10 The Global Algebraic Frame of the Perception-Action Cycle

Gerald Sommer

Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel, Germany

10.1 Introduction 222
10.2 Design of behavior-based systems 223
10.2.1 From knowledge-based to behavior-based systems . . 223
10.2.2 Metaphor of biological competence 224
10.2.3 From natural to artificial behavior-based systems . . . 225
10.2.4 Bottom-up approach of design 226
10.2.5 Grounding of meaning of equivalence classes 227
10.2.6 General principles of behavior-based system design . 228
10.3 Algebraic frames of higher-order entities 230
10.3.1 Perception, action and geometry 230
10.3.2 Behavioral modulation of geometric percepts 231
10.3.3 Roots of geometric algebra 232
10.3.4 From vector spaces to multivector spaces 233
10.3.5 Properties of multivector spaces 235
10.3.6 Functions, linear transformations and mappings . . . 237
10.3.7 Qualitative operations with multivectors 238
10.3.8 Geometric algebra of the Euclidean space 239
10.3.9 Projective and kinematic spaces in geometric algebra 241
10.3.10 Geometric entities in geometric algebra 243
10.3.11 Operational entities in geometric algebra 246
10.4 Applications of the algebraic framework 249
10.4.1 Image analysis and Clifford Fourier transform 249
10.4.2 Pattern recognition and Clifford MLP 255
10.5 Summary and conclusions 259
10.6 References 260

Handbook of Computer Vision and Applications
Volume 3
Systems and Applications

ISBN 0–12–379773-X/$30.00

10.1 Introduction

In the last decade the maturation of a new family of intelligent systems has occured. These systems are designed to model the behavior-based paradigm. The roots of the paradigm can be found in the rather disparate disciplines of natural science and philosophy. From a technical point of view they can be summarized as the cybernetic interpretation of living systems on several levels of abstraction and granularity. Although not always recognized, the approach aims at the design of autonomous technical systems.

In this chapter we want to consider the design of behavior-based technical systems in the frame of a general mathematical language, that is, an algebra. This algebra should be powerful enough to integrate different contributing scientific disciplines in a unique scheme and should contribute to overcoming some of the current limitations and shortcomings inherent to those disciplines. The language we are using is Clifford algebra [1] in its geometrically interpreted version as geometric algebra [2]. The use of this algebra results from searching for the natural science principles forming the basis of systematic system design according to the behavior-based paradigm of design. Therefore, we have to consider the motivations of proceeding in this way and the ideas basic to this approach. The outline of the chapter will be as follows. In Section 10.2, we will introduce the basic assumptions of the metaphor of intelligent systems we are using. We will derive its usefulness from engineering requirements with respect to the desired features of technical systems. We will then give some arguments in support of a natural science-based design approach and some already identified key points will be discussed. One of the most important of them will be a kind of mathematical equivalence of the perception and action tasks of a system. Perception will be understood as recognition of regular spatiotemporal patterns of the perceivable environment, whose equivalence classes are verified using action. Conversely, action will be understood as generation of regular spatiotemporal patterns in the reachable environment, whose equivalence classes are verified using perception. From this mutual support of perception and action the need of a mathematical framework like that of geometric algebra will be deduced. This will be done in Section 10.3 together with a description of its basic features for modeling geometric entities and operations on these entities in Euclidean space. We will omit the analysis of manifolds. Instead, in Section 10.4 we will demonstrate the use of this algebraic language with respect to some basic problems related to the analysis of manifolds, their recognition, and transformation. There, we demonstrate the impact of the presented algebraic frame for the theory of multidimensional signals and neural computing.

We will follow a grand tour from engineering to philosophy (Section 10.2), from philosophy to natural science (Section 10.2), from natural science to mathematics (Section 10.3), and finally from mathematics to engineering (Section 10.4) with emphasis on Section 10.3.

10.2 Design of behavior-based systems

10.2.1 From knowledge-based to behavior-based systems

The old dream of human beings, that of designing so-called intelligent machines, seemed possible as a result of the famous 1956 Dartmouth conference, during which such disciplines as "artificial intelligence" and "cognitive science" emerged. Following the zeitgeist, both disciplines were rooted in the *computational theory of mind*, and for a long time they were dominated by the symbol processing paradigm of Newell and Simon [3]. In that paradigm, intelligence is the exclusive achievement of human beings and is coupled with the ability of reasoning on categories.

Because of the assumed metaphorical power of computers to be able to interpret human brain functions, it is not surprising that this view has been inverted, that is, to be put to work to design intelligent machines. The strong relations of the metaphor to symbolic representations of knowledge with respect to the domain of interest not only enforced the development of *knowledge-based* or *expert systems* (KBS) but also considerably determined the directions of development of robotics [4] and computer vision [5]. The corresponding engineering paradigm of the metaphor is called *knowledge-based system* design.

Although the knowledge-based approach to vision and robotics achieved remarkable success in man-made environments, the final result has been less than satisfactory. There is no scaling of the gained solutions to natural environments. As well, within the considered frame, system performance is limited not only from an engineering point of view. This results from the natural limitations of explicit modeling of both the domain and the task. Therefore, the systems lack robustness and adaptability. Yet these properties are the most important features of those technical systems for which engineers strive. The third drawback of the most contemporary available systems is their lack of stability.

All three properties are strongly related to the philosophy of system design. The problems result from the metaphor of intelligent machines. From a cybernetic point of view, all biological systems, whether plants, animals or human beings, are robust, adaptable and stable systems, capable of perceiving and acting in the real world. The observation of the interaction of biological systems with their environment and among themselves enables us to formulate another metaphor. This approach to intelligence is a *socioecological theory of competence*, which we will call *metaphor of biological competence*. Its engineering paradigm is called *behavior-based system* design.

10.2.2 Metaphor of biological competence

Biological competence is the capability of biological systems to do those things right that are of importance for the survival of the species and/or the individual. We use the term *competence* instead of intelligence because the last one is restricted to the ability to reason, which only human beings and some other primates possess. In contrast to *intelligence*, competence is common to all biological systems. It summarizes rather different capabilities of living organisms on several levels of consideration as those of perception, action, endocrine and thermal regulation or, of the capability of engaging in social relations.

For an observer of the system, its competence becomes visible as an adequate activity. The mentioned activity is called *behavior*, a term borrowed from ethology. Of course, behavior can also happen internally as recognition or reasoning. It can be reactive, retarded or planned. In any case, it is the answer of the considered system to a given situation of the environment with respect to the goal or *purpose*, and is both caused by the needs and constrained by the limits of its physical resources. The attribute adequate means that the activated behavior contributes something to reach the goal while considering all circumstances in the right way. This is an expression of competence.

Behavior obviously subsumes conceptually at least three constituents: an *afferent* (sensoric) and an *efferent* (actoric) interaction of the system with its *environment*, and any *kind of relating the one* with the other (e. g. by mapping, processing, etc.). Because of the mutual support and mutual evaluation of both kinds of interaction a useful system theoretical model is cyclic arrangement instead of the usual linear input/output model. If the frame of system consideration is given by the whole biological system (a fly, a tree or a man), we call this arrangement the *perception-action cycle* (PAC). Of course, the mentioned conceptual separation does not exist for real systems. Therefore, having artificial systems follow the behavior-based system (BBS) design of the metaphor of biological competence is a hard task. Koenderink [6] called *perception* and *action* two sides of the same entity.

An important difference between behavior and PAC thus becomes obvious. While behavior is the observable manifestation of competence, the perception-action cycle is the frame of dynamics the system has to organize on the basis of the gained competence. This distinction will become important in Section 10.2.4 with respect to the designers' task.

Competence is the ability of the system to organize the PAC in such a way that the inner degrees of freedom are ordered as a result of the perceived order of the environment and the gained order of actions with respect to the environment. A competent system has the capability to separate relevant from irrelevant percepts and can manage its task in the total complexity of concrete environmental situations. This is what is called *situatedness of behavior*. With the term *corporeality of behavior* the dependence of the competence from the specific needs and limitations of the physical resources should be named. This also includes determining the influence of the environment on the adaptation of the corporeality during phylogenesis. Both features of behavior are in sharp contrast to the computational theory of mind. Gaining competence is a double-track process. On the one hand, learning from experience is very important. But learning from scratch the entirety of competence will be too complex with respect to the individual lifespan. Therefore, learning has to be biased by either acquired knowledge from phylogenesis or from the learner.

In the language of *nonlinear dynamic systems*, behavior has the properties of an *attractor*. This means that it is *robust* with respect to small distortions and *adaptable* with respect to larger ones. Because of its strong relation to purpose, behavior has the additional properties of usefulness, efficiency, effectiveness, and suitability. From this it follows that an actual system needs a plethora of different behaviors and some kind of control to use the right one. This control may be, for example, event-triggering or purposive decision-making using either intuition or reasoning, respectively. Behaviors can be used to conceptualize a system as a conglomerate of cooperating and competing perception-action cycles. The loss of one behavior does not result in the breakdown of the whole system. This is the third important property from the engineer's point of view, and is called *stability*.

10.2.3 From natural to artificial behavior-based systems

A general consequence of the sketched metaphor of biological competence seems to be that each species and even each individual needs its own architecture of behaviors. This is true to a great extent. This consequence results from different physical resources and different situative embeddings of the systems. For instance, the vision task is quite different for an ant, a frog, or a chimpanzee. There is no unique general theory of biological competences as regards vision; this is in contrast to the postulation by Marr [5]. This conclusion may result either in resignation or in dedicated design of highly specialized systems. Because of the limitations of knowledge-based approaches the last way is the one that contemporary engineers are adopting for industrial applications. However, there is no reason to follow it in the longer term.

Indeed, biological competence cannot be formulated as laws of nature in the same way as has been the case for laws of gravity. We should bear in mind that competence takes these natural laws into account. There must be several more or less general principles that biological systems use towards learning or applying competence. What at a first glance is recognized as a set of heuristics will be ordered with respect to its common roots. While as a matter of basic research these principles have to be identified, over the long term it is hoped that we will be able to follow the engineering approach. This need of identification requires observation of natural systems using, for example, ethology, psychophysics and neural sciences, and the redefining of already known classes of problems.

From a system theoretical point of view, behavior-based systems are *autonomous systems*. They are *open systems* with respect to their environment, *nonlinear systems* with respect to the situation dependence of their responses to sensory stimuli, and *dynamic systems* with respect to their ability to activate different behaviors. In traditional engineering, the aggregation of a system with its environment would result in a *closed system*. As the mutual influences between the environment and the original system are finite ones with respect to a certain behavior, this kind of conception seems to make sense. We call this approach the *top-down design* principle. The observation of an interesting behavior leads to modeling and finally to the construction and/or implementation of the desired function. However, this mechanistic view of behavior has to be assigned to the computational theory of mind with all the mentioned problems. Instead, we indeed have to consider the system as an open one.

10.2.4 Bottom-up approach of design

In order to make sure that the behavior will gain attractor properties, the designer has to be concerned that the system can find the attractor basin by self-organization. The system has to sample stochastically the space of its relations to the environment, thus to find out the regularities of perception and action and to respond to these by self-tuning of its parameters. This is *learning by experience* and the design principle is bottom-up directed.

The *bottom-up design* of behavior-based systems requires the designer to invert the description perspective with respect to behavior to the synthesizing perspective with respect to the perception-action cycle. Instead of programming a behavior, *learning* of competence has to be organized. The resulting behavior can act in the world instead of merely a model of it. Yet correctness that can be proved will be replaced by observable success.

A pure bottom-up strategy of design will make no sense. We know that natural systems extensively use knowledge or *biasing* as genetic information, instincts, or in some other construct. Both learning paradigms and the necessary quantity and quality of biasing are actual fields of research in neural computation. For instance, *reinforcement learning* of sonar-based navigation of a mobile robot can be accelerated by a factor of a thousand by partitioning the whole task into situation dependent subtasks and training them separately. Another approach with comparable effect will be gained by delegation of some basic behaviors to programmed instincts. The division between *preattentive visual tasks* and *attentive visual tasks* or the preprocessing of visual data using operations comparable to simple cells in the primary visual cortex plays a comparable role in the learning of visual recognition.

Thus, to follow the behavior-based paradigm of system design requires finding good learning strategies not only from *ontogenesis* of biological systems but also from *phylogenesis*. The latter one can be considered as responsible for guaranteeing good beginning conditions for individual learning of competence. Without the need of explicit representation such kind of knowledge should be used by the designer of artificial behavior-based systems.

The bottom-up design principle is not only related to the learning of regularities from seemingly unrelated phenomena of the environment. It should also be interpreted with respect to the mutual dependence of the tasks. *Oculomotor behaviors* like visual *attention*, *foveation* and *tracking* are basic ones with respect to *visual behaviors* as *estimation of spatial depth* and this again may be basic with respect to *visual navigation*. Because of the forementioned reduction of learning complexity by partitioning a certain task, a bootstrap strategy of design will be the preferential one. Such stepwise extension of competences will not result in quantitatively linear *scaling of capabilities*. But from the interacting and competing competences qualitatively new competences can *emerge*, which are more than the union of the basic ones. From this nonlinear scaling of capabilities it follows that the design of behavior-based systems cannot be a completely determined process.

10.2.5 Grounding of meaning of equivalence classes

In KBS design we know the problem of the missing *grounding of meaning of categories*. In BBS design this problem becomes obsolete as the so-called *signal-symbol gap* vanishes. *Categories* belong to the metaphor of the computational theory of mind. They are only necessary with respect to modeling, reasoning, and using language for communication. Most biological systems organize their life on a pre-categorical level using *equivalence classes* with respect to the environment and the self. Equivalence classes are those entities that stand for the gained inner order as a result of learning. Their meaning is therefore grounded in the experience made while trying to follow the purpose. In contrast to classical definitions of equivalence classes they have to be multiply supported. This is an immediate consequence of the intertwined structure of the perception-action cycle. If we distinguish between equivalence classes for perception and action, we mean that they are useful with respect to these tasks [7]. This does not mean that they are trained either by perception or action in separation. Instead, their development out of random events is always rooted in both multiple sensory clues using *actoric verification*. Actoric verification means using actions to get useful sensory data. This mutual support of equivalence classes by perception and action has to be considered in the design of special systems for computer vision or visual robotics [8]. A visual navigating system does not need to represent the equivalence classes of all objects the system may meet on its way but useful relations to any objects as equivalence classes of sensorimotorics. We will come back to this point in Sections 10.2.6 and 10.3.2. The grounding of meaning in physical experience corresponds to the construction or selection of the *semantic aspect* of *information*. The *pragmatic aspect* of information is strongly related to the purpose of activity. The *syntactic aspect* of the information is based on sensorics and perception. Cutting of the perception-action cycle into perception and action as separated activities thus will result in the loss of one or another aspect of the forementioned trinity.

10.2.6 General principles of behavior-based system design

We have identified so far the two most important principles of BBS design: the knowledge biased bottom-up approach and the multiple support of equivalence classes. Here we will proceed to identify some additional general principles of design from which we will draw some conclusions for the adequate algebraic embedding of the PAC.

Minimal effort and *purposive opportunism* are two additionally important features of behavior. These result from the limited resources of the system, the inherent real-time problem of the PAC, and the situative embedding of purposively driven behavior. From these two features we can formulate the following required *properties of behavior*:

1. **fast**: with respect to the time scale of the PAC in relation to that of the events in the environment;
2. **flexible**: with respect to changes of the environment in relation to the purpose;
3. **complete**: with respect to the minimal effort to be purposive;
4. **unambiguous**: with respect to the internalized purpose; and
5. **selective**: with respect to purposive activation of alternative behaviors.

These need result in challenging functional architectures of behavior-based systems. The realization of useful strategies has to complete more traditional schemes as pattern recognition or control of motion.

Thus, the cyclic scheme of the PAC will project also to the internal processes of purposive information control. This has been shown by the author [9] with respect to the design of a visual architecture.

As a further consequence there is a need of equivalence classes of *graduated completeness*. It may be necessary and sufficient to use only rough versions of equivalence classes for motion or recognition. This requires their definition as *approximations* and their use in a scheme of *progression*. Ambiguity need not be resolved at all steps but has to vanish at the very end of processing. Besides, if there are other useful ways or hints that result in unambiguous behaviors the importance of a single one will be relative. Simultaneously the *principle of consensus* [10] should be used. Such representations of equivalence classes have to be sliced within the abstraction levels of equivalence classes while traditional schemes only support slicing between abstraction levels. The basis functions of steerable filter design [11] in early vision represent such a useful principle. Another example would be the eigenvalue decomposition of patterns [12] and motion [13].

This quantitative scaling of completeness has to be supplemented by a scheme of *qualitative completeness* of equivalence classes. The most intuitive example for this is given by the task of visual navigation [14]. In this respect only in certain situations will it be necessary to compute a Euclidean reconstruction of 3-D objects. In standard situations there is no need to recognize objects. Rather, holding some dynamically stable relations to conjectured obstacles will be the dominant task. For this task of minimal effort a so-called scaled relative nearness [15] may be sufficient, which represents depth order as qualitative measure for a gaze fixating and moving uncalibrated stereo camera [16].

Here the oculomotor behavior of *gaze fixation* is essential for getting depth order. Looking at a point at infinity does not supply such a clue. The general frame of switching between *metric*, *affine* or *projective representations of space* was proposed by Faugeras [17].

Generally speaking, in behavior-based systems recognition will dominate reconstruction. But in highly trained behaviors even recognition can be omitted. Instead, blind execution of actions will be supported by the multiple-based equivalence classes. This means that, for example, visually tuned motor actions will not need visual feedback in the trained state. In that state the sequence of motor signals permits the execution of motions in a well-known environment using the principle of trust with respect to the stability of the conditions.

There are other diverse general principles that are used by natural systems and should be used for BBS design but must be omitted here due to limited space.

10.3 Algebraic frames of higher-order entities

10.3.1 Perception, action and geometry

In this section we want to examine some essential features of perception and action to conclude hereof a mathematical framework for their embedding in the next sections. In Section 10.2.5 we learned from the tight coupling of both behavioral entities the need for common support of equivalence classes. Here we want to emphasize the *equivalence of perception and action* with respect to their aims from a geometrical point of view. In this respect we implicitly mean visual perception. Mastering geometry indeed is an essential part of both perception and action, see [18] or the following citations:

> Koenderink [19]: "The brain is a geometry engine."
> Pellionisz and Llinas [20]: "The brain is for (geometrical) representation of the external world."
> von der Malsburg [21]: "The capabilities of our visual system can only be explained on the basis of its power of creating representations for objects that somehow are isomorphic to their real structure (whatever this is)."

The *representation problem*, as indicated by the citations, will be of central importance in this section. A system, embedded in Euclidean space and time, is seeking to perceive structures or patterns of high regularity, and is seeking to draw such patterns by egomotion. The equivalence classes of a competent system correspond to smooth manifolds of low local *intrinsic dimension* [12]. While some patterns of motion are globally 1-D structures (as gestures), others (such as facial expressions) are globally 2-D structures but often can be locally interpreted as 1-D structures.

Both perceptual and motor generated patterns are of high *symmetry*. In at least one dimension they represent a certain conception of *invariance* as long as no event (in case of action) or no boundary (in case of perception) causes the need to switch to another principle of invariance. On the other hand, completely uncorrelated patterns have an infinite or at least large intrinsic dimension. The smooth manifolds of the trained state emerge from initially uncorrelated data as a result of learning.

We introduced the equivalence classes as manifolds to correspond with the bottom-up approach of statistical analysis of data from $\mathbb{R}^N$, as is the usual way of neural computation. Within such an approach the equivalence classes can be interpreted as (curved) subspaces $\mathbb{R}^M$, $M < N$, which constitute point aggregations of $\mathbb{R}^N$. This approach is useful from a computational point of view and indeed the paradigm of artificial neural nets has been proven to learn such manifolds in a topology preserving manner (see, e.g., Bruske and Sommer [22]).

Yet such a *distributed representation* of equivalence classes is inefficient with respect to operations, as comparison, decision finding, or planning. Therefore, equivalence classes have to be represented additionally as *compact geometric entities* on a higher level of abstraction. For instance, searching for a chair necessitates representing a chair itself. Although we do not know the best way of representation yet, in the following sections we will develop an approach to represent so-called higher order geometric entities as directed numbers of an algebra. In the language of neural computation these may be understood as the activation of a grandmother neuron. In Section 10.4.2 we will discuss such representations in an algebraically extended multilayer perceptron.

10.3.2 Behavioral modulation of geometric percepts

Animals have to care for globally spatiotemporal phenomena in their environment. They use as behavior-based systems both gaze control and oculomotor behaviors to reduce the complexity of both vision and action [7, 8]. In this context, vision and action are related in two complementary senses:

1. **vision for action** : similar visual patterns cause similar motor actions; and
2. **action for vision** : similar motor actions cause similar visual patterns.

This implicitly expresses the need to understand vision as a serial process in the same manner as action. Global percepts result from a quasi-continuous sequence of local percepts. These are the result of *attentive vision*, which necessarily uses fixation as an oculomotor behavior. Fixation isolates a point in space whose meaning with respect to its environment will be evaluated [23]. Although signal theory supplies a rich toolbox, a complete local structure representation can only be provided by the algebraic embedding of the task, which will be presented in Section 10.4.1.

Tracking as another *gaze control* behavior is useful to transform temporal patterns of moving objects into a stationary state. Conversely a dynamic perception of nonmoving objects results from egomotion of the head. The control of the eyes' gaze direction, especially with respect to the intrinsic time scale of the PAC, is a demanding task if the head and body are able to move. Kinematics using *higher-order entities*, see Section 10.3.9, will also be important with respect to fast navigation in the presence of obstacles. It will be necessary to plan egomotion on the basis of visual percepts while considering the body as a constraint. That problem of trajectory planning is well known as the "piano mover problem." This means that not only distances between points but also between any entities such as *points*, *lines* or *planes* are of importance. In this respect, an alternative to the forementioned sequential process of attentive vision will gain importance. Indeed, attentive vision is accompanied by a fast parallel recognition scheme called preattentive vision. In *preattentive vision* the percepts of internalized concepts of certain regular patterns (also higher-order entities) pop out and cause a shift of attention. All visual percepts origin from retinal sensory stimuli, which are projections from the 3-D environment and have to be interpreted accordingly. The proposed algebraic frame will turn out to be intuitively useful with respect to projective reconstruction and recognition. As outlined in Section 10.2.6, oculomotor behaviors are used for fast switching between the projective, affine or similarity group of the stratified Euclidean space. In section Section 10.3.9 we will show how to support these group actions and those of rigid movements just by switching the signature of the algebra.

10.3.3 Roots of geometric algebra

We have argued in Section 10.3.1 from the position of minimal effort that to hold a PAC running one needs higher-order geometric entities. Besides, we indicated the problems of locally recognizing multidimensional signals, and the problems of locally organizing chained movements in space, respectively. All three problems have a common root in the limited representation capability of linear vector spaces, respectively, of linear *vector algebra*. Although there are available in engineering and physics more powerful algebraic languages, such as Ricci's *tensor algebra* [24] or Grassmann's *exterior algebra* [25], their use is still limited in the disciplines contributing to the design of artificial PAC (see for instance [26, 27]) for the application of tensor calculus for multi-dimensional image interpretation, or [28] with respect to applications of exterior calculus to computer vision. We decided on *geometric algebra* or GA [2, 29] because of its universal nature and its intuitive geometric interpretation. The main bulk of this algebra rests on *Clifford algebra* [1] (see [30] for a modern introduction). Yet it covers also vector algebra, tensor algebra, *spinor algebra*, and *Lie algebra*. In a natural way it includes the algebra of *complex numbers* [31] and that of *quaternionic numbers* (or *quaternions*) [32]. The useful aspect of its universality is its capability of supporting quantitative, qualitative and operational aspects in a homogeneous frame. Therefore, all disciplines contributing to PAC, for example, *computer vision*, *multidimensional signal theory*, *robotics*, and *neural computing* can be treated in one algebraic language. Of course, the existing isomorphism between the algebras enables the decision for one or another. But often such decisions are costly. Such costs include redundancy, coordinate dependence, nonlinearity, or incompatible system architecture. The GA is most commonly used in physics, although some related conceptions as screw theory of kinematics can also be found in robotics [33, 34]. The first attempts on the study of behavior control were published by Tweed et al. [35] and Hestenes [36, 37].

10.3.4 From vector spaces to multivector spaces

A *geometric algebra* G_n is a linear space of dimension 2^n, which results from a *vector space* V_n of dimension n by endowing it with a new type of product called geometric product. The entities $\mathbf{A}$ of G_n are called *multivectors*

$$\mathbf{A} = \sum_{k=0}^{n} \mathbf{A}_k \tag{10.1}$$

where $\mathbf{A}_k = \langle \mathbf{A} \rangle_k$, $k \leq n$ are *homogeneous multivectors* of *grade* k or simply *k-vectors*. The GA is a linear and associative algebra with respect to addition and multiplication, endowed with additive and multiplicative identities, algebraically closed and commutative with respect to addition but not with respect to multiplication [2]. Instead, for any $\mathbf{A}, \mathbf{B} \in G_n$ the product $\mathbf{AB}$ is rather complicated related to $\mathbf{BA}$. The product of any two homogeneous multivectors $\mathbf{A}_r$, $\mathbf{B}_s$, $r + s \leq n$, results in an *inhomogeneous multivector* $\boldsymbol{C} \in G_n$ consisting of a set of different grade homogeneous multivectors.

$$\mathbf{C} = \mathbf{A}_r \mathbf{B}_s = \langle \mathbf{A}_r \mathbf{B}_s \rangle_{|r-s|} + \langle \mathbf{A}_r \mathbf{B}_s \rangle_{|r-s|+2} + \cdots + \langle \mathbf{A}_r \mathbf{B}_s \rangle_{r+s} \tag{10.2}$$

This equation offers the desired property of the algebra that the product of two entities results in a set of other *entities* of different grade. This can be best understood if we consider the lowest grades k-vectors. They correspond to the following nomenclature: $k = 0$: *scalars*, $k = 1$: *vectors*, $k = 2$: *bivectors*, ...

We start with vectors $\boldsymbol{a}, \boldsymbol{b} \in V_n$. They will remain the same in G_n because for any vector $\boldsymbol{a}$ we will have $\mathbf{A}_1 = \langle \mathbf{A} \rangle_1 = \boldsymbol{a}$. Therefore, we will symbolize also in GA a 1-vector as a vector of V_n. Vectors follow the *geometric product*

$$\mathbf{C} = \boldsymbol{ab} = \boldsymbol{a} \cdot \boldsymbol{b} + \boldsymbol{a} \wedge \boldsymbol{b} \tag{10.3}$$

with the *inner product*

$$\mathbf{C}_0 = \langle \boldsymbol{ab} \rangle_0 = \boldsymbol{a} \cdot \boldsymbol{b} = \frac{1}{2}(\boldsymbol{ab} + \boldsymbol{ba}) \tag{10.4}$$

and the *outer product*

$$\mathbf{C}_2 = \langle \boldsymbol{ab} \rangle_2 = \boldsymbol{a} \wedge \boldsymbol{b} = \frac{1}{2}(\boldsymbol{ab} - \boldsymbol{ba}) \tag{10.5}$$

so that $\mathbf{C}$ is a mixture of a scalar and a bivector. Therefore, in Eq. (10.2) the term $\langle \mathbf{A}_r \mathbf{B}_s \rangle_{|r-s|} = \mathbf{A}_r \cdot \mathbf{B}_s$ stands for a pure inner product and the term $\langle \mathbf{A}_r B_s \rangle_{r+s} = \mathbf{A}_r \wedge \mathbf{B}_s$ stands for a pure outer product component. If in Eq. (10.3) $\boldsymbol{a} \wedge \boldsymbol{b} = 0$, then

$$\boldsymbol{ab} = \boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{b} \cdot \boldsymbol{a} = \boldsymbol{ba} \tag{10.6}$$

and, otherwise if $\boldsymbol{a} \cdot \boldsymbol{b} = 0$, then

$$\boldsymbol{ab} = \boldsymbol{a} \wedge \boldsymbol{b} = -\boldsymbol{b} \wedge \boldsymbol{a} = -\boldsymbol{ba} \tag{10.7}$$

Hence, from Eq. (10.6) and Eq. (10.7) follows that *collinearity* of vectors results in commutativity and *orthogonality* of vectors results in anti-commutativity of their geometric product. These properties are also valid for all multivectors.

The expansion of a GA G_n for a given vector space V_n offers a rich hierarchy of structures, which are related to the subspace conception of the algebra. From the vector space V_n that is spanned by n linear independent vectors $\boldsymbol{a}_i = a_i \boldsymbol{e}_i$, $\boldsymbol{e}_i$ unit basis vector, results one unique maximum grade multivector $\mathbf{A}_n = \langle \mathbf{A} \rangle_n$ that factorizes according

$$\mathbf{A}_n = \prod_{k=1}^{n} \boldsymbol{a}_k \tag{10.8}$$

On each other grade k any k linear independent vectors $\boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_k}$ will factorize a special homogeneous k-vector, which is called *k-blade* $\mathbf{B}_k$ thus

$$\boldsymbol{B}_k = \prod_{j=1}^{k} \boldsymbol{a}_j \tag{10.9}$$

There are $l = \binom{n}{k}$ linear independent k-blades $\mathbf{B}_{k_1}, \ldots, \mathbf{B}_{k_l}$ that span the linear subspace G_k of all k-vectors $\mathbf{A}_k \in G_n$, so that all $\mathbf{A}_k$, $k = 1, \ldots, n$, with

$$\mathbf{A}_k = \sum_{j=1}^{l} \mathbf{B}_{k_j} \tag{10.10}$$

finally complete with any $\mathbf{A}_0$ the inhomogeneous multivector $\mathbf{A}$ of the algebra $G_n(\mathbf{A})$, following Eq. (10.1). Hence, each k-blade $\mathbf{B}_k$ corresponds to a vector subspace $V_k = \langle G_n(\mathbf{B}_k) \rangle_1$, consisting of all vectors $\boldsymbol{a} \in V_n$ that meet the collinearity condition

$$\boldsymbol{a} \wedge \mathbf{B}_k = 0 \tag{10.11}$$

From this it follows that, as with the vector $\boldsymbol{a}$ and the k-blades $\mathbf{B}_k$, respectively, the vector subspaces $V_k = \langle G_n(\mathbf{B}_k)\rangle_1$ have a unique direction. Therefore, k-vectors are also called *directed numbers*. The direction of $\mathbf{B}_k$ is a *unit k-blade*

$$\mathbf{I}_k \;=\; \boldsymbol{e}_{i_1} \wedge \boldsymbol{e}_{i_2} \wedge \cdots \wedge \boldsymbol{e}_{i_k} \tag{10.12}$$

so that $\mathbf{B}_k = \lambda_k \mathbf{I}_k$, $\lambda_k \in \mathbb{R}$, respectively $\lambda_k \in \mathbf{A}_0$ and $\mathbf{e}_{i_j} \in \langle G_n(\mathbf{I}_k)\rangle_1$. Of course, the same factorization as Eq. (10.9) is valid for $k = n$. The only n-blade $P \equiv \mathbf{B}_n$ is called *pseudoscalar*. Its direction is given by the *unit pseudoscalar I* that squares to

$$I^2 \;=\; \pm 1 \tag{10.13}$$

Finally, a remarkable property of any G_n should be considered. Each subset of all even grade multivectors constitutes an even *subalgebra* G_n^+ of G_n so that the linear space spanned by G_n can be expressed as the sum of two other linear spaces

$$G_n \;=\; G_n^- \;+\; G_n^+ \tag{10.14}$$

Because G_n^- is not closed with respect to multiplication, it does not represent a subalgebra.

10.3.5 Properties of multivector spaces

In this subsection we want to present some basic properties of a GA as a representation frame that demonstrate both its superiority in comparison to vector algebra and its compatibility with some other algebraic extensions.

A multivector $\mathbf{A}$ has not only a direction but also a *magnitude* $|\mathbf{A}|$, defined by

$$|\mathbf{A}|^2 = \langle \tilde{\mathbf{A}}\mathbf{A}\rangle_0 \geq 0 \tag{10.15}$$

where $|\boldsymbol{A}| = 0$ only in case of $\boldsymbol{A} = 0$ and $\tilde{\mathbf{A}}$ is the *reversed* version of $\mathbf{A}$, which results from the reversed ordering of the vectorial factors of the blades. Because these behave as

$$|\mathbf{B}_k|^2 = |\boldsymbol{a}_1 \cdots \boldsymbol{a}_k|^2 = |\boldsymbol{a}_1|^2 \cdots |\boldsymbol{a}_k|^2 \geq 0 \tag{10.16}$$

we get

$$|\mathbf{A}|^2 = |\langle\mathbf{A}\rangle_0|^2 + |\langle\mathbf{A}\rangle_1|^2 + \cdots + |\langle\mathbf{A}\rangle_n|^2 \tag{10.17}$$

The magnitude has the properties of a norm of $G_n(\mathbf{A})$ yet has to be specified, if necessary, by endowing the algebra with a signature, see for example, Section 10.3.8. A geometrical interpretation can be associated with a k-blade. It corresponds with a directed k-dimensional hypervolume of $G_k(\mathbf{A})$, whose magnitude is given by Eq. (10.15) and whose k-dimensional direction is given by Eq. (10.12). The inner product of multivectors expresses a degree of similarity, while the outer product expresses a degree of dissimilarity. The great power of the geometric product results from simultaneously holding both. In other words, the closeness of the GA with respect to the geometric product results from the properties of the pseudoscalar P that uniquely determines the properties of the vector space V_n. Because the unit pseudoscalar I factorizes with respect to a chosen unit k-blade $\mathbf{I}_k$ such that

$$\mathbf{I}_k \mathbf{I}_{n-k} = I \tag{10.18}$$

any G_n relates two mutual orthogonal vector subspaces $V_k = G_1(\mathbf{I}_k)$ and $V_{n-k} = G_1(\mathbf{I}_{n-k})$. This results in the definition of a *dual k-vector*

$$\mathbf{A}^*_{n-k} = \mathbf{A}_k I^{-1} = \mathbf{A}_k \cdot I^{-1} \tag{10.19}$$

because $II^{-1} = 1$. Thus, the *duality operation* [38] changes the multivector basis and enables us to consider any entity from its dual aspect. For given $\mathbf{A}_r$ and $\mathbf{B}_s$ the duality of their inner and outer products becomes obvious

$$\mathbf{A}_r \cdot \mathbf{B}^*_s = (\mathbf{A}_r \wedge \mathbf{B}_s)^* \tag{10.20}$$

Because in case of $r + s = n$ it follows

$$P = \lambda I = \mathbf{A}_r \wedge \mathbf{B}_s \tag{10.21}$$

also the scalar λ can be interpreted as the dual of the pseudoscalar P,

$$\lambda = [P] = PI^{-1} = (\mathbf{A}_r \wedge \mathbf{B}_s)I^{-1} = \mathbf{A}_r \cdot \mathbf{B}^*_s \tag{10.22}$$

In Grassmann algebra [25, 28] the extra operator *bracket* [.], is introduced to define exteriors (or extensors) as the subspaces of the algebra. While exteriors are in one-to-one relation to k-vectors, and while Eq. (10.9) corresponds to Grassmann's *progressive product*, the bracket replaces the missing inner product in Grassmann algebra (see Eq. (10.22)) but is not an operation of the algebra. Furthermore, because

$$[P] = [\boldsymbol{a}_1 \boldsymbol{a}_2 ... \boldsymbol{a}_n] = \det(V_n) \tag{10.23}$$

the scalar λ may be considered as a definition of the *determinant* of the vector coordinates $\{a_i,\ i = 1, ..., n\}$ in V_n. In contrast to this coordinate dependence of the determinant, λ may be computed completely coordinate independent from multivectors as higher-order entities.

In Section 10.3.8 we will show that k-vectors correspond to *tensors of rank k*. If the values of tensors are inhomogeneous multivectors, conventional tensor analysis can be considerably enriched within GA (see e.g., Hestenes and Sobczyk [2]).

10.3.6 Functions, linear transformations and mappings

The aim of this subsection is to introduce some fundamental aspects of GA that are of immediate importance in the frame of PAC design. The first one is the *exponential function* of a multivector $\mathbf{A} \in G_n$ as a mapping $\exp(\mathbf{A}) : G_n \longrightarrow G_n$, algebraically defined by

$$\exp(\mathbf{A}) \equiv \sum_{i=0}^{\infty} \frac{\mathbf{A}^i}{i!} \tag{10.24}$$

The known relation

$$\exp(\mathbf{A})\exp(\mathbf{B}) = \exp(\mathbf{A}+\mathbf{B}) = \exp(\mathbf{C}) \tag{10.25}$$

is only valid in case of collinearity of $\mathbf{A}$ and $\mathbf{B}$, else $\mathbf{C} \neq \mathbf{A} + \mathbf{B}$. The invalidity of Eq. (10.25) in the algebraic embedding of the N-D Fourier transform will be surmounted in Section 10.4.1. Equation (10.24) can be expressed by the even and odd parts of the exponential series

$$\exp(\mathbf{A}) = \cosh(\mathbf{A}) + \sinh(\mathbf{A}) \tag{10.26}$$

Alternatively, if I is a unit pseudoscalar with $I^2 = -1$ and, if $\mathbf{A}$ is another multivector which is collinear to I, then

$$\exp(\mathbf{IA}) = \cos(\mathbf{A}) + I\ \sin(\mathbf{A}) \tag{10.27}$$

Because of Eq. (10.19) or, equivalently $\mathbf{A}_k I = \mathbf{A}^*_{n-k}$, Eq. (10.27) can be read with respect to a dual multivector exponential.

a) $k = 3 : \mathbf{A}_3 = \mathbf{A}_0 I,\ \mathbf{A}_0 \in \langle G_3(\mathbf{A})\rangle_0$
$\quad\exp(\mathbf{A}_3) = \cos(\mathbf{A}_0) + I\sin(\mathbf{A}_0)$
b) $k = 2 : \mathbf{A}_2 = \mathbf{A}_1 I,\ \mathbf{A}_1 = \mathbf{A}_0\boldsymbol{e}, \mathbf{A}_0 = |\mathbf{A}_1|$
$\quad\exp(\mathbf{A}_2) = \cos(\mathbf{A}_1) + I\sin(\mathbf{A}_1)$
$\quad\exp(\boldsymbol{A}_2) = \cos(\boldsymbol{A}_0) + I\boldsymbol{e}\sin(\boldsymbol{A}_0)$
c) $k = 1 : \mathbf{A}_1 = \mathbf{A}_2 I,\ \mathbf{A}_1 = \mathbf{A}_0\boldsymbol{e},\ \mathbf{A}_0 = |\mathbf{A}_1|$
$\quad\exp(\mathbf{A}_1) = \cosh(\mathbf{A}_0) + \boldsymbol{e}\sinh(\mathbf{A}_0)$
d) $k = 0 : \mathbf{A}_0 = \mathbf{A}_3 I$
$\quad\exp(\mathbf{A}_0) = \cosh(\mathbf{A}_0) + \sinh(\mathbf{A}_0)$

The second point to be considered is the behavior of *linear transformations* $\boldsymbol{L}$ on a vector space V_n with respect to the algebraic embedding into G_n. The extended version $\mathcal{L}$ of the linear transformation $\boldsymbol{L}$ is yet a linear one in G_n and distributive with respect to the outer product.

$$\mathcal{L}(\mathbf{A} \wedge \mathbf{B}) = (\mathcal{L}\mathbf{A}) \wedge (\mathcal{L}\mathbf{B}) \tag{10.28}$$

Because of the preserving property with respect to the outer product, this behavior of a linear transformation is called *outermorphism* [39].

The inner product will not be generally preserved. For any k-blade $\mathbf{B}_k$ we have

$$\mathcal{L}\mathbf{B}_k = (\mathcal{L}\boldsymbol{a}_1) \wedge (\mathcal{L}\boldsymbol{a}_2) \wedge \ldots \wedge (\mathcal{L}\boldsymbol{a}_k) \tag{10.29}$$

and for any multivector $\mathbf{A}$ it will be grade-preserving, thus

$$\mathcal{L}(\langle \mathbf{A} \rangle_k) = \langle \mathcal{L}\mathbf{A} \rangle_k \tag{10.30}$$

Now, the mapping of oriented multivector spaces of different maximum grade will be outlined (which is called *projective split* by Hestenes [39]). This is much more than the subspace philosophy of vector algebra. Although it could be successfully applied with respect to *projective geometry* and *kinematics* (see Section 10.3.9), its great potential has not been widely recognized yet. Given a vector space V_n and another V_{n+1} and their respective geometric algebras G_n and G_{n+1}, then any vector $\boldsymbol{X} \in \langle G_{n+1} \rangle_1$ can be related to a corresponding vector $\boldsymbol{x} \in V_n$ with respect to a further unit vector $\boldsymbol{e} \in V_{n+1}, \boldsymbol{e}^2 = 1$, by

$$\mathbf{X}\boldsymbol{e} = \mathbf{X} \cdot \boldsymbol{e}(1 + \boldsymbol{x}) \tag{10.31}$$

or

$$\boldsymbol{x} = \frac{\mathbf{X} \wedge \boldsymbol{e}}{\mathbf{X} \cdot \boldsymbol{e}} \tag{10.32}$$

Thus, the introduction of the reference vector $\boldsymbol{e}$ resembles the representation of $\boldsymbol{x}$ by *homogeneous coordinates* in G_{n+1}, and it includes this important methodology. Yet it goes beyond and can be used to linearize nonlinear transformations and to relate projective, affine, and metric geometry in a consistent manner. Besides, Hestenes [39] introduced the so-called conformal split to relate G_n and G_{n+2} via a unit 2-blade.

10.3.7 Qualitative operations with multivectors

So far we have interpreted the inner and outer products as *quantitative operations*. We also have seen that both the duality operation and the projective split are *qualitative operations*. While the first one realizes a mapping of multivector subspaces with respect to a complementary basis, the second one relates multivector spaces which differ in dimension by one.

With the interesting properties of raising and lowering the grade of multivectors, the outer and inner products also possess qualitative aspects. This is the reason why in geometric algebra higher-order (geometric or kinematic) entities can be simply constructed and their relations can be analyzed, in sharp contrast to vector algebra. Because we so far do not need any metrics of the algebra, the set algebraic aspects of vector spaces should be related to those of the GA. The union ($\cup$) and intersection ($\cap$) of blades, respectively, their vector subspaces (see Eq. (10.11)), goes beyond their meaning in vector algebra. An equation such as

$$\langle G_n(\mathbf{A} \wedge \mathbf{B})\rangle_1 = \langle G_n(\mathbf{A})\rangle_1 \cup \langle G_n(\mathbf{B})\rangle_1 \tag{10.33}$$

where $\mathbf{A}, \mathbf{B}$ are blades, has to be interpreted as an oriented union of oriented vector subspaces [2]. With respect to the basic operations of the *incidence algebra*, that means the meet and the join operations, the extended view of GA will be best demonstrated. The *join* of two blades of grade r and s

$$\mathbf{C} = \mathbf{A} \bigwedge \mathbf{B} \tag{10.34}$$

is their common dividend of lowest grade. If $\mathbf{A}$ and $\mathbf{B}$ are linear independent blades, the join and the outer product are operations of the same effect, thus $\mathbf{C}$ is of grade $r+s$. Otherwise, if $\mathbf{A} \wedge \mathbf{B} = 0$, the join represents the spanned subspace. The *meet*, on the other hand, $\mathbf{C} = \mathbf{A} \bigvee \mathbf{B}$, is indirectly defined with respect to the join of both blades by

$$\mathbf{C}^* = (\mathbf{A} \bigvee \mathbf{B})^* = \mathbf{A}^* \bigwedge \mathbf{B}^* \tag{10.35}$$

It represents the common factor of $\mathbf{A}$ and $\mathbf{B}$ with greatest grade. In case of $r + s = n$ the meet is of grade $|r - s|$ and will be expressed by

$$\mathbf{C} = \boldsymbol{A}^* \cdot \boldsymbol{B} \tag{10.36}$$

Because the Grassmann algebra is based on both the *progressive product* and the *regressive product* of the incidence algebra, it does not surprise that both GA and Grassmann algebra are useful for *projective geometry*, although the projective split and the duality principle of GA enable a more intuitive geometric view and help enormously in algebraic manipulations.

10.3.8 Geometric algebra of the Euclidean space

In this subsection we will consider the GAs of the Euclidean space E_3 and of the Euclidean plane E_2.

From now on we have to consider metric GAs because their vector spaces are metrical ones. A vector space V_n should be endowed with a *signature* (p, q, r) with $p + q + r = n$, so that $G_{p,q,r}$ indicates the number of basis vectors $\boldsymbol{e}_i$, $i = 1, \ldots, n$ that square in the following way

$$\boldsymbol{e}_{i_j}^2 = \begin{cases} 1 & \text{if} \quad j \in \{1, \ldots, p\} \\ -1 & \text{if} \quad j \in \{p+1, \ldots, p+q\} \\ 0 & \text{if} \quad j \in \{p+q+1, \ldots, p+q+r\} \end{cases} \tag{10.37}$$

We call an algebra with $r \neq 0$ a *degenerated* algebra because those basis vectors with $\boldsymbol{e}_{i_j}^2 = 0$ do not contribute to a *scalar product* of the vector space. This results in a constraint of viewing the vector space on a submanifold. In case of $r = 0$ we omit this component and will write $G_{p,q}$. Furthermore, if additionally $q = 0$, the vector space will have a *Euclidean metric*.

For a given dimension n of a vector space V_n there are GAs that are algebraically isomorphic (but not geometrically equivalent), for example, $G_{1,1}$ and $G_{2,0}$, while others are not isomorphic (as $G_{0,2}$ to the other two). In any case, the partitioning $V_n = V_p \cup V_q \cup V_r$ will result in the more or less useful properties of the corresponding algebra $G_{p,q,r}$. We will consider a constructive approach of GAs resulting from the projective split Eq. (10.31) and the separation of an algebraic space into an even and an odd linear subspace following Eq. (10.14). While $G_{p,q}^-$ contains the original vector space by $V_n = \langle G_{p,q} \rangle_1$, there exists an *algebra isomorphism* with respect to $G_{p,q}^+$ [39]

$$G_{p',q'} = G_{p,q}^+ \tag{10.38}$$

The projective split results for a given unit 1-blade $\boldsymbol{e}$ in $p' = q$, $q' = p - 1$ in case of $\boldsymbol{e}^2 \geq 0$, respectively, $p' = p$, $q' = q - 1$ for $\boldsymbol{e}^2 \leq 0$. We will discuss some basic results for $E_3 = \mathbb{R}^3$ and will consider $G_{3,0}$:

Example 10.1: ***Geometric Algebra of the Euclidean Space***

dim $(G_{3,0}) = 8$ (1 scalar, 3 vectors, 3 bivectors, 1 pseudoscalar)
basis $(G_{3,0})$: $\{1, \boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3, \boldsymbol{e}_{23}, \boldsymbol{e}_{31}, \boldsymbol{e}_{12}, \boldsymbol{e}_{123} \equiv I\}$
$\boldsymbol{e}_1^2 = \boldsymbol{e}_2^2 = \boldsymbol{e}_3^2 = 1$, $\boldsymbol{e}_{23}^2 = \boldsymbol{e}_{31}^2 = \boldsymbol{e}_{12}^2 = -1$, $\boldsymbol{e}_{123}^2 = I^2 = -1$
$G_{3,0}^+ \simeq \mathbb{H}$
dim $(G_{3,0}^+) = 4$
basis $(G_{3,0}^+)$: $\{1, \boldsymbol{e}_{23}, \boldsymbol{e}_{31}, \boldsymbol{e}_{12}\}$

Here $\boldsymbol{e}_{ij} = \boldsymbol{e}_i \boldsymbol{e}_j = \boldsymbol{e}_i \wedge \boldsymbol{e}_j$ and $\boldsymbol{e}_{123} = \boldsymbol{e}_1 \boldsymbol{e}_2 \boldsymbol{e}_3$. Application of the duality principle Eq. (10.18) results in $\boldsymbol{e}_{23} = I\boldsymbol{e}_1 \equiv$ -$\boldsymbol{i}$, $\boldsymbol{e}_{31} = I\boldsymbol{e}_2 \equiv \boldsymbol{j}$, and $\boldsymbol{e}_{12} = I\boldsymbol{e}_3 \equiv \boldsymbol{k}$. This 4-D linear space is algebraically isomorphic to the *quaternion algebra*, thus $G_{3,0}^+ \equiv \mathbb{H}$.

While quaternions are restricted to $\mathbb{R}^3$, the same principle applied to $\mathbb{R}^2$ will result in an algebraic isomorphism to *complex numbers* $\mathbb{C}$.

Example 10.2: ***Geometric Algebra of the Euclidean Plane***

dim $(G_{2,0}) = 4$ (1 scalar, 2 vectors, 1 pseudoscalar)
basis $(G_{2,0})$: $\{1, \boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_{12} \equiv I\}$
$\boldsymbol{e}_1^2 = \boldsymbol{e}_2^2 = 1$, $\boldsymbol{e}_{12}^2 = I^2 = -1$
$G_{2,0}^+ \simeq \mathbb{C}$
dim $(G_{2,0}^+) = 2$
basis $(G_{2,0}^+)$: $\{1, \boldsymbol{e}_{12}\}$

Here the imaginary unit $\boldsymbol{i}$ is either a bivector $\boldsymbol{e}_{12}$ or a pseudoscalar I. While in the first case it represents the unit of the directed area embedded in $\mathbb{R}^2$, it indicates in the second case that the unit pseudoscalar is orthogonal to the unit scalar. Thus, any $\boldsymbol{x} \in \mathbb{R}^2$, $\boldsymbol{x} = x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2$, may also be represented as complex number $\mathbf{S} \in \mathbb{C}, \mathbf{S} = x_1 + x_2\boldsymbol{i}$. If $\boldsymbol{i}$ is interpreted as a bivector, the term $\boldsymbol{S}$ will also be called *spinor*.

Any bivector $\mathbf{B} \in G_{3,0}$, $\mathbf{B} = \boldsymbol{a} \wedge \boldsymbol{c}$, may either be represented with respect to the bivector basis

$$\mathbf{B} = B_{23}\boldsymbol{e}_{23} + \mathbf{B}_{31}\boldsymbol{e}_{31} + \mathbf{B}_{12}\boldsymbol{e}_{12} \tag{10.39}$$

as the *pure quaternion* or *vector quaternion*

$$\mathbf{B} = B_{23}\boldsymbol{i} + B_{31}\boldsymbol{j} + B_{12}\boldsymbol{k} \tag{10.40}$$

or as the *dual vector*

$$\mathbf{B} = I\boldsymbol{b} = I(b_1\boldsymbol{e}_1 + b_2\boldsymbol{e}_2 + b_3\boldsymbol{e}_3) \tag{10.41}$$

with $b_1 = B_{23}$, $b_2 = B_{31}$, $b_3 = B_{12}$. The bivector coordinates

$$B_{ij} = (\boldsymbol{a} \wedge \boldsymbol{c})_{ij} = a_ic_j - a_jc_i \tag{10.42}$$

represent an antisymmetric tensor of rank two. Because $\boldsymbol{b} = \mathbf{B}^*$ is orthogonal to $\mathbf{B}$, it corresponds to the vector *cross product* in $G_{3,0}$

$$\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{c} = -I(\boldsymbol{a} \wedge \boldsymbol{c}) \tag{10.43}$$

which is defined in $\mathbb{R}^3$.

10.3.9 Projective and kinematic spaces in geometric algebra

Here we will consider non-Euclidean deformations of the space E_3, which are useful to transform nonlinear problems to linear ones. This can be done by an *algebraic embedding* of E_3 in an extended vector space $\mathbb{R}^4$. In case of the *projective geometry* the result will be the *linearization* of *projective transformations*, and in case of the *kinematics* the *linearization* of the *translation* operation.

If we consider a 3-D *projective space* P_3, it may be extended to a 4-D space $\mathbb{R}^4$ using the *projective split* with respect to a unit 1-blade $\boldsymbol{e}_4$, $\boldsymbol{e}_4^2 = 1$. Thus, $G_{1,3}$ is the GA of the 3-D projective space. Its properties are

Example 10.3: ***Geometric Algebra of the Projective Space***

dim $(G_{1,3}) = 16$ (1 scalar, 4 vectors, 6 bivectors, 4 trivectors, 1 pseudoscalar)
basis $G_{1,3}$: $\{1, \boldsymbol{e}_k, \boldsymbol{e}_{23}, \boldsymbol{e}_{31}, \boldsymbol{e}_{12}, \boldsymbol{e}_{41}, \boldsymbol{e}_{42}, \boldsymbol{e}_{43}, I\boldsymbol{e}_k, \boldsymbol{e}_{1234} = I;$
$k = 1, 2, 3, 4\}$
$\boldsymbol{e}_j^2 = -1$, $j = 1, 2, 3$, $\boldsymbol{e}_4^2 = 1$, $\boldsymbol{e}_{1234}^2 = I^2 = -1$

This formulation leads to a *Minkowski metric* of the space $\mathbb{R}^4$ [40]. The projective split with respect to the homogeneous coordinate vector $\boldsymbol{e}_4$ relates the entities of $\mathbb{R}^4$ to their representations in E_3 such that any $\boldsymbol{x} \in E_3$ can be represented by any $\mathbf{X} \in \mathbb{R}^4$ by

$$\boldsymbol{x} = \frac{\mathbf{X} \wedge \boldsymbol{e}_4}{X_4} \tag{10.44}$$

The coordinate $\boldsymbol{e}_4$ corresponds to the direction of the projection [38].

Finally, we will consider the extension of the Euclidean space E_3 to a 4-D *kinematic space* $\mathbb{R}^4$ using the projective split with even the same unit 1-blade $\boldsymbol{e}_4$, but $\boldsymbol{e}_4^2 = 0$. Thus, $G_{3,0,1}$ will have the following properties:

Example 10.4: ***Geometric Algebra of the Kinematic Space***

dim $(G_{3,0,1}) = 16$ (1 scalar, 4 vectors, 6 bivectors, 4 trivectors, 1 pseudoscalar)
basis $(G_{3,0,1}) : \{1, \boldsymbol{e}_k, \boldsymbol{e}_{23}, \boldsymbol{e}_{31}, \boldsymbol{e}_{12}, \boldsymbol{e}_{41}, \boldsymbol{e}_{42}, \boldsymbol{e}_{43}, I\boldsymbol{e}_k, \boldsymbol{e}_{1234} = I;$
$k = 1, 2, 3, 4\}$
$\boldsymbol{e}_j^2 = 1,\ j = 1, 2, 3\ ,\ \boldsymbol{e}_4^2 = 0,\ \boldsymbol{e}_{1234}^2 = I^2 = 0$
$G_{3,0,1}^+ \simeq \mathbb{H} + I\mathbb{H}$
dim $(G_{3,0,1}^+) = 8$
basis $(G_{3,0,1}^+) : \{1, \boldsymbol{e}_{23}, \boldsymbol{e}_{31}, \boldsymbol{e}_{12}, \boldsymbol{e}_{41}, \boldsymbol{e}_{42}, \boldsymbol{e}_{43}, \boldsymbol{e}_{1234} = I\}$

It can be recognized that the bivector basis of $G_{3,0,1}^+$ has to be divided into two groups $\{\mathbf{e}_{23}, \mathbf{e}_{31}, \mathbf{e}_{12}\}$ and $\{\mathbf{e}_{41}, \mathbf{e}_{42}, \mathbf{e}_{43}\}$ with respect to a duality operation. From this it results that the basis can be built by two sets of quaternions, the one dual to the other. The synthesized algebra is isomorphic to the *algebra of dual quaternions* $\hat{\mathbb{H}}$ [31, 41]

$$G_{3,0,1}^+ \simeq \mathbb{H} + I\mathbb{H} \equiv \hat{\mathbb{H}} \tag{10.45}$$

with $I^2 = 0$.

This algebra is of fundamental importance for handling *rigid displacements* in 3-D space as *linear transformations* [32, 33, 34]. This intuitively follows from the property of a unit quaternion coding general 3-D rotation of a point, represented by a vector, around another point as center of rotation. Instead, in $G_{3,0,1}^+$ there are two lines, related by a rigid displacement in space. Thus, the basic assumption in the presented extension of $G_{3,0}^+$ to $G_{3,0,1}^+$ is to use the representation of the 3-D space by lines instead of points [38, 42]. The resulting geometry is therefore called *line geometry*, respectively, *screw geometry*. The last term results from the fact that the axis of a screw displacement will be an invariant of rigid displacements in $G_{3,0,1}^+$, just as the point that represents the center of a general rotation will be the invariant of this operation in $G_{3,0}^+$.

10.3.10 Geometric entities in geometric algebra

We will express basicgeometric entities in the algebraic languages of the Euclidean, projective and kinematic spaces, respectively. Geometric *entities* are *points*, *lines*, and *planes* as the basic units or *irreducible invariants* of yet higher-order agglomerates. The goal will not be to construct a world of polyhedrons. But the mentioned entities are useful for modeling local processes of perception and action, Section 10.3.1, because of the low dimensionality of the local state space.

In the following we will denote points, lines, and planes by the symbols **X**, **L** , and **E**, respectively, if we mean the entities, which we want to express in any GA.

First, to open the door of an *analytic geometry* (see, e.g., Hestenes [29, chapter 2-6.]), we will look at the entities of $G_{3,0}$. In this algebra the identity

$$\mathbf{X} = \boldsymbol{x} \tag{10.46}$$

$\mathbf{X} \in \langle G_{3,0}\rangle_1, \boldsymbol{x} \in E_3$ is valid. The *point* conception is the basic one and all other entities are aggregates of points. Thus,

$$\boldsymbol{x} \wedge I = 0 \tag{10.47}$$

implicitly defines E_3 by the unit pseudoscalar expressing its collinearity with all $\boldsymbol{x} \in E_3$.

The same principle can be used to define a *line* $\boldsymbol{l}$ through the origin by the nonparametric equation $\boldsymbol{x} \wedge \boldsymbol{l} = 0$, therefore, $\boldsymbol{l} = \lambda\bar{\boldsymbol{l}}$ is the set $\{\boldsymbol{x}\}$ of points belonging to the line, and

$$\mathbf{L} = \boldsymbol{l} \tag{10.48}$$

tells us that all such lines are 1-blades $\mathbf{L} \in \langle G_{3,0}\rangle_1$ of direction $\bar{\boldsymbol{l}}$ and $\lambda \in \langle G_{3,0}\rangle_0$ now defines all points belonging to that subspace.

A more general definition of directed lines, not passing the origin, is based on Hesse's normal form. In this case the line is an inhomogeneous multivector, consisting of the vector $\boldsymbol{l}$ and the bivector **M**

$$\mathbf{L} = \boldsymbol{l} + \mathbf{M} = (1 + \boldsymbol{d})\boldsymbol{l} \tag{10.49}$$

Because Eq. (10.49) is in GA, we have a geometric product on the right side. While the vector $\boldsymbol{l}$ is specifying the *direction* of the line, the bivector $\mathbf{M} = \boldsymbol{dl} = \boldsymbol{d} \wedge \boldsymbol{l}$ specifies its *moment*. From the definition of the *moment* we see that $\boldsymbol{d}$ is a vector orthogonal to $\boldsymbol{l}$, directed from origin to the line. It is the minimal *directed distance* of the line from the origin. In the hyperplane spanned by **M** will also lie the normal vector of the line $\boldsymbol{l}^{-1}$, so that $\boldsymbol{d} = \mathbf{M}\boldsymbol{l}^{-1} = \mathbf{M} \cdot \boldsymbol{l}^{-1}$. Besides, we recognize that both $\boldsymbol{l}$ and $\mathbf{M}$ or $\boldsymbol{l}$ and $\boldsymbol{d}$ are *invariants* of the subspace line. From this it follows that all points $\boldsymbol{x} \in \mathbf{L}$ will parametrically define the line by $\boldsymbol{x} = (\mathbf{M} + \lambda)\boldsymbol{l}^{-1}$.

With the same model of Hesse's normal form we can express the plane $\mathbf{E}$ by the 2-blade *direction* $\mathbf{P}$ and the *moment* $\mathbf{M} = \boldsymbol{d}\mathbf{P} = \boldsymbol{d} \wedge \mathbf{P}$, $\mathbf{M} \in \langle G_{3,0} \rangle_3$,

$$\mathbf{E} = \mathbf{P} + \mathbf{M} = (1 + \boldsymbol{d})\mathbf{P} \tag{10.50}$$

Here again $\boldsymbol{d} = \mathbf{M}\mathbf{P}^{-1} = \mathbf{M} \cdot \mathbf{P}^{-1}$ is the *directed distance* of the plane to the origin and all points belonging to the subspace plane are constrained by $\boldsymbol{x} = (\mathbf{M} + \lambda)\mathbf{P}^{-1}$.

Second, we will demonstrate the *incidence algebra* of points, lines and planes in *projective geometry* $G_{1,3}$.

In $G_{3,0}$ equations such as $\boldsymbol{x} \wedge \boldsymbol{l} = 0$ or $\boldsymbol{x} \wedge (\boldsymbol{l} + \mathbf{M}) = 0$ specify the incidence of any point $\boldsymbol{x}$ and line $\boldsymbol{l}$, respectively, line $\mathbf{L}$. In $G_{1,3}$ the operations joint and meet will result in the corresponding incidences. Let $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \in \langle G_{1,3} \rangle_1$ be noncollinear points. Any $\mathbf{X} \in \langle G_{1,3} \rangle_1$ will be related with $\boldsymbol{x} \in \mathbb{R}^3$ by the *projective split*. Following the *join* Eq. (10.34), we construct a *line* $\mathbf{L} \in \langle G_{1,3} \rangle_2$ by

$$\mathbf{L}_{12} = \mathbf{X}_1 \bigwedge \mathbf{X}_2 \tag{10.51}$$

and a *plane* $\mathbf{E} \in \langle G_{1,3} \rangle_3$ by

$$\mathbf{E}_{123} = \mathbf{X}_1 \wedge \mathbf{X}_2 \wedge \mathbf{X}_3 \tag{10.52}$$

Each point $\mathbf{X}$ on a line $\mathbf{L}$ results in $\mathbf{X} \wedge \mathbf{L} = 0$, and each point $\mathbf{X}$ on a plane $\mathbf{E}$ results in $\mathbf{X} \wedge \mathbf{E} = 0$, thus $\mathbf{L}$ and $\mathbf{E}$ specify subspaces. The intersections of entities result from the *meet* operation Eq. (10.36). If $\mathbf{X}_s$ is a *point of intersection* of a plane $\mathbf{E}$ and a noncollinear line $\mathbf{L}$ we get, for example,

$$\mathbf{X}_s = \mathbf{L} \bigvee \mathbf{E} = \sigma_1 \mathbf{Y}_1 + \sigma_2 \mathbf{Y}_2 + \sigma_3 \mathbf{Y}_3 \tag{10.53}$$

with $\mathbf{E} = \mathbf{Y}_1 \bigwedge \mathbf{Y}_2 \bigwedge \mathbf{Y}_3$, $\mathbf{L} = \mathbf{X}_1 \bigwedge \mathbf{X}_2$, and $\sigma_1, \sigma_2, \sigma_3 \in \langle G_{1,3} \rangle_0$ are the *brackets* of pseudoscalars of each four points:

$$\sigma_1 = [\mathbf{X}_1\mathbf{X}_2\mathbf{Y}_2\mathbf{Y}_3], \quad \sigma_2 = [\mathbf{X}_1\mathbf{X}_2\mathbf{Y}_3\mathbf{Y}_1], \quad \text{and} \quad \sigma_3 = [\mathbf{X}_1\mathbf{X}_2\mathbf{Y}_1\mathbf{Y}_2]$$

The intersection of two noncollinear planes $\mathbf{E}_1 = \mathbf{X}_1 \bigwedge \mathbf{X}_2 \bigwedge \mathbf{X}_3$ and $\mathbf{E}_2 = \mathbf{Y}_1 \bigwedge \mathbf{Y}_2 \bigwedge \mathbf{Y}_3$ will result in the *intersection line*

$$\mathbf{L}_s = \mathbf{E}_1 \bigvee \mathbf{E}_2 = \sigma_1(\mathbf{Y}_2 \bigwedge \mathbf{Y}_3) + \sigma_2(\mathbf{Y}_3 \bigwedge \mathbf{Y}_1) + \sigma_3(\mathbf{Y}_1 \bigwedge \mathbf{Y}_2) \tag{10.54}$$

with $\sigma_1 = [\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Y}_1]$, $\sigma_2 = [\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Y}_2]$ and $\sigma_3 = [\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Y}_3]$. The derivations have been omitted, see, for example, [43]. On the base of this incidence algebra the constraints resulting from two or three cameras on a geometrically intuitive way can be developed [43].

Finally, we will summarize the representation of points, lines, and planes as entities of the *algebra of kinematics* $G^+_{3,0,1}$, with respect to their constituents in $\mathbb{R}^3$. We learned that in this case the Euclidean space was modeled by the set of all possible lines. From this results the importance of the bivector base of $G^+_{3,0,1}$. Besides, we identified the dualities $\boldsymbol{e}_{41} = I\boldsymbol{e}_{23}, \boldsymbol{e}_{42} = I\boldsymbol{e}_{31}, \boldsymbol{e}_{43} = I\boldsymbol{e}_{12}$, which resulted in a *dual quaternion algebra*. Thus, also the desired entities should have dual quaternion structure $\hat{\mathbf{Z}} = \mathbf{Z} + I\mathbf{Z}^*$ with $\mathbf{Z}, \mathbf{Z}^* \in \mathbb{H}$ and $\hat{\mathbf{Z}} \in \hat{\mathbb{H}}$, $\mathbf{Z} = (Z_1, Z_2, Z_3, Z_4)$, $\mathbf{Z}^* = (Z_1^*, Z_2^*, Z_3^*, Z_4^*)$, so that $\hat{Z}_i = Z_i + IZ_i^*$ are *dual numbers*, from which $\hat{\mathbf{Z}}$ may be represented as 4-tuple of dual numbers $\hat{\mathbf{Z}} = (\hat{Z}_1, \hat{Z}_2, \hat{Z}_3, \hat{Z}_4)$. The components $i = 1, 2, 3$ are from $\mathbb{R}^3$ and, therefore, are representing the vector part of a quaternion, $i = 4$ should indicate the scalar component of a quaternion. From this it follows that any $\hat{\mathbf{X}} \in G^+_{3,0,1}$ should be constraint to the hyperplane $X_4 = 1$, even as the other entities $\hat{\mathbf{L}}$ and $\hat{\mathbf{E}}$.

A *line* $\hat{\mathbf{L}} = \hat{\mathbf{X}}_1 \wedge \hat{\mathbf{X}}_2$, $X_{14} = X_{24} = 1$, may be expressed by

$$\hat{\mathbf{L}} = \mathbf{L} + I\mathbf{L}^* \tag{10.55}$$

$I^2 = 0$, where the scalar components of $\mathbf{L}$ and $\mathbf{L}^*$ are *Plücker coordinates* of the line $\hat{\mathbf{L}}$ that also define the coordinate bivector of a plane through the origin of $\mathbb{R}^4$. This plane is spanned by the moment $\mathbf{M} = \boldsymbol{d}\mathbf{L}$ that intersects E_3, defined as a hyperplane in $\mathbb{R}^4$, in the line $\hat{\mathbf{L}}$. Now, we can identify $\mathbf{L}$ as the *line direction*, represented as an element of $\mathbb{H}$ by

$$\mathbf{L} = L_{23}\boldsymbol{e}_{23} + L_{31}\boldsymbol{e}_{31} + L_{12}\boldsymbol{e}_{12} \tag{10.56}$$

and its dual part by the *moment*

$$\mathbf{L}^* = \mathbf{M} = L_{41}\boldsymbol{e}_{23} + L_{42}\boldsymbol{e}_{31} + L_{43}\boldsymbol{e}_{12} \tag{10.57}$$

thus

$$\hat{\mathbf{L}} = \mathbf{L} + I\mathbf{M} \tag{10.58}$$

For the representation of a *point* $\mathbf{X}$ in line geometry we choose two points $\hat{\mathbf{X}}_1 = (0, 0, 0, 1)$ and $\hat{\mathbf{X}}_2 = (X_1, X_2, X_3, 1)$. This results in

$$\hat{\mathbf{X}} = 1 + I\mathbf{X} \tag{10.59}$$

where $\mathbf{X}$ is identified as the point $\hat{\mathbf{X}}_2$ on the hyperplane $X_4 = 1$.

Finally, let us consider a *plane* $\hat{\mathbf{E}}$ as a tangential plane at a point $\hat{\mathbf{X}}$ that is collinear with the line and orthogonal to its *moment*. We get

$$\hat{\mathbf{E}} = \mathbf{P} + I\mathbf{M} \tag{10.60}$$

where $\mathbf{P}$ is the 2-blade *direction* of Eq. (10.50) and $I\mathbf{M} = \mathbf{D}$ determines the 2-blade directed distance to the origin.
At this point we summarize that the considered entities assume different representations in dependence of the interesting aspects. By switching the signature of the embedding algebra, the one or another aspect comes into play. But if we consider the objects of interest in computer vision as moving rigid bodies, projective and kinematics interpretations have to be fused in a common algebraic approach. This will be a matter of future research.

10.3.11 Operational entities in geometric algebra

Operational entities are geometric transformations that code the net movements of any geometric entity. The simplest ones are *reflection*, *rotation*, *translation* and *scaling* of points. A general principle to get the complete set of primitive geometric operations in E_3 is given by the *Lie group operations*. This space of geometric transformations is of dimension 6. In the given context, first of all we are only interested in a subset related to rotation and translation. If rigid point agglomerations are of interest, or if the degrees of freedom increase because of increasing dimension of the space $\mathbb{R}^n$, also these primitive operations will become complicated, time consuming, or they will assume nonlinear behavior. Because these operations mostly are of interest in combinations, the question is if there are algebraic embeddings in which these combine to new ones with own right to be considered as basic operations. An example is *rigid displacement*. There has been considerable work in robotics to reach low symbolic complexity for coding the movement of complex configurations and simultaneously to reach low numeric complexity [33, 44, 45, 46]. In the frame of PAC, in addition the fusion of geometric models for geometric signal deformations and robot actions becomes an important problem, for example, with respect to *camera calibration* [47], *pose estimation*, or *gesture tracking* [48]. We call the geometric transformations entities because they themselves are represented by multivectors in GA. Of course, their special form and their properties depend strongly on the special algebraic embedding of the problem.

First, we will consider $G_{3,0}$, see Example 10.1, and $G_{2,0}$, see Example 10.2. Because of Eqs. (10.3) and (10.27), any two vectors $\boldsymbol{a}, \boldsymbol{b} \in G_{2,0}$ result in an inhomogeneous multivector

$$\mathbf{C} = \boldsymbol{ab} = \cos\theta + I\sin\theta = \exp(I\theta) \qquad (10.61)$$

where θ is the radian measure of the enclosed angle and the bivector $\Theta = I\theta$ describes an angular relation between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ in the plane. Indeed, Θ is a directed area, circularly enclosed between $\boldsymbol{a}$ and

$\boldsymbol{b}$, whose direction indicates a rotation from $\boldsymbol{a}$ to $\boldsymbol{b}$. A rotation from $\boldsymbol{b}$ to $\boldsymbol{a}$ would result in the reversed multivector

$$\tilde{\mathbf{C}} = \boldsymbol{ba} = \cos\theta - I\sin\theta = \exp(-I\theta) \tag{10.62}$$

Any rotated vectors $\boldsymbol{a}$ and $\boldsymbol{a}'$ are related by either $\boldsymbol{a}' = \boldsymbol{a}\exp(I\theta)$ in case of an orthogonal rotation or by $\boldsymbol{a}' = \exp(\frac{I\theta}{2})\boldsymbol{a}\exp(-\frac{I\theta}{2})$ in case of a general rotation. We write instead (see Hestenes [29])

$$\boldsymbol{a}' = \mathbf{Ra}\tilde{\mathbf{R}} \tag{10.63}$$

and call

$$\mathbf{R} = \exp\left(\frac{I\theta}{2}\right) \tag{10.64}$$

a *rotor* with $\mathbf{R}\tilde{\mathbf{R}} = 1$.

In comparison to other operational codes of rotation the rotor has several advantages. Its structure as a bivector is the same as that of the objects. It is a linear operator and as such it is the same, whatever the grade of the geometric object and whatever the dimension of the space is. In contrast to matrix algebra a rotor is completely coordinate independent. The linearity results in $\mathbf{R}_3 = \mathbf{R}_2\mathbf{R}_1$, thus

$$\boldsymbol{a}'' = \mathbf{R}_3\boldsymbol{a}\tilde{\mathbf{R}}_3 = \mathbf{R}_2\boldsymbol{a}'\tilde{\mathbf{R}}_2 = \mathbf{R}_2\mathbf{R}_1\boldsymbol{a}\tilde{\mathbf{R}}_1\tilde{\mathbf{R}}_2 \tag{10.65}$$

Because a rotor is a bivector, it will be isomorphic to a unit quaternion in $G^+_{3,0}$. If we couple rotation with scaling, then

$$\mathbf{S} = s + \mathbf{R}, \tag{10.66}$$

$s \in \langle G_n \rangle_0$, will be a *spinor*. Equation (10.66) can be expressed in multiplicative form because of Eq. (10.64). Then the action of a spinor corresponds to the action of another rotor R' that is not of unit magnitude. The algebra $G^+_{3,0}$ is also called *spinor algebra*.

A translation in $\mathbb{R}^n$ corresponds to an n-dimensional vector, say $\boldsymbol{t}$. With respect to a single point $\mathbf{x} \in \mathbb{R}^n$ the translation will be a linear operation because $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}$. But this is not valid if two points $\boldsymbol{x}$ and $\boldsymbol{y}$, related by a distance $\boldsymbol{d} = \boldsymbol{y} - \boldsymbol{x}$, will simultaneously be translated by the same vector because

$$\boldsymbol{d} + \boldsymbol{t} \neq (\boldsymbol{y} + \boldsymbol{t}) - (\boldsymbol{x} + \boldsymbol{t}) \tag{10.67}$$

A rigid transformation preserves the distance between points of any aggregation of such. It results in a rigid displacement of the aggregation. *Rigid transformations* are rotation and translation, both coupled together. With the result of Eq. (10.67) no linear method exists in Euclidean space to perform general rigid displacements by linear transformations.

The introduction of *homogeneous coordinates* results in several different linearizations of rigid transformations. But the best algebraic embedding of the problem is given by the GA of the *kinematic space*, Example 10.4, or *motor algebra* $G^+_{3,0,1}$. In this algebra the rigid transformations are expressed by linear relation of two lines. One *oriented line* is representing a rotation $\mathbf{R}$ using a bivector, another oriented line is representing a translation $\mathbf{T}$, also using a bivector.. This is the model of a *motor* $\hat{\mathbf{C}}$ as introduced by [49] and developed by Bayro-Corrochano et al. [50].

$$\hat{\mathbf{C}} = \hat{\mathbf{X}}_1\hat{\mathbf{X}}_2 + \hat{\mathbf{X}}_3\hat{\mathbf{X}}_4 = \mathbf{R} + I\mathbf{R}' \tag{10.68}$$

where $\mathbf{R}$ is the rotor as introduced in Eq. (10.64), and $\mathbf{R}'$ is another rotor, modified by a *translator*,

$$\mathbf{T} = \exp\left(\frac{\boldsymbol{t}I}{2}\right) = 1 + I\frac{\boldsymbol{t}}{2} \tag{10.69}$$

$\boldsymbol{t} = t_1\boldsymbol{e}_{23} + t_2\boldsymbol{e}_{31} + t_3\boldsymbol{e}_{12}$, see Bayro-Corrochano et al. [50]. The translator can be seen in these terms as a plane of rotation, translated by $\boldsymbol{t}$ from the origin in direction of the rotation axis. The motor degenerates in case of coplanarity of both axes to a rotor, representing pure rotation, else it represents a *screw motion* by

$$\hat{\mathbf{C}} = \mathbf{R} + I\frac{\mathbf{t}}{2}\mathbf{R} = \left(1 + I\frac{\mathbf{t}}{2}\right)\mathbf{R} = \mathbf{TR} \tag{10.70}$$

Notice that Eq. (10.70) represents a motor as an element of the motor algebra and that Eq. (10.68) represents a motor as an element of the dual quaternion algebra, $\hat{\mathbf{C}} \in \hat{\mathbb{H}}$. Although both are isomorphic algebras, the motor algebra offers more explicitly the relation between translation and rotation and, therefore, will be more usefully applied in modeling chained movements. If we apply this code of a rigid transformation to a line $\hat{\mathbf{L}} = \mathbf{L} + I\mathbf{M}$, see Eq. (10.58), then the transformed line will be

$$\hat{\mathbf{L}}' = \mathbf{L}' + I\mathbf{M}' = \hat{\mathbf{C}}\hat{\mathbf{L}}\hat{\tilde{\mathbf{C}}} \tag{10.71}$$

or in full [50]

$$\hat{\mathbf{L}}' = \mathbf{RL\tilde{R}} + I(\mathbf{RL\tilde{R}}' + \mathbf{R}'\mathbf{L\tilde{R}} + \mathbf{RM\tilde{R}}) \tag{10.72}$$

The motor algebra and the isomorphic dual quaternion algebra not only have importance for dense coding of robotic tasks, for example, chained links, but they will also have high impact on the analysis of manifolds such as images and image sequences. In the motor algebra not only linearization of certain transformations will be gained. It also enables us to use a high degree of invariance with respect to the related geometric entities.

10.4 Applications of the algebraic framework

We have motivated the need of a powerful framework of algebraic nature from the viewpoint of designing behavior-based systems and from well-known intrinsic problems, such as limitations of the involved disciplines. Furthermore, we could present a useful framework that was formulated more than one hundred years ago but little noticed to date in engineering applications. This framework is geometric algebra or *Clifford algebra*.

We are developing the application of the algebra with respect to robotics, computer vision, multidimensional signal theory, pattern recognition, and neural computing. In the last years we were able to publish several preliminary results. The actual list of publications and reports can be found on www: `http://www.ks.informatik.uni-kiel.de`. But in fact, both the shaping of the framework and the application of tools are in their infancy. We will not go into details of applications in computer vision and robotics. Instead, we will introduce two aspects that are related to the use of GA with respect to multidimensional structure analysis and recognition. Both topics stress that the GA will have a great impact on describing manifolds. This is also the field where the global frame of events, entities, or patterns meets the local frame. This local frame is strongly related to Lie algebra and Lie groups. Because every matrix *Lie group* can be realized as a *spin group*, and because spin groups consist of even products of unit vectors—generators of a spin group are bivectors, it does not surprise that GA will give the right frame for a Lie group based design of the local frame, see, for example, [2, chapter 8] and [51, 52].

10.4.1 Image analysis and Clifford Fourier transform

The recognition of patterns in images means capturing the patterns as a whole and comparing these entities with equivalence classes at hand. In image processing as in engineering in general, linear methods as linear transformations or linear filters are preferable to non linear ones because they are easier to design and handle. Unfortunately, linear methods most often are not useful to describe *multidimensional structures* or to design filters that specifically make explicit such structures. In case of global patterns that consist of multiple entities—but which do not have to be distinguished, this does not matter. The Fourier transform, for example, is a global, linear and complete transformation whose power spectrum in such cases can be used to distinguish between several global patterns. In case of linear filters the forementioned problems, quite the reverse, do limit the set of patterns that can be modeled to be cross-correlated. Filters are not universal tools but highly specified ones. They have not to be complete in general, yet complete with respect to the modeled structures.

There are several algebraic approaches to overcome these limitations, for example, the *tensor filter approach* [26, 27] or the *Volterra filter approach* [53, 54, 55]. Both approaches resemble to a certain extent the way we want to sketch. This does not surprise because GA and the other methods are based on related algebraic roots. Yet there are specific differences that should not be discussed here.

In the Volterra series approach of nonlinearity there is an interesting equivalence of the order of nonlinearity and the dimension of the operator, so that the product of both remains constant. That means, if the order of nonlinearity of image processing in dimension N is given by just this dimension and the dimension of an operator should be the same, then the constant would be N^2. To design instead an equivalent filter of first order would necessitate a filter of dimension N^2. The basic origin of the nonlinearity problem in the case of multidimensional LSI operators is that their eigenvectors should correspond with the dimension of the signals. Because the eigenvectors of LSI operators correspond to the basis functions of the Fourier transform, the key for solving the problem is strongly related to the algebraic embedding of the Fourier transform.

These difficulties are also mentioned in textbooks on signal theory—there is no unique definition of a multidimensional phase in Fourier transform. Precisely the phase is the feature that represents the correlations, respectively symmetries of multidimensional structure. It is not a matter of fate to have no multidimensional phase, but a matter of algebraic adequate embedding of the problem. Zetzsche and Barth [56] argued that for 2-D signals the 1-D basis functions of the Fourier transform should be coupled by a logical AND-operator. Later they presented a linear approach of 2-D filtering using the Volterra series method [57]. To prevent 4-D filters, they developed an interesting scheme to operate on a parametric surface in the 4-D space.

Our proposition is that the linear transformation of an N-D signal has to be embedded in a geometric algebra $G_{N,0}$, which means in a linear space of dimension 2^N (see [58]).

We call the *N-D Fourier transform* $\mathbf{F}^N(\boldsymbol{u})$ in GA the *Clifford Fourier transform* (*CFT*); it reads [58]

$$\mathbf{F}^N(\boldsymbol{u}) = \mathcal{F}_c\{f\} = \int \cdots \int f(\boldsymbol{x}) \mathbf{C}^N_{\boldsymbol{u}}(\boldsymbol{x}) d^N \boldsymbol{x} \tag{10.73}$$

$\boldsymbol{x}, \boldsymbol{u} \in \mathbb{R}^N$, with the *CFT kernel*

$$\mathbf{C}^N_{\boldsymbol{u}}(\boldsymbol{x}) = \prod_{k=1}^{N} \exp(-2\pi i_k u_k x_k) \tag{10.74}$$

The inverse CFT is given by

$$f(\boldsymbol{x}) = \mathcal{F}_c^{-1}\{\mathbf{F}^N\}(\boldsymbol{x}) = \int \cdots \int \mathbf{F}^N(\boldsymbol{u})\tilde{\mathbf{C}}_{\boldsymbol{x}}^N(\boldsymbol{u})d^N\boldsymbol{u} \tag{10.75}$$

where the kernel of the inverse CFT

$$\tilde{\mathbf{C}}_{\mathbf{x}}^N(\boldsymbol{u}) = \prod_{k=0}^{N-1} \exp(2\pi i_{N-k}u_{N-k}x_{N-k}) \tag{10.76}$$

is similar to the reverse of a 2^N-vector in the language of GA and corresponds to the conjugate. The CFT kernel spans the 2^N-D space of the GA in case of Fourier transform of an N-D signal. Each 1-D kernel component

$$\mathbf{C}_{u_k}^1(x_k) = \exp(-2\pi i u_k x_k) \tag{10.77}$$

is a bivector (compare Section 10.3.6), $\mathbf{C}_{u_k}^1(x_k) \in G_{2,0}^+$, see Example 10.2.

In the 2-D case we have a quaternionic Clifford kernel $\mathbf{C}_{\boldsymbol{u}}^2(\boldsymbol{x}) \in G_{3,0}^+$ (see Example 10.1),

$$\mathbf{C}_{\mathbf{u}}^2(\boldsymbol{x}) = \mathbf{C}_{u_1}^1(x_1) \wedge \mathbf{C}_{u_2}^1(x_2) \tag{10.78}$$

In that algebra exist three different unit 2-blades, which results in enough degrees of freedom for complete representation of all symmetries of the signal.

The 2-D CFT will be called *quaternionic Fourier transform* (QFT). We can generalize the scheme to the preceding proposition about the adequate embedding of N-D Fourier transforms. With this the *Hartley transform* assumes an interesting place in the order scheme [58]. While the Hartley transform covers all symmetries of a signal, the 1-D complex Fourier transform is adequate to represent even and odd *symmetry* in one direction, and the CFT allows for separate representation of even and odd symmetry in orthogonal directions without fusion as in the case of 2-D complex Fourier transform.

This important property becomes visible in Fig. 10.1. There are two nested plots of basis functions of the 2-D Fourier transform in spatial domain with each five frequency samples in orthogonal directions. The basis functions of the complex Fourier transform (top) only represent 1-D structures. Even and odd symmetry occur only in one direction. The basis functions of the quaternionic Fourier transform (bottom) in contrast represent 2-D structures. Even and odd symmetries occur in each 2-D basis function in orthogonal directions, besides in the degenerated cases $\boldsymbol{x} = 0$ or $\boldsymbol{y} = 0$.

Because $\mathbf{F}^2(\boldsymbol{u}) \in G_{3,0}^+$, there are one real and three imaginary components

$$\mathbf{F}^2 = \mathcal{R}(\mathbf{F}^2) + i\mathcal{I}(\mathbf{F}^2) + j\mathcal{J}(\mathbf{F}^2) + k\mathcal{K}(\mathbf{F}^2) \tag{10.79}$$

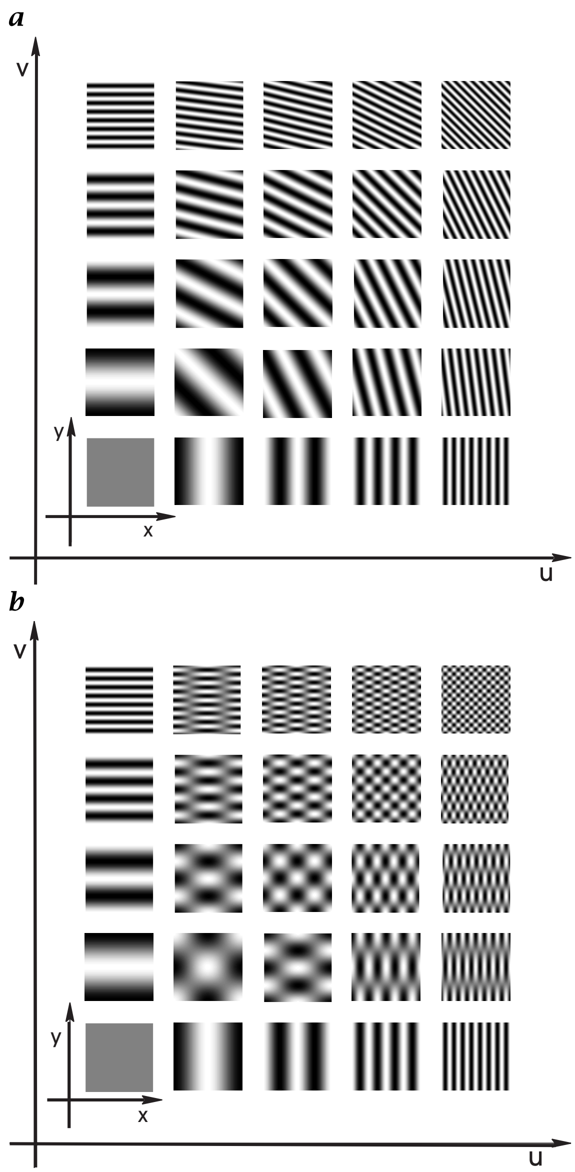

Figure 10.1: *Basis functions of **a** the 2-D complex Fourier transform; and **b** the 2-D quaternionic Fourier transform.*

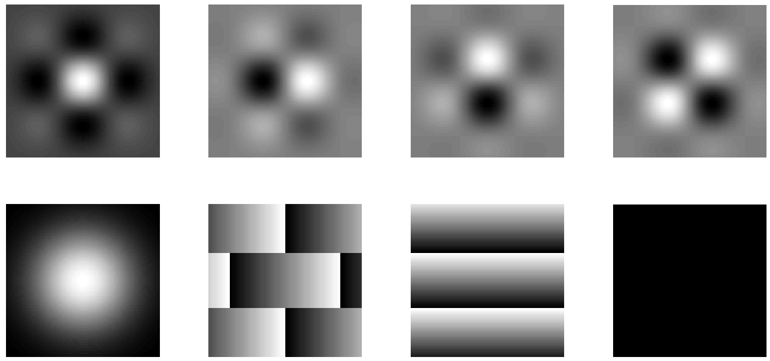

Figure 10.2: *Quaternionic Gabor filter: real and three imaginary components (top); magnitude and three phase angles (bottom).*

which we give a *triple of phase angles* [59]

$$(\Phi, \Theta, \Psi) \in [-\pi, \pi[\times [-\frac{\pi}{2}, \frac{\pi}{2}[\times [-\frac{\pi}{4}, \frac{\pi}{4}] \tag{10.80}$$

This global scheme of algebraic embedding gains its importance from the ability to transfer it to a local scheme, thus to extract the *local phase* of any 2-D structures by means of linear filters. This necessitates formulating the algebraic extension of the analytic signal [60]. The *quaternionic analytic signal* of a real 2-D signal in frequency domain reads

$$\mathbf{F}_A^2(\boldsymbol{u}) = (1 + \mathrm{sign}(u_1))(1 + \mathrm{sign}(u_2))\mathbf{F}^2(\boldsymbol{u}) \tag{10.81}$$

and in spatial domain

$$f_A^2(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{n} \cdot \boldsymbol{f}_{\mathcal{H}i}(\boldsymbol{x}) \tag{10.82}$$

with the Hilbert transform $\boldsymbol{f}_{\mathcal{H}i}(\boldsymbol{x}) = (f_{\mathcal{H}_1}, f_{\mathcal{H}_2}, f_{\mathcal{H}})^T$,

$$f_{\mathcal{H}} = f ** \frac{1}{\pi^2 xy}, \; f_{\mathcal{H}_1} = f ** \frac{\delta(y)}{\pi x}, \; f_{\mathcal{H}_2} = f ** \frac{\delta(x)}{\pi y} \tag{10.83}$$

and $\boldsymbol{n} = (i, j, k)^T$. This scheme of *Hilbert transform* to compute the 2-D analytic signal resembles the results of Hahn [61, 62], but has better algebraic properties, see [60].

Finally, the computation of the local spectral features can be done with an algebraically extended Gabor filter [63]. Figure 10.2 shows on the top row from left to right the real, the i-imaginary, the j-imaginary, and the k-imaginary components of the *quaternionic Gabor filter*. On the bottom row from left to right there is shown the magnitude and

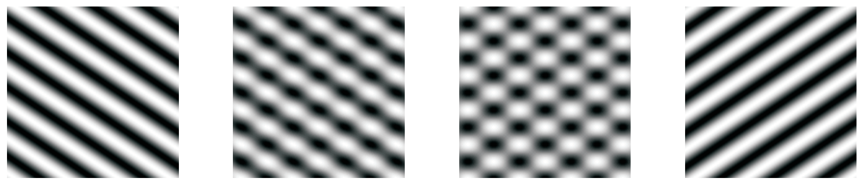

Figure 10.3: *Four texture patterns. Left and right: pure 1-D signals, embedded into 2-D; middle: weighted superpositions to true 2-D signals.*

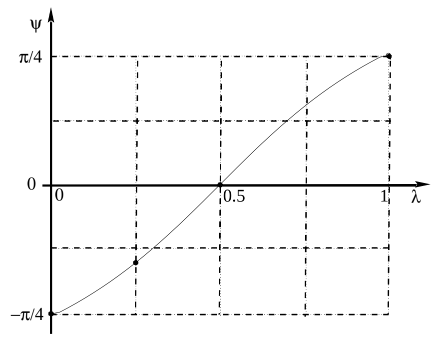

Figure 10.4: *Phase angle Ψ of QFT discriminates the textures of Fig. 10.3.*

the phases Φ, Θ, and Ψ. We will demonstrate the advantage of using the presented scheme of the *2-D phase*. In Fig. 10.3 four textures can be seen that result from superposition of two unbent cosine structures $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ with different spreading directions in 2-D following the rule $\boldsymbol{f}(\lambda) = (1-\lambda)\boldsymbol{f}_1 + \lambda\boldsymbol{f}_2$. From left to right we have $\lambda = 0, 0.25, 0.5$ and 1. Both complex and quaternionic Gabor filters result in the same horizontal frequency of 0.035 pixel^{-1} and in the same vertical frequency of 0.049 pixel^{-1} for all images. Therefore, no linearly local 1-D analysis could discriminate these patterns. But from the quaternionic phase we get also the parameter Ψ. In Fig. 10.4 we see that the quaternionic phase Ψ well discriminates the textures of Fig. 10.3. It should be noticed that in case of complex Gabor filters four convolutions have been applied in total (two in horizontal and two in vertical direction), while quaternionic Gabor filters need the same number of convolutions. Thus, we conclude that an adequate algebraic embedded design of LSI-operators will result in complete representations of local N-D structures. The development of a linear multidimensional system theory is on the way. A *quaternionic FFT algorithmus* has already been developed.

10.4.2 Pattern recognition and Clifford MLP

Neural networks play a crucial role in designing behavior-based systems. Learning of competence guarantees the designer robustness and invariance of the systems to a certain degree. Within the class of feedforward nets there are two important classes—perceptron derivatives and *radial basis function* (RBF, see Volume 2, Section 23.4) derivatives (for a review of neural networks see Volume 2, Chapter 23). *Multilayer perceptrons* (MLPs, see Volume 2, Section 23.2) have been proven to be the best universal approximators. They operate with a global activation function. Nets of RBF use a local sampling scheme of the manifold. Especially the MLP can be embedded into the geometric algebra [64, 65].

The basic motivation of this research is to design neural nets not only as best universal approximators but as specialists for certain classes of geometric or operational entities. The aim is to find new approaches for the design of PAC systems. The key for reaching this consists of a kind of algebraic blowing up the linear associator because it operates only in the vector space scheme. This means it can handle points only as geometric entities but does not use the whole spectrum of entities with which the linear space of the GA is endowed. The assumption that an MLP can successfully use the additional degrees of freedom of the algebraic coding has been confirmed. Additional degrees of freedom do not only result in better generalization, but accelerate learning. Our limited experience permits interpreting the results as positive ones. We learned that the algebraically extended nets use multivectors both as geometric and operational entities. We call the algebraically embedded MLP a *Clifford MLP* or CMLP . Of course, we know from Section 10.3 that there are multiple algebras. Our design scheme enables us to activate the one or the other, thus we are able to look at the data from several different viewpoints [65].

There is also some research from other groups with respect to design complex and quaternionic neurons, see, for example, [66, 67, 68, 69]. However they developed either special nets for complex or quaternionic numbers or could not handle geometric algebras with *zero divisors* [69]. Our design scheme is bottom up, starting from first principles of geometric algebra and resulting in general CMLPs that use a component wise *activation function* that results in an automorphism that prevents zero divisor problems during backpropagation.

The mapping function of a traditional McCulloch-Pitts neuron for input x, output o, threshold θ and activation function f is given by

$$o = f\left(\sum_{i=1}^{N} w_i x_i + \theta\right) \tag{10.84}$$

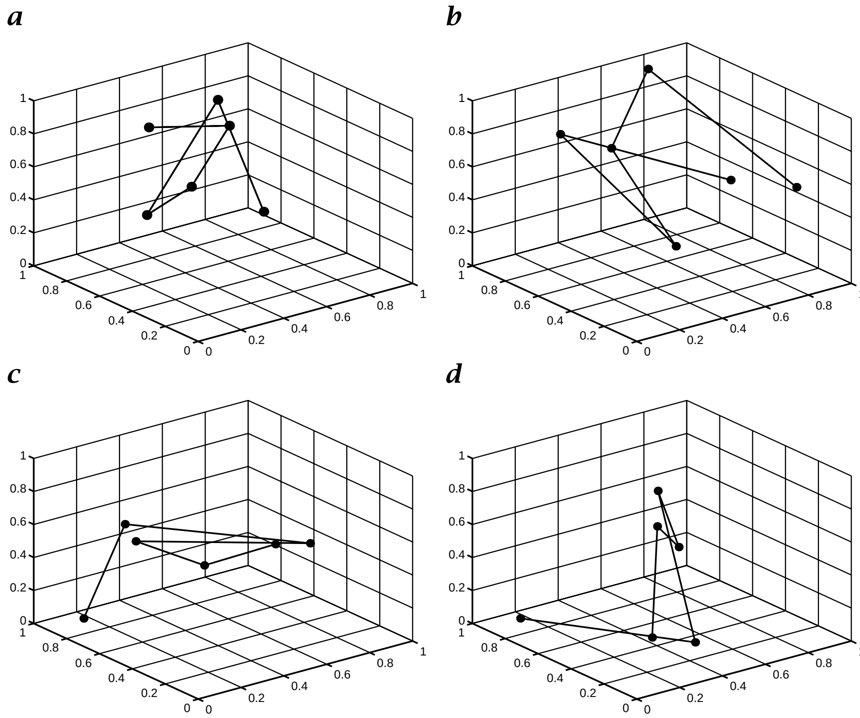

Figure 10.5: *Three-dimensional training patterns for MLP- and CMLP-based classification of rigid point agglomerates.*

The *Clifford McCulloch-Pitts neuron* [70] on the other hand has the structure

$$\boldsymbol{o} = \boldsymbol{f}(\boldsymbol{w}\boldsymbol{x} + \boldsymbol{\theta}) = \boldsymbol{f}(\boldsymbol{w} \wedge \boldsymbol{x} + \boldsymbol{w} \cdot \boldsymbol{x} + \boldsymbol{\theta}) \tag{10.85}$$

It contains the scalar product as a vector algebra operation from Eq. (10.84)

$$\boldsymbol{f}(\boldsymbol{w} \cdot \boldsymbol{x} + \theta) = f(\alpha_0) \equiv f(\sum_{i=1}^{N} w_i x_i + \theta) \tag{10.86}$$

and the nonscalar components of any grade of the GA

$$\boldsymbol{f}(\boldsymbol{w} \wedge \boldsymbol{x} + \boldsymbol{\theta} - \theta) = f(\alpha_1)\boldsymbol{e}_1 + f(\alpha_2)\boldsymbol{e}_2 + \cdots + f(\alpha_{2^N-1})\boldsymbol{e}_{1\ldots N} \tag{10.87}$$

To demonstrate the gained performance in comparison to a usual MLP we show in Figure 10.5 four different 3-D figures that are coded with respect to their 3-D point coordinates. The chords connecting two points should demonstrate the shape as 3-D rigid agglomerates of points. The task is simply to learn these patterns with three real hidden neurons of an MLP, respectively, one quaternionic hidden neuron

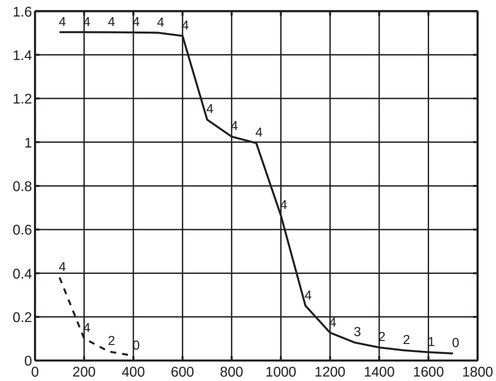

Figure 10.6: *Rate of false classifications by MLP (solid line) and CMLP (broken line) for the patterns of Fig. 10.5.*

of a CMLP, and to gain zero false classifications. In Fig. 10.6 we see the error of learning versus the number of cycles. Besides, the curves (MLP (solid line), CMLP (broken line)) are labeled with the numbers of false classified patterns. Only if the learning error measure drops to approximately 0.02 will both nets gain zero false classifications. Yet the single quaternionic hidden neuron of the CMLP does learn the task with 1/4 the cycles the three real hidden neurons of the MLP need. The performance gain of the CMLP results from its intrinsic capability to represent $G^+_{3,0}$. If we would interpret the need of one or three hidden neurons by the two kinds of nets only with respect to representing the components of the 3-D point coordinate vectors, accelerated learning by the CMLP would not result. Instead, the CMLP uses bivectors for representation of the chords between points. These higher-order entities result in an increase of convergence to gain competence as in Fig. 10.6.

Because GA represents not only geometric but also operational entities, it is not surprising that a CMLP is able to learn geometric transformations. We will demonstrate this for a *quaternionic CMLP*, again with one hidden neuron. In Fig. 10.7 we see a patch of connected planes in 3-D space. To the left is the starting pose, to the right the ending pose. The learned transformation is an affine one. The top row is showing the training data with the following parameters of transformation: rotation 45° with respect to axis $[0, 0, 1]$, scaling factor 0.8, translation vector $[0.4, -0.2, 0.2]$. On the bottom row we see the test data, which are changed with respect to the training data by: rotation -60° with respect to axis $[0.5, \sqrt{0.5}, 0.5]$. In Fig. 10.8 we see the comparison of the demanded pose (crosses) with the computed one (solid lines). The quaternionic CMLP uses one hidden neuron to learn exactly any agglomerates of lines with respect to a similarity transformation.

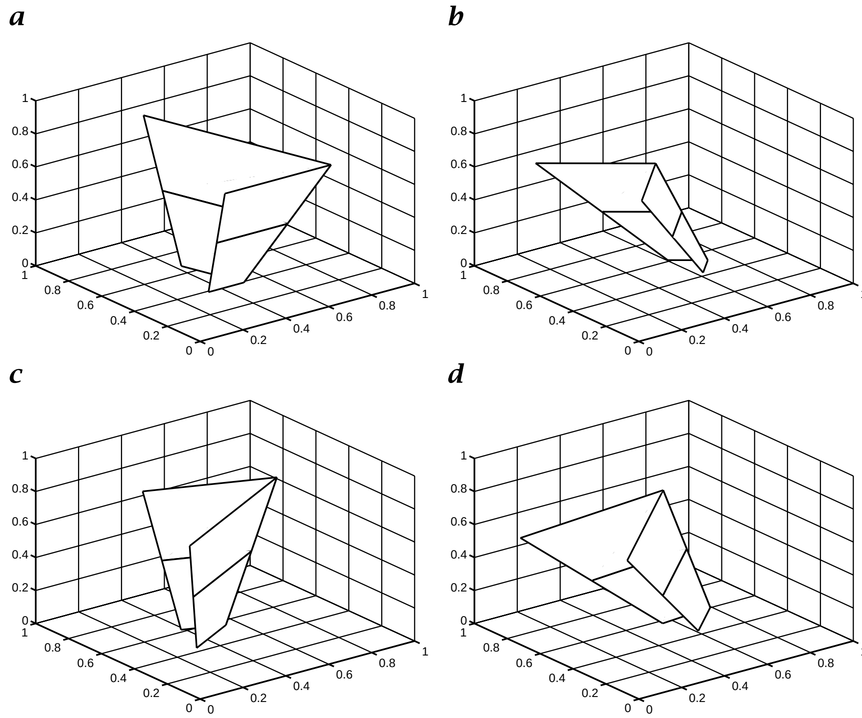

Figure 10.7: *Learning of an affine transformation by a quaternionic CMLP. Top row: training data; bottom row: test data; left column: input data; right column: output data.*

A generalization such as the one shown in Fig. 10.7 cannot be obtained using an MLP with an arbitrary number of neurons. In [68] we can find a complex MLP with an algebraic structure that fits our CMLP on the level of complex numbers. There are also some presentations of learned geometric transformations and the functional analytic interpretation. Our presented frame of algebraic interpretation of the results is more general. It works for all algebraic embeddings. Any Clifford neuron can learn all the group actions that are intrinsic to the corresponding algebra and that are not limited to linear transformations as in the case of a real perceptron.

The presented results demonstrate that algebraic embedded neural nets are worth considering and that we can design on that base a new class of neural nets that constitute an agglomeration of experts for higher-order entities and geometric transformations, respectively. Because both MLPs and CMLPs operate with linear separation functions for each hidden neuron, the algebraic embedding supports this scheme of coding.

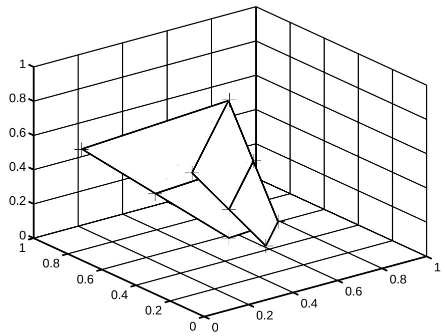

Figure 10.8: *Actual results (solid lines) and demanded results (crosses) of learning an affine transformation by a quaternionic CMLP, see Fig. 10.7.*

10.5 Summary and conclusions

We presented an algebraic language for embedding the design of behavior-based systems. This is geometric algebra, also called Clifford algebra (Clifford originally gave it the name geometric algebra). Our motivation in introducing this language for different disciplines (e.g., computer vision, signal theory, robotics, and neural computing) is to overcome some of the intrinsic problems of the disciplines and to support their fusion in the frame of the perception-action cycle.

We demonstrated the basic ideas of the framework to make obvious both their potential and the demanding task of developing this language to a tool package that is comparable to vector algebra.

Although both the research work and the applications are in their infancy, we presented some examples that can provide us with an impression of the gained attractive performance. We could demonstrate that the embedding of a task into geometric algebra opens the door to linear operations with higher order entities of geometric and kinematic schemes. On that basis, traditional problems may be reformulated with the effects of linearization, higher efficiency, coordinate independency and greater compactness on a symbolic level of coding. As a result of research in this field, one day we can motivate the VLSI design of Clifford processors to gain real profit from the compactness of the language if used for the design of PAC systems.

A dream could become reality: That in the future we have autonomous robots, acting in the complex real world with (visual) percepts and brains that are computers that are capable of representing and manipulating multivectors.

Acknowledgments

The presented research work is supported by DFG grant So 320-2-1 and Studienstiftung des deutschen Volkes.

This chapter was written using research results to which several staff members and many students of the Cognitive Systems Research Group of the Kiel University made contributions. These are E. Bayro-Corrochano, K. Daniilidis (currently in Philadelphia), T. Bülow, S. Buchholz, Yiwen Zang, M. Felsberg, V. Banarer, D. Kähler, and B. Rosenhahn. Our secretary F. Maillard had the hard job of preparing the TEX file.

Our group also thanks J. Lasenby, Cambridge University, for cooperation and D. Hestenes, Tempe, AZ for his stimulating interest in our work.

10.6 References

[1] Clifford, W. K., (1882). *Mathematical Papers.* London: Mcmillan.

[2] Hestenes, D. and Sobczyk, G., (1984). *Clifford Algebra to Geometric Calculus.* Dordrecht: D. Reidel Publ. Comp.

[3] Newell, A. and Simon, H. A., (1976). Computer science as empirical enquiry: symbol and search. *Comm. of the ACM*, **19**:113–126.

[4] Lozano-Perez, T., (1983). Spatial planning: a configuration space approach. *IEEE Trans. Computers*, **32**:108–120.

[5] Marr, D., (1982). *Vision.* San Francisco: W.K. Freeman.

[6] Koenderink, J. J., (1992). Wechsler's vision. *Ecological Psychology*, **4**:121–128.

[7] Hamada, T., (1997). Vision, action and navigation in animals. In *Visual Navigation*, Y. Aloimonos, ed., pp. 6–25. Mahwah, NJ: Lawrence Erlbaum Assoc., Publ.

[8] Fermüller, C. and Aloimonos, Y., (1995). Vision and action. *Image and Vision Computing*, **13**:725–744.

[9] Sommer, G., (1997). Algebraic aspects of designing behavior based systems. In *Algebraic Frames for the Perception-Action Cycle*, G. Sommer and J. Koenderink, eds., Lecture Notes in Computer Science, 1315, pp. 1–28. Berlin, Heidelberg: Springer.

[10] Pauli, J., (1997). Projective invariance and orientation consensus for extracting boundary configurations. In *Mustererkennung 1997*, E. Paulus and F. Wahl, eds., pp. 375–383. Berlin, Heidelberg: Springer.

[11] Michaelis, M. and Sommer, G., (1995). A Lie group approach to steerable filters. *Pattern Recognition Letters*, **16**:1165–1174.

[12] Bruske, J. and Sommer, G., (1997). An algorithm for intrinsic dimensionality estimation. In *Computer Analysis of Images and Patterns*, G. Sommer, K. Daniilidis, and J. Pauli, eds., Lecture Notes in Computer Science, 1296, pp. 9–16. Berlin, Heidelberg: Springer.

[13] Sanger, T. D., (1995). Optimal movement primitives. In *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, eds., Vol. 7, pp. 1023–1030. Cambridge, MA: MIT Press.

[14] Aloimonos, Y. (ed.), (1997). *Visual Navigation.* Mahwah, NJ: Lawrence Erlbaum Assoc., Publ.

[15] Garding, J., Porill, J., Mayhew, J. E. W., and Frisby, J. P., (1995). Stereopsis, vertical disparity and relief transformations. *Vision Research*, **35**:703–722.

[16] Buck, D., (1997). *Projektive Tiefenrepräsentation aus partiell kalibrierten Stereokamerasystemen.* Diploma thesis, Institut für Informatik, Christian-Albrechts-Universität zu Kiel, Kiel, Germany.

[17] Faugeras, O., (1995). Stratification of three-dimensional vision: projective, affine, and metric representations. *Jour. Opt. Soc. Amer.*, **A12**:465–484.

[18] Koenderink, J. J., (1993). Embodiments of geometry. In *Brain Theory*, A. Aertsen, ed., pp. 3–28. Amsterdam: Elsevier Science Publ.

[19] Koenderink, J. J., (1990). The brain a geometry engine. *Psychological Research*, **52**:122–127.

[20] Pellionisz, A. and Llinas, R., (1985). Tensor network theory of the metaorganization of functional geometries in the central nervous system. *Neuroscience*, **16**:245–273.

[21] von der Malsburg, C., (1990). Considerations for a visual architecture. In *Advanced Neural Computers*, R. Eckmiller, ed., pp. 303–312. Amsterdam: Elsevier Science Publ.

[22] Bruske, J. and Sommer, G., (1995). Dynamic cell structure learns perfectly topology preserving map. *Neural Computation*, **7**:845–865.

[23] J. J. Koenderink, A. K. and van Doorn, A., (1992). Local operations: The embodiment of geometry. In *Artificial and Biological Vision Systems*, G. Orban and H.-H. Nagel, eds., ESPRIT Basic Research Series, pp. 1–23. Berlin: Springer.

[24] Danielson, D. A., (1992). *Vectors and Tensors in Engineering and Physics.* Redwood City: Addison-Wesley Publ. Comp.

[25] Frankel, T., (1997). *The Geometry of Physics.* Cambridge: Cambridge University Press.

[26] Jähne, B., (1995). *Spatio-Temporal Image Processing, Theory and Scientific Applications*, Vol. 751 of *Lecture Notes in Computer Science.* Berlin: Springer.

[27] Knutsson, H., (1989). Representing local structure using tensors. In *Proc. 6th Scand. Conf. Image Analysis*, pp. 244–251. Finland: Oulu.

[28] Carlson, S., (1994). The double algebra: an effective tool for computing invariants in computer vision. In *Applications of Invariance in Computer Vision*, J. L. Mundy, A. Zisserman, and D. Forsyth, eds., Lecture Notes in Computer Science, 825, pp. 145–164. Berlin: Springer.

[29] Hestenes, D., (1986). *New Foundations for Classical Mechanics.* Dordrecht: Kluwer Acad. Publ.

[30] Porteous, I. R., (1995). *Clifford Algebra and Classical Groups.* Cambridge: Cambridge University Press.

[31] Yaglom, I. M., (1968). *Complex Numbers in Geometry.* New York: Academic Press.

[32] Blaschke, W., (1960). *Kinematik und Quaternion.* Berlin: Deutscher Verlag der Wissenschaften.

[33] McCarthy, J. M., (1990). *Introduction to Theoretical Kinematics.* Cambridge, MA: MIT Press.

[34] Murray, R. M., Li, Z., and Sastry, S. S., (1994). *A Mathematical Introduction to Robot Manipulation.* Boca Raton: CRC Press.

[35] Tweed, D., Cadera, W., and Vilis, T., (1990). Computing tree-dimensional eye position with quaternions and eye velocity from search coil signals. *Vision Research*, **30**:97–110.

[36] Hestenes, D., (1994). Invariant body kinematics: I. Saccadic and compensatory eye movements. *Neural Networks*, **7**:65–77.

[37] Hestenes, D., (1994). Invariant body kinematics: II. Reaching and neurogeometry. *Neural Networks*, **7**:79–88.

[38] Hestenes, D. and Ziegler, R., (1991). Projective geometry with Clifford algebra. *Acta Applicandae Mathematicae*, **23**:25–63.

[39] Hestenes, D., (1991). The design of linear algebra and geometry. *Acta Applicandae Mathematica*, **23**:65–93.

[40] Lasenby, J., Bayro-Corrochano, E., Lasenby, A., and Sommer, G., (1996). A new methodology for computing invariants in computer vision. In *Proc. 13th Int. Conf. on Pattern Recognition*, Vol. A, pp. 393–397. Vienna: IEEE Computer Soc. Press.

[41] Kantor, I. L. and Solodovnikov, A. S., (1989). *Hypercomplex Numbers.* New-York: Springer.

[42] Rooney, J., (1978). A comparison of representations of general spatial screw displacements. *Environment and Planning*, **B,5**:45–88.

[43] Bayro-Corrochano, E., Lasenby, J., and Sommer, G., (1996). Geometric algebra: a framework for computing point and line correspondences and projective structure using n uncalibrated cameras. In *Proc. 13th Int. Conf. on Pattern Recognition*, Vol. A, pp. 333–338. Vienna: IEEE Computer Soc. Press.

[44] Funda, J. and Paul, R. P., (1990). A computational analysis of screw transformations in robotics. *IEEE Trans. Rob. & Automation*, **6**:348–356.

[45] Rooney, J., (1977). A survey of representations of spatial rotation about a fixed point. *Environment and Planning*, **B,4**:185–210.

[46] Aspragathos, N. and Dimitros, J., (1998). A comparative study of three methods for robot kinematics. *IEEE Trans. Syst., Man, and Cybern.—Part 13*, **28**:135–145.

[47] Daniilidis, K. and Bayro-Corrochano, E., (1996). The dual-quaternion approach to hand–eye calibration. In *Proc. 13th Int. Conf. on Pattern Recognition*, Vol. A, pp. 318–322. Vienna: IEEE Computer Soc. Press.

[48] Bregler, C. and Malik, J., (1997). *Video Motion Capture.* Tech. Rpt. UCB//CSD–97, 973, Berkeley: Comp. Sci. Div., Univ. of California.

[49] Clifford, W. K., (1873). Preliminary sketch of bi-quaternions. *Proc. London Math. Soc.*, **4**:381–395.

[50] Bayro-Corrochano, E., Daniilidis, K., and Sommer, G., (1997). Hand–eye calibration in terms of motion of lines using geometric algebra. In *Lappeenranta: Proc. 10th Scand. Conf. on Image Analysis*, pp. 397–404.

[51] Doran, C., (1994). *Geometric Algebra and its Application to Mathematical Physics.* PhD thesis, Univ. of Cambridge.

[52] Doran, C., Hestenes, D., Sommen, F., and Acker, N. V., (1993). Lie groups as spin groups. *J. Math. Phys.*, **34**:3642–3669.

[53] Ramponi, G., (1990). Bi-impulse response design of isotropic quadratic filters. In *Proc. IEEE*, 78, pp. 665–677.

[54] Schetzen, M., (1981). Nonlinear system modeling based on the Wiener theory. *Proc. IEEE*, pp. 1557–1573.

[55] Sicuranza, G. L., (1992). Quadratic filters for signal processing. *Proc. IEEE*, pp. 1263–1285.

[56] Zetzsche, C. and Barth, E., (1990). Fundamental limits of linear filters in the visual processing of two–dimensional signals. *Vision Research*, **30**: 1111–1117.

[57] Krieger, G. and Zetzsche, C., (1996). Nonlinear image operators for the evaluation of local intrinsic dimensionality. *IEEE Trans. Image Process.*, **5**:1026–1042.

[58] Bülow, T. and Sommer, G., (1997). Algebraically extended representations of multidimensional signals. In *Proc. 10th Scand. Conf. on Image Analysis*, pp. 559–566. Lappeenranta.

[59] Bülow, T. and Sommer, G., (1997). Das Konzept einer zweidimensionalen Phase unter Verwendung einer algebraisch erweiterten Signalrepräsentation. In *Mustererkennung 1997*, E. Paulus and B. F. Wahl, eds., pp. 351–358.

[60] Bülow, T. and Sommer, G., (1998). A new approach to the 2–D analytic signal. In *DAGM'98.* submitted.

[61] Hahn, S. L., (1992). Multidimensional complex signals with single-orthant spectra. *Proc. IEEE*, pp. 1287–1300.

[62] Hahn, S. L., (1996). *Hilbert Transforms in Signal Processing.* Boston: Artech House.

[63] Bülow, T. and Sommer, G., (1997). Multi-dimensional signal processing using an algebraically extended signal representation. In *Algebraic Frames for the Perception-Action Cycle*, G. Sommer and J. J. Koenderink, eds., Lecture Notes in Computer Science, 1315, pp. 148–163. Berlin, Heidelberg: Springer.

[64] Bayro-Corrochano, E. and Buchholz, S., (1997). Geometric neural networks. In *Algebraic Frames of the Perception–Action Cycle*, G. Sommer and J. J. Koenderink, eds., Lecture Notes in Computer Science, 1315, pp. 3l79–394. Berlin, Heidelberg: Springer.

[65] Buchholz, S., (1997). *Algebraische Einbettung Neuronaler Netze.* Diploma thesis, Cognitive Systems Group, Inst. of Comp. Sci., Univ. of Kiel, Germany.

[66] Arena, P., Fortuna, L., Muscato, G., and Xibilia, M. G., (1996). Multilayer perceptrons to approximate quaternion valued functions. *Neural Networks*, **9**:1–8.

[67] Georgiou, G. and Koutsougeras, C., (1992). Complex domain backpropagation. *IEEE Trans. Circ. and Syst. II*, **39**:330–334.

[68] Nitta, T., (1997). An extension of the back-propagation algorithm to complex numbers. *Neural Networks*, **10**:1391–1415.

[69] Pearson, J. K., (1994). *Clifford Networks.* PhD thesis, Univ. of Kent.

[70] Bayro-Corrochano, E., Buchholz, S., and Sommer, G., (1996). Selforganizing Clifford neural network using geometric algebra. In *Proc. Int. Conf. Neural Networks, Washington, DC*, pp. 120–125.