{I}_E + \boldsymbol{n}\boldsymbol{E})\boldsymbol{E}\boldsymbol{I}_E \\ &= \mathbf{e}(\boldsymbol{t} \wedge \boldsymbol{n})\boldsymbol{I}_E + \boldsymbol{n}\boldsymbol{I}_E \\ &= \mathbf{e}(\boldsymbol{t} \cdot (\boldsymbol{n}\boldsymbol{I}_E)) + \boldsymbol{n}\boldsymbol{I}_E \\ &= \mathbf{e}(\boldsymbol{t} \cdot \boldsymbol{l}) + \boldsymbol{l},\end{aligned} \tag{3.74}$$

since the direction $\boldsymbol{n}$ of the line corresponds to the dual of the rotation plane $\boldsymbol{l}$, $\boldsymbol{n} = \boldsymbol{l}^{\star}$.

Vice versa: Given the dual line $\underline{\boldsymbol{L}}^{\star}$ (with unit direction) in the space, the corresponding motor describing a general rotation around this line is given by

$$\begin{aligned}\boldsymbol{M} &= \exp\left(-\frac{\theta}{2}\underline{\boldsymbol{L}}^{\star}\boldsymbol{I}_C\right) \\ &= \exp\left(-\frac{\theta}{2}\underline{\boldsymbol{L}}\right).\end{aligned} \tag{3.75}$$

Note that the line $\underline{\boldsymbol{L}}$ must be scaled with respect to the direction, $\|\boldsymbol{n}\| = 1$, since the scaling of the line is directly connected to the amount of the rotation θ. This shows that a line is a generator of a general rotation. We will now continue with screw motions.

Screw motions can be used to describe rigid body motions. Already as early as 1830 Chasles proved that every rigid body motion can be realized by a rotation around an axis combined with a translation parallel to that axis, see also [25, 30]. This is called a *screw motion*. The infinitesimal version of a screw motion is called a twist and it provides a description of the instantaneous velocity of a rigid body in terms of its linear and angular components. A screw motion is defined by an axis $\boldsymbol{l}$, a pitch h and a magnitude θ. The *pitch* of the screw is the ratio of translation to rotation, $h := \frac{d}{\theta}$ ($d, \theta \in \mathbb{R}$, $\theta \neq 0$). If $h \to \infty$, then the corresponding screw motion consists of a pure translation along the axis of the screw. The principle of a screw motion is visualized

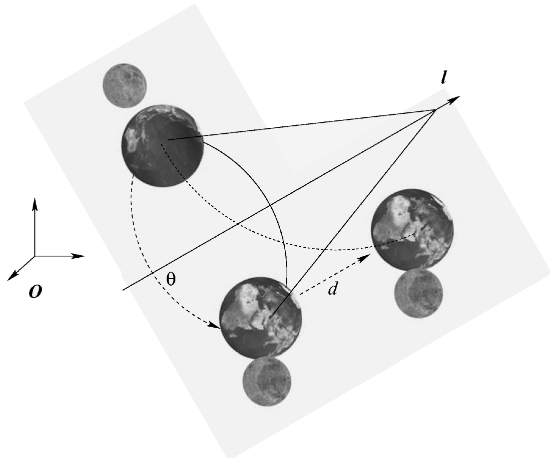


Figure 6. Visualization of a screw motion along $\boldsymbol{l}$.

in Fig. 6. To model a screw motion, the entity has to be translated during a general rotation with respect to the rotation axis. The resulting motor can be calculated in the following way,

$$\begin{aligned}\boldsymbol{M} &= \boldsymbol{T}_{d\boldsymbol{n}}\boldsymbol{T}\boldsymbol{R}\widetilde{\boldsymbol{T}} \\ &= \exp\left(\frac{\mathbf{e}d\boldsymbol{n}}{2}\right)\exp\left(-\frac{\theta}{2}(\boldsymbol{l} + \mathbf{e}(\boldsymbol{t} \cdot \boldsymbol{l}))\right) \\ &= \exp\left(\frac{\mathbf{e}d\boldsymbol{n}}{2} - \frac{\theta}{2}(\boldsymbol{l} + \mathbf{e}(\boldsymbol{t} \cdot \boldsymbol{l}))\right) \\ &= \exp\left(-\frac{\theta}{2}\left(\boldsymbol{l} + \mathbf{e}\left(\underbrace{\boldsymbol{t} \cdot \boldsymbol{l} - \frac{d}{\theta}\boldsymbol{n}}_{\boldsymbol{m}}\right)\right)\right) \\ &= \exp\left(-\frac{\theta}{2}(\boldsymbol{l} + \mathbf{e}\boldsymbol{m})\right).\end{aligned} \tag{3.76}$$

The bivector in the exponential part, $-\frac{\theta}{2}(\boldsymbol{l} + \mathbf{e}\boldsymbol{m})$, is a twist. The vector $\boldsymbol{m}$ is a vector in $\mathbb{R}^3$ which can be decomposed in an orthogonal and parallel part with respect to $\boldsymbol{n} = \boldsymbol{l}^{\star}$. If $\boldsymbol{m}$ is zero, the motor $\boldsymbol{M}$ describes a pure rotation. If $\boldsymbol{m} \perp \boldsymbol{l}^{\star}$, the motor describes a general rotation. For $\boldsymbol{m} \not\perp \boldsymbol{l}^{\star}$, the motor describes a screw motion.

4. The Pose Problem in Conformal Geometry

Let us recall Fig. 1 for the 2D-3D pose estimation problem: We assume the knowledge of a 3D object model and observe it in an image of a calibrated camera. The aim is to find the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ of the object, which leads to the best fit of the reference model with the actual extracted entities. To describe the pose scenario, it is crucial to interact entities between mathematical spaces involved in the pose problem.

4.1. *Interacting Entities in Euclidean, Projective and Conformal Geometry*

So far we are able to use CGA for the formalization of involved entities and their transformations. To formalize the scenario of Fig. 1 in a suitable way, the aim is now to describe the interaction of projective and conformal geometry. As mentioned earlier, the interaction of the different strata of the hierarchy is only poorly lit in the last years. E.g., Ruf [38] concerns this problem, but only for point features in the framework of matrix calculus. We want to extend the problem to more general object features and use in this context the conformal geometric algebra. To enable interaction between strata we use algebras for the projective and Euclidean space, respectively, as subalgebras of the CGA. It turns out that it is possible to switch between entities between conformal and projective representations by using multiplicative operators.

The main strategy to estimate the pose of the rigid object of Fig. 1 is very simple. It is summarized in Fig. 7 for the case of points: Compute the projection rays as projective reconstruction of the image points, and compare them (in the Euclidean space) with the object model points after the movement. But in detail, several algebraic transformations have to be performed: Firstly, the image entities are projective reconstructed and converted in a conformal representation. Then the model features are transformed in the conformal space. To get a distance measure in the Euclidean space, in the last step, the transformed model entities and reconstructed image features are compared by suitable scaled constraint equations.

projective plane	$\boldsymbol{a}_i^2 \rightarrow \boldsymbol{L}_{\boldsymbol{a}_i^2}$ projective reconstruction $\Longrightarrow$	*projective space*
projective space	projective conformal extension of $\boldsymbol{L}_{\boldsymbol{a}_i^2}$ $\Longrightarrow$	*conformal space*
conformal space	transformation of $\boldsymbol{X}^1 \rightarrow \boldsymbol{X}^2$ $\Longrightarrow$	*conformal space*
conformal space	comparison of $\boldsymbol{X}^2, \boldsymbol{L}_{\boldsymbol{a}_i^2}$ $\Longrightarrow$	*conformal space*
conformal space	scaling $\Longrightarrow$	*Euclidean space*

Figure 7. Involved geometric spaces of the 2D-3D pose estimation problem.

Table 5. Different mathematical spaces with their corresponding geometric algebras and point representations.

Space	Algebra	Point representation
3D Euclidean	$\mathcal{G}_{3,0}$	$\boldsymbol{x} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + x_3\mathbf{e}_3$
2D projective	$\mathcal{G}_{2,1}$	$\boldsymbol{x}_{\boldsymbol{p}_2} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + \mathbf{e}_-$
3D projective	$\mathcal{G}_{3,1}$	$\boldsymbol{X}_{\boldsymbol{p}_3} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + x_3\mathbf{e}_3 + \mathbf{e}_-$
3D kinematic	$\mathcal{G}^+_{3,0,1}$	$\boldsymbol{X} = 1 + \boldsymbol{I}(x_1\mathbf{e}_{23} + x_2\mathbf{e}_{31} + x_3\mathbf{e}_{12})$
3D conformal	$\mathcal{G}_{4,1}$	$\underline{\boldsymbol{X}} = \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2\mathbf{e} + \mathbf{e}_0$ $\underline{\boldsymbol{X}}^\star = \mathbf{e} \wedge \underline{\boldsymbol{X}}$

We can summarize the involved mathematical spaces and their corresponding geometric algebras in the following manner: The Euclidean framework can be represented by using the algebra $\mathcal{G}_{3,0}$, and $\mathcal{G}_{3,1}$ can be used to represent the projective space [19]. The projective plane is represented by the algebra $\mathcal{G}_{2,1}$. One way of defining a kinematic space is given by the motor algebra $\mathcal{G}^+_{3,0,1}$ [2, 3]. Another way is given by embedding the kinematics into the 3D conformal space represented by $\mathcal{G}_{4,1}$ [24]. Table 5 gives an overview of representations of points using different algebras. As can be seen, the relation

$$\mathcal{G}_{4,1} \supseteq \mathcal{G}_{3,1} \supseteq \mathcal{G}_{3,0} \tag{4.1}$$

is valid, but only for $\mathcal{G}_{3,0}$ limited to points. Both algebras for the projective and Euclidean space constitute subspaces of the linear space of the conformal geometric algebra. Since only points are modeled in $\mathcal{G}_{3,0}$ the *direction* of modeling the pose problem is consistent with the increasing possibilities by using higher geometric algebras: Reconstruct from the projective plane one dimensional higher entities and work in the projective or conformal space, respectively. In these spaces we have more possibilities of expressing geometry. Therefore the modeling of the pose problem follows the direction

$$\mathcal{G}_{3,0}, \mathcal{G}_{2,1} \Rightarrow \mathcal{G}_{3,1} \Rightarrow \mathcal{G}_{4,1}. \tag{4.2}$$

In the following, we will introduce operators which not only relate linear spaces of the considered algebras but guarantee the mapping between the algebraic properties. This means, we define operators which transform the representation of the entities of the conformal space into equivalent entities in the projective space, and vice versa. The possibility to change the representation of an entity enables us to pick up the advantages of each algebra, and so to use the better suited algebra for each subproblem.

4.2. *Change of Representations of Geometric Entities*

In this section it will be shown, how to transform these representations. The operators between the conformal and projective space will be denoted as *conformal projective split* and *projective conformal extension*, according to the *projective split* [19] which enables a change between the projective and the Euclidean space and the *conformal split* [17, 19] which enables a change between the Euclidean and conformal space. This means, by using these different splits and extensions, it is possible to describe the whole stratification hierarchy. This will lead to a compact formulation of the 2D-3D pose estimation problem.

We will start with the first two spaces, the conformal space to describe kinematics and conformal geometry, and the projective space which can be considered as subspace of the former one. The two operators to switch between geometric algebras representing these spaces are summarized in the following theorems:

Theorem 4.1. *To change an entity* Θ *given in the projective representation*, Θ_p, *to the conformal representation*, Θ_c, $\Theta_p \in \{\boldsymbol{X}, \boldsymbol{L}, \boldsymbol{P} \in \mathcal{G}_{3,1}\} \rightarrow \Theta_c \in \{\underline{\boldsymbol{X}}^\star, \underline{\boldsymbol{L}}^\star, \underline{\boldsymbol{P}}^\star \in \mathcal{G}_{4,1}\}$, *the following operator has to be applied*:

$$\Theta_c = \mathbf{e} \wedge \Theta_p. \tag{4.3}$$

Note. Since circles and spheres are no entities of the projective space, we can not transform them between the projective and conformal space. This leads to remarkable consequences for the pose estimation problem, discussed in the later sections.

For the proof of the theorem it is sufficient to show the simple relation $\mathbf{e} \wedge \mathbf{e}_- = \boldsymbol{E}$,

$$\begin{aligned} \mathbf{e} \wedge \mathbf{e}_- &= (\mathbf{e}_- + \mathbf{e}_+) \wedge \mathbf{e}_- \\ &= \mathbf{e}_+ \wedge \mathbf{e}_- = \boldsymbol{E}. \end{aligned} \tag{4.4}$$

To make the involved geometry more clear, we will compute the representation changes of points, lines and planes explicitly:

$$\begin{aligned} \boldsymbol{X} \in \mathcal{G}_{3,1} &= \boldsymbol{x} + \mathbf{e}_- \\ &\rightarrow \mathbf{e} \wedge (\boldsymbol{x} + \mathbf{e}_-) \\ &= \mathbf{e} \wedge \boldsymbol{x} + \mathbf{e} \wedge \mathbf{e}_- \\ &= \mathbf{e}\boldsymbol{x} + \boldsymbol{E} = \underline{\boldsymbol{X}}^\star \in \mathcal{G}_{4,1} \qquad (4.5) \\ \boldsymbol{L} \in \mathcal{G}_{3,1} &= \mathbf{e}_-\boldsymbol{r} + \boldsymbol{m} \\ &\rightarrow \mathbf{e} \wedge (\mathbf{e}_-\boldsymbol{r} + \boldsymbol{m}) \\ &= \mathbf{e}\boldsymbol{m} + \mathbf{e} \wedge (\mathbf{e}_-\boldsymbol{r}) \\ &= \boldsymbol{E}\boldsymbol{r} + \mathbf{e}\boldsymbol{m} = \underline{\boldsymbol{L}}^\star \in \mathcal{G}_{4,1} \qquad (4.6) \\ \boldsymbol{P} \in \mathcal{G}_{3,1} &= \mathbf{e}_-\boldsymbol{n} + d\boldsymbol{I}_E \\ &\rightarrow \mathbf{e} \wedge (\mathbf{e}_-\boldsymbol{n} + d\boldsymbol{I}_E) \\ &= \boldsymbol{E}\boldsymbol{n} + \mathbf{e}d\boldsymbol{I}_E = \underline{\boldsymbol{P}}^\star \in \mathcal{G}_{4,1} \qquad (4.7) \end{aligned}$$

Now, we describe how to switch representations from the conformal space into the projective space.

Theorem 4.2. *To change an entity* Θ, *given in the conformal representation*, Θ_c, *to the projective representation*, Θ_p, $\Theta_c \in \{\underline{\boldsymbol{X}}^\star, \underline{\boldsymbol{L}}^\star, \underline{\boldsymbol{P}}^\star \in \mathcal{G}_{4,1}\} \rightarrow \Theta_p \in \{\boldsymbol{X}, \boldsymbol{L}, \boldsymbol{P} \in \mathcal{G}_{3,1}\}$, *the following operator has to be applied*:

$$\Theta_p = \mathbf{e}_+ \cdot \Theta_c. \tag{4.8}$$

For the proof it is sufficient to show the following identity

$$\begin{aligned} \Theta_p &= \mathbf{e}_+ \cdot (\mathbf{e} \wedge \Theta_p), \\ \mathbf{e}_+ \cdot (\mathbf{e} \wedge \Theta_p) &= \underbrace{(\mathbf{e}_+ \cdot \mathbf{e})}_{1} \wedge \Theta_p - \mathbf{e} \wedge \underbrace{(\mathbf{e}_+ \cdot \Theta_p)}_{0} = \Theta_p. \end{aligned} \tag{4.9}$$

We call the operations "$\mathbf{e}\wedge$" and "$\mathbf{e}_+\cdot$" the *projective conformal extension* and *conformal projective split*, respectively.

The transformation between the algebras for the projective and Euclidean space is much simpler. Lines and planes can be represented in the Euclidean space, but as mentioned before, these are only artificially generated representations which are not generated by the algebra itself. As a consequence, only for points the transformation can be described in a suitable way. The transformations are leaned on Hestenes' formalization in [19] and can be written in the following way,

$$\begin{aligned} \boldsymbol{X} \rightarrow \frac{(\boldsymbol{X} \wedge \mathbf{e}_-) \cdot \mathbf{e}_-}{\boldsymbol{X} \cdot \mathbf{e}_-} &= \boldsymbol{x}, \qquad \boldsymbol{x} \in \mathcal{G}_{3,0} \\ \boldsymbol{x} \rightarrow \boldsymbol{x} + \mathbf{e}_- &= \boldsymbol{X}, \qquad \boldsymbol{X} \in \mathcal{G}_{3,1}. \end{aligned}$$

Table 6. Interaction between algebras of the Euclidean, projective and conformal space.

Euclidean space		Projective space		Conformal space
$\mathcal{G}_{3,0}$	$\subseteq$	$\mathcal{G}_{3,1}$	$\subseteq$	$\mathcal{G}_{4,1}$
Θ_e	$\xrightarrow{\boldsymbol{x}+\mathbf{e}_-}$	Θ_p	$\xrightarrow{\mathbf{e}\wedge\Theta_p}$	Θ_c
	$\xleftarrow{\frac{(\boldsymbol{X}\wedge\mathbf{e}_-)\cdot\mathbf{e}_-}{\boldsymbol{X}\cdot\mathbf{e}_-}}$		$\xleftarrow{\mathbf{e}_+\cdot\Theta_c}$	
Θ_e	$\longrightarrow$	$\mathbf{e}\wedge(\boldsymbol{x}+\mathbf{e}_-)$	$\longrightarrow$	
	$\longleftarrow$	$\frac{((\mathbf{e}_+\cdot\underline{\boldsymbol{X}})\wedge\mathbf{e}_-)\cdot\mathbf{e}_-}{(\mathbf{e}_+\cdot\underline{\boldsymbol{X}})\cdot\mathbf{e}_-}$	$\longleftarrow$	Θ_c

Table 6 gives an overview of the three main involved spaces and their interaction.

To estimate a rigid body motion of an entity given in projective geometry, we change its representation in a conformal one, compute the rigid body motion and go back to the projective space: Let Θ_p be an entity given in the projective space, and $\boldsymbol{t}$ a translation vector in Euclidean space. In conformal geometric algebra, the translator has the following structure, $\boldsymbol{T} = (1 + \frac{\mathbf{e}\boldsymbol{t}}{2})$ and $\widetilde{\boldsymbol{T}} = (1 - \frac{\mathbf{e}\boldsymbol{t}}{2})$. Then a multiplicative formulation of the translated entity in the projective space is given by

$$\Theta'_p = \mathbf{e}_+\cdot\underbrace{(\boldsymbol{T}\underbrace{(\mathbf{e}\wedge\underbrace{\Theta_p}_{\text{projective}})}_{\text{conformal}}\widetilde{\boldsymbol{T}})}_{\text{projective}}. \tag{4.10}$$

To compute joins and meets of entities, given in conformal geometric algebra, we change their representations to the projective space, perform the incidence operation and go back to the conformal space. As an example, the intersection (denoted with the operator $\vee_c$) of a line $\underline{\boldsymbol{L}}^\star$ with a plane $\underline{\boldsymbol{P}}^\star$ is given by

$$\underline{\boldsymbol{L}}^\star\vee_c\underline{\boldsymbol{P}}^\star = \underbrace{\mathbf{e}\wedge\underbrace{((\mathbf{e}_+\cdot\underbrace{\underline{\boldsymbol{L}}^\star}_{\text{conformal}})\vee(\mathbf{e}_+\cdot\underbrace{\underline{\boldsymbol{P}}^\star}_{\text{conformal}}))}_{\text{projective}}}_{\text{conformal}} \tag{4.11}$$

(4.12)

To explicitly compute the coordinates of the intersection point of two lines, $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$, given in the projective space, we intersect these lines in the projective space and use the projective split to get the intersection point in the geometric algebra of the Euclidean space,

$$\boldsymbol{x} = \underbrace{\frac{1}{\underbrace{\boldsymbol{L}_1\vee\boldsymbol{L}_2\cdot\mathbf{e}_-}_{\text{projective}}}(((\underbrace{\boldsymbol{L}_1\vee\boldsymbol{L}_2}_{\text{projective}})\wedge\mathbf{e}_-)\cdot\mathbf{e}_-)}_{\text{Euclidean}}. \tag{4.13}$$

These examples show, how to interact between the Euclidean, projective and conformal framework.

4.2.1. Pose Constraints in Conformal Geometric Algebra. This section gives a brief preview how the interaction of entities in geometric algebras will be applied on the pose problem. As mentioned earlier, the main problem in the pose scenario is, how to compare 2D image features with 3D Euclidean object features. Our constraint equations will lead to equations of the following structure (here just for point correspondences),

$$\lambda((\boldsymbol{M}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{M}})\ \underline{\times}\ \mathbf{e}\wedge(\boldsymbol{O}\wedge\boldsymbol{x}))\cdot\mathbf{e}_+ = 0. \tag{4.14}$$

The interpretation of the equation is simple an the equation can be separated in the following manner,

$$\underbrace{\lambda(\underbrace{\underbrace{(\boldsymbol{M}\underbrace{\underline{\boldsymbol{X}}}_{\substack{\text{object point}\\ \text{in conformal space}}}\widetilde{\boldsymbol{M}})}_{\text{rigid motion of the object point}}\ \underline{\times}\ \underbrace{\mathbf{e}\wedge(\underbrace{\boldsymbol{O}}_{\substack{\text{optical}\\ \text{center}}}\wedge\underbrace{\boldsymbol{x}}_{\substack{\text{image}\\ \text{point}}})}_{\substack{\text{projection ray,}\\ \text{reconstructed from the image point}\\ \text{in conformal space}}}}_{\substack{\text{collinearity of the transformed object}\\ \text{point with the reconstructed line}}})\cdot\mathbf{e}_+}_{\text{geometric distance measure between 3D line and 3D point}} = 0. \tag{4.15}$$

The mathematical spaces involved here are

$$\underbrace{\lambda(\underbrace{(\boldsymbol{M}\underbrace{\underline{\boldsymbol{X}}}_{CS}\widetilde{\boldsymbol{M}})}_{CS}\ \underline{\times}\ \underbrace{\mathbf{e}\wedge\underbrace{(\underbrace{\boldsymbol{O}}_{PS}\wedge\underbrace{\boldsymbol{x}}_{PP})}_{PS}}_{CS})}_{ES}\cdot\mathbf{e}_+ = 0. \tag{4.16}$$

Here does *PP* abbreviate *projective plane*, *PS projective space*, *CS conformal space* and *ES* the *Euclidean space*. Furthermore, will Part II [36] show that the used commutator and anti-commutator products can be used to describe a geometric distance measure, to ensure good conditioned equations in the presence of noise. This will become more clear in Part II.

The main advantages of the constraint equations are the following: Firstly, the constraints are expressed in a multiplicative manner, they are concise and easy to interpret (see Eq. (4.16)). This is the basis for further extensions, like kinematic chains and other higher order algebraic entities. Secondly, the whole geometry within the scenario is concerned and strictly modeled. This ensures an optimal treating of the geometry and the knowledge that no geometric aspects have been neglected or approximated which is sometimes done in the literature by e.g. using orthographic camera models.

5. Summary and Discussion

This work is concerned with the theoretical foundations of the 2D-3D pose estimation problem. Firstly, Faugeras' stratification hierarchy is identified as an important concept in the pose estimation problem. But since it is based solely on point concepts we introduce the conformal geometric algebra which provides a homogeneous model for stereographic geometry and is therefore well suited to deal with projective geometry on the one hand and kinematics on the other hand. The multivector concepts of geometric algebras lead to a new stratification hierarchy which contains as highest algebra the conformal geometric algebra. Since conformal geometry is not well known for solving computer vision problems, we introduce the stereographic projections and sphere concepts in some detail in the context of the pose problem.

The usefulness of this approach is as preview shown in Section 4.2.1: we gain compact constraint equations with a strict modeling of all geometric aspects. We will apply this in Part II [36] for simultaneous pose estimation of different image and object features, e.g. containing points, lines, planes, circles, spheres or kinematic chains. Our very recent work [33, 34] concerns further extensions of this approach e.g. by modeling cycloidal curves and free-form contours.

Acknowledgments

We would like to thank Oliver Granert, Daniel Grest, Norbert Krüger and Christian Perwass for fruitful discussions and hints for performing this work. This work has been supported by DFG Graduiertenkolleg No. 357 and by EC Grant IST-2001-3422 (VISATEC).

Notes

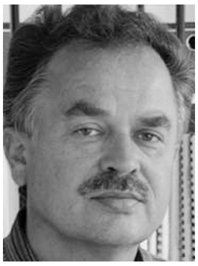
1. Problems can occur if the object is very large (e.g. a hallway) with some object features very near to the camera plane and other object features far away from the camera plane. In such situations, the spatial distance (which will be minimized) of the near objects influence the equations to a lesser extent than the far object features.
2. In this example, $\langle , \rangle$ denotes the scalar product of vectors, and $\times$ denotes the cross product.
3. The inverse pseudoscalar does not exist, since $\boldsymbol{I}^2 = 0$.
4. We call such an operation also a *general rotation*.
5. In the fourth equation is made use of the property $\boldsymbol{g}\exp(\xi)\widetilde{\boldsymbol{g}} = \exp(\boldsymbol{g}\xi\widetilde{\boldsymbol{g}})$ for $\boldsymbol{g}\widetilde{\boldsymbol{g}} = 1$. This property can be proven by an induction on the series expression of the exponential function.

References

1. H. Araujo, R.L. Carceroni, and C.M. Brown, "A fully projective formulation to improve the accuracy of Lowe's pose-estimation algorithm," *Computer Vision and Image Understanding CVIU*, Vol. 70, pp. 227–238, 1998.
2. E. Bayro-Corrochano, "The geometry and algebra of kinematics," Vol. 40, pp. 457–472, 2001.
3. E. Bayro-Corrochano, K. Daniilidis, and G. Sommer, "Motor algebra for 3D kinematics: The case of the hand-eye calibration," *Journal of Mathematical Imaging and Vision*, Vol. 13, pp. 79–100, 2000.
4. J.R. Beveridge, "Local search algorithms for geometric object recognition: Optimal correspondence and pose," Technical Report CS 93–5, University of Massachusetts, 1993.
5. W. Blaschke, "Kinematik und Quaternionen, Mathematische Monographien 4," *Deutscher Verlag der Wissenschaften*, 1960.
6. C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, 1998, pp. 8–15 .
7. A. Chiuso and G. Picci, "Visual tracking of points as estimation on the unit sphere," in *The Confluence of Vision and Control*, Springer-Verlag, 1998, pp. 90–105.
8. CLU Library, "A C++ Library for Clifford Algebra," available at http://www.perwass.de/CLU *Cognitive Systems Group, University Kiel,* 2001.
9. L. Dorst, "'The inner products of geometric algebra," in *Applied Geometric Algebras for Computer Science and Engineering (AGACSE)*, L. Dorst, C. Doran, and J. Lasenby (Eds.), Birkhäuser Verlag, 2001, pp. 35–46.
10. L. Dorst, "Honing geometric algebra for its use in the computer sciences," Vol. 40, pp. 127–152, 2001.
11. O. Faugeras, "Stratification of three-dimensional vision: Projective, affine and metric representations," *Journal of Optical Society of America*, Vol. 12, No. 3, pp. 465–484, 1995.
12. J. Gallier, *Geometric Methods and Applications for Computer Science and Engineering*, Springer Verlag: New York, 2001.

13. J.E. Gilbert and M.A.M. Murray, *Clifford Algebras and Dirac Operators in Harmonic Analysis*, Cambridge University Press, 1991.
14. W.E.L. Grimson, *Object Recognition by Computer*, The MIT Press: Cambridge, MA, 1990.
15. D. Hestenes, "The design of linear algebra and geometry," *Acta Applicandae Mathematicae*, Vol. 23, pp. 65–93, 1991.
16. D. Hestenes, "Invariant body kinematics: I. Saccadic and compensatory eye movements," *Neural Networks*, Vol. 7, pp. 65–77, 1994.
17. D. Hestenes, H. Li, and A. Rockwood, "New algebraic tools for classical geometry," Vol. 40, pp. 3–23, 2001.
18. D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus*, D. Reidel Publ. Comp., Dordrecht, 1984.
19. D. Hestenes and R. Ziegler, "Projective geometry with Clifford algebra," *Acta Applicandae Mathematicae*, Vol. 23, pp. 25–63, 1991.
20. J.R. Holt and A.N. Netravali, "Uniqueness of solutions to structure and motion from combinations of point and line correspondences," *Journal of Visual Communication and Image Representation*, Vol. 7, No. 2, pp. 126–136, 1996.
21. H.H. Homer, "Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 6, pp. 530–541, 1991.
22. R. Horaud, T.Q. Phong, and P.D. Tao, "Object pose from 2-d to 3-d point and line correspondences," *International Journal of Computer Vision*, Vol. 15, pp. 225–243, 1995.
23. H. Klingspohr, T. Block, and R.-R. Grigat, "A passive real-time gaze estimation system for human-machine interfaces," in *Computer Analysis of Images and Patterns (CAIP)*, G. Sommer, K. Daniilidis, and J. Pauli (Eds.), LNCS 1296, Springer-Verlag Heidelberg, 1997, pp. 718–725.
24. H. Li, D. Hestenes, and A. Rockwood, "Generalized homogeneous coordinates for computational geometry," Vol. 40, pp. 27–52, 2001.
25. R.M. Murray, Z. Li, and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.
26. T. Needham, *Visual Complex Analysis*, Oxford University Press, 1997.
27. C. Perwass and J. Lasenby, "A novel axiomatic derivation of geometric algebra," Technical Report CUED/F—INFENG/TR.347, Cambridge University Engineering Department, 1999.
28. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes* in C, Cambridge University Press, 1993.
29. Project homepage, Geometric Algebras and the Perception-Action Cycle See links at `http://www.ks.informatik.uni-kiel.de/modules.php?name=Projekte&file=index&func=hp&prid=14`
30. J. Rooney, "A comparison of representations of general screw displacement," *Environment and Planning*, Vol. B5, pp. 45–88, 1978.
31. B. Rosenhahn, O. Granert, and G. Sommer, "Monocular pose estimation of kinematic chains," in *Applied Geometric Algebras for Computer Science and Engineering, Birkhäuser Verlag*, L. Dorst, C. Doran, and J. Lasenby (Eds.), 2001, pp. 373–383.
32. B. Rosenhahn and J. Lasenby, "Constraint equations for 2D-3D pose estimation in conformal G algebra," Technical Report CUED/F - INFENG/TR.396, Cambridge University Engineering Department, 2000.
33. B. Rosenhahn, Ch. Perwass, and G. Sommer, "Pose estimation of 3D free-form contours," Technical Report 0207,Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik, 2002.
34. B. Rosenhahn, Ch. Perwass, and G. Sommer, "Pose estimation of 3D free-form contours in conformal geometry," in *Proceedings of Image and Vision Computing (IVCNZ)*, D. Kenwright (Ed.), New Zealand, 2002, pp. 29–34.
35. B. Rosenhahn and G. Sommer, "Adaptive pose estimation for different corresponding entities," in *Pattern Recognition, 24th DAGM Symposium*, L. Van Gool (Ed.), Springer-Verlag: Berling Heidelberg, LNCS 2449, 2002, pp. 265–273.
36. B. Rosenhahn and G. Sommer, "Pose estimation in conformal geometric algebra. Part II: Real-time pose estimation using extended feature concepts ," *Journal of Mathematical Imaging and Vision*, Vol. 22, No. 1, pp. 49–70, 2005.
37. B. Rosenhahn, Y. Zhang, and G. Sommer, "Pose estimation in the language of kinematics," in *Second International Workshop, Algebraic Frames for the Perception-Action Cycle*, G. Sommer and Y.Y. Zeevi (Eds.), LNCS 1888, Springer-Verlag, Heidelberg, 2000, pp. 284–293.
38. A. Ruf and R. Horaud, "Vision-based guidance and control of robots in projective space,"' in *Proceedings, 6th European Conference on Computer Vision (ECCV)*, Part II, Vernon D. (Ed.), LNCS 1843, Springer-Verlag Heidelberg, 2000, pp. 50–66.
39. F. Shevlin, "Analysis of orientation problems using Plücker lines," *International Conference on Pattern Recognition, Brisbane*, Vol. 1, pp. 685–689, 1998.
40. G. Sommer (Eds.), *Geometric Computing with Clifford Algebra*, Springer Verlag Heidelberg, 2001.
41. G. Sommer, B. Rosenhahn, and Y. Zhang, "Pose estimation using geometric constraints," in *Multi-Image Search and Analysis*, R. Klette, Th. Huang, and G. Gimmel'farb (Eds.), LNCS 2032, Springer-Verlag, Heidelberg, 2001, pp. 153–170.
42. A. Ude, "Filtering in a unit quaternion space for model-based object tracking," *Robotics and Autonomous Systems*, Vol. 28, Nos. 2/3, pp. 163–172, 1999.
43. M.W. Walker and L. Shao, "Estimating 3-d location parameters using dual number quaternions," *CVGIP: Image Understanding*, Vol. 54, No. 3, pp. 358–367, 1991.
44. I.M. Yaglom, Klein Felix, and Lie Sophus, *Birkhäuser*, Boston, 1988.

Bodo Rosenhahn gained his diploma degree in Computer Science in 1999. Since then he has been pursuing his Ph.D. at the Cognitive Systems Group, Institute of Computer Science, Christian-Albrechts University Kiel, Germany. He is working on geometric applications of Clifford algebras in computer vision.



Prof. Dr. Gerald Sommer received a diploma degree in physics from the Friedrich-Schiller-Universität Jena, Germany, in 1969, a Ph.D. degree in physics from the same university in 1975, and a habilitation degree in engineering from the Technical University Ilmenau, Germany, in 1988. Since 1993 he is leading the research group Cognitive Systems at the Christian-Albrechts-Universität Kiel, Germany. Currently he is also the scientific coordinator of the VISATEC project.