Robot Vision in the Language of Geometric Algebra

Gerald Sommer and Christian Gebken
*Department of Computer Science, Christian-Albrechts-University, Kiel
Germany*

1. Introduction

In recent years, robot vision became an attractive scientific discipline. From a technological point of view, its aim is to endow robots with visual capabilities comparable to those of human beings. Although there is considerable endeavour, the progress is only slowly proceeding, especially in comparison to the level of behavior of human beings in natural environments. This has its reason in lacking insight into the organization principles of cognitive systems. Therefore, from a scientific point of view, robot vision is a test bed for understanding more on cognitive architectures and the mutual support of vision and action in cognitive systems. While in natural systems self-organization of structures and data flow is responsible for their success, in case of technical systems, the designer has to model cognitive systems. Modeling needs a theoretical base which is rooted in the state-of-art knowledge in science, mathematics and engineering.

The most difficult problem to be solved is the design of a useful cognitive architecture. This concerns e.g. the gathering and use of world knowledge, controlling the interplay of perception and action, the representation of equivalence classes, invariants and concepts. Besides, hard real-time requirements have to be considered. The most attractive approach to the design of a cognitive architecture is the framework of behavior-based systems (Sommer, 1997). A behavior is represented by a perception-action cycle. Remarkable features of such architecture are the tight coupling of perception and action, and learning the required competences (Pauli, 2001) from experience.

Another problem to be coped with in designing robot vision systems is the diversity of contributing disciplines. These are signal theory and image processing, pattern recognition including learning theory, robotics, computer vision and computing science. Because these disciplines developed separately, they are using different mathematical languages as modeling frameworks. Besides, their modeling capabilities are limited. These limitations are caused to a large extend by the dominant use of vector algebra. Fortunately, geometric algebras (GA) as the geometrically interpreted version of Clifford algebras (CA) (Hestenes & Sobczyk, 1984) deliver a reasonable alternative to vector algebra.

The aim of this contribution is to promote the use of geometric algebra in robot vision systems based on own successful experience over one decade of research. The application of GA within a behavior based design of cognitive systems is the long-term research topic of the Kiel Cognitive Systems Group (Sommer, 1999). Such a coherent system has to be an

Source: Vision Systems: Applications, ISBN 978-3-902613-01-1
Edited by: Goro Obinata and Ashish Dutta, pp. 608, I-Tech, Vienna, Austria, June 2007

embodiment of the geometry and the stochastic nature of the external world. That is, it should enable both internal processes converging at reasonable interpretations of the world and performing useful actions in the environment. We will report on some novel results achieved within the last years which extend the survey papers (Sommer, 2004; Sommer, 2005).

Our main contributions to applications of geometric algebra in robot vision are focussing on the following problems:

- Development of a signal theory for local analysis of multi-dimensional signals (Sommer & Zang, 2005)
- Formulation of computer vision in the framework of conformal geometry (Rosenhahn & Sommer, 2005a and 2005b)
- Knowledge based neural learning by using algebraic constraints (Buchholz & Sommer, 2006)
- Higher-order statistics (Buchholz & Le Bihan, 2006) and estimations (Perwass et al., 2006) in GA.

More details of the results contributed by the Kiel Cognitive Systems Group can be found in the publications and reports on the website http://www.ks.informatik. uni-kiel.de. Here we will report from an engineer's point of view. But the reader should be aware that GA constitutes a framework which has to be adapted to the problem at hand. Therefore, the system designer has to shape this mathematical language in a task related manner. This is both a challenge and a chance at the same time.

In section 2, we will present a bird's eye view on geometric algebra and will also motivate its use in robot vision. Special emphasis will be on the conformal geometric algebra (CGA). A novel approach to local image analysis based on embedding the curvature tensor of differential geometry into a Clifford analysis setting will be presented in section 3. Sections 4 and 5 are dedicated to our recent progress on estimations from uncertain data in CGA. We will handle uncertainty for geometric entities and kinematic operations as well. Parameter estimation methods, based on the principle of least squares adjustment, will be used for evaluating multi-vectors and their respective uncertainties. Also, in section 5 we will focus on the problem of pose estimation in case of uncertain omnidirectional vision. In addition, we will present a novel generalized camera model, the so-called inversion camera model. Again, we will take advantage of the representation power of CGA.

2. A Bird's-eye View on Geometric Algebra

In this section we will sketch the basic features of a geometric algebra representation and compare it with a vector space representation. Special emphasis is laid on the conformal geometric algebra. In addition, we introduce the key ideas of the tensor notation of GA representations and the coupling of the conformal embedding and stochastic concepts.

2.1 Comparison of Vector Algebra and Geometric Algebra

As mentioned in the introduction, the limited modeling capabilities within the disciplines contributing to robot vision are caused to a large extend by the use of vector algebra. That statement has to be justified. First, a vector space is a completely unstructured algebraic framework whose entities, that is the vectors, are directed numbers. This is a richer representation than having only scalars at hand. But the product of vectors, the scalar product, destroys the direction information originally represented in the pair of vectors by mapping them to a scalar. Second, we are mostly interested in vector spaces with Euclidean norm. The basic geometric entities of Euclidean spaces are points. A Euclidean vector space can thus be interpreted as an infinite set of points. There is no possibility of formulating useful subspace concepts in the vector space but set based ones. Third, a cognitive system is reasoning and acting on global geometric entities, like a tea pot. It makes no sense to decompose the world phenomena into point-like entities. Fourth, the most important transformation in robot vision, that is rigid body motion (RBM), has no linear representation in Euclidian space. Instead, if we are interested in describing RBM of points, we have to take advantages of an algebraic trick as extending the dimension of the space for remaining in terms of linear operations. There is no general way for generalizing this trick within the vector space concept to other geometric entities (as a pair of points or a line). Therefore, most of the basic disciplines of robot vision are getting stuck in non-linearities. The resulting iterative solutions are intractable in real-time applications. Finally, besides translation, all other operational entities acting on a vector are not itself elements of the algebra. This makes the description of actions based on certain transformation groups a difficult task.

Geometric algebra enables to overcome most of those problems, at least to a certain extend. In fact, if not specified, the term geometric algebra represents a whole family of geometric algebras. The designer has to select the right one for the problem at hand or has to design a special one with the desired features. Hence, its use enables a knowledge based system design in an algebraic framework which can represent the geometry of interest. Representing geometry in an algebraic framework means thinking in a Kleinian sense (Brannan et al., 1999). Any GA has the following features:

1. It is a linear space, which can be mapped to a vector space again. Its elements are multivectors, that is directed numbers of mixed grade. It has a rich subspace structure with each subspace having algebraic properties and interpretations in a geometric or operational sense of representing entities of a certain grade, e.g. of higher order.
2. It represents a geometry of interest. That means, it models geometric spaces equipped with basic geometric entities and a range of higher order geometric entities with useful algebraic properties. Besides, it represents a Clifford group the elements of which are linear operational entities. This makes non-linear operations in vector spaces to linear ones in the chosen GA. That is, both geometric and operational entities are elements of the algebra.
3. A geometric algebra is equipped with a geometric product the action of which on multivectors not only enables mappings into certain subspaces but from which also incidence algebraic operations between subspaces can be derived.

This as a whole makes GA a powerful tool for modeling in robot vision and beyond.

2.2 Basic Structure of Geometric Algebra

Here we will only present a sketch of the rich structure represented by a geometric algebra. For more details see (Hestenes & Sobczyk, 1984) or the introduction paper (Hestenes et al., 2001), respectively the tutorial report (Perwass & Hildenbrand, 2003).

A geometric algebra $\mathbb{R}_{p,q,r}$ is a linear space of dimension 2^n constructed from a vector space $\mathbb{R}^{p,q,r}$ with signature (p,q, r), $n = p+q+r$, which we denote $\mathbb{R}_{p,q,r} = \mathbb{G}(\mathbb{R}^{p,q,r})$. The algebra is built by applying the geometric product to the basis vectors $\mathbf{e}_i$ of $\mathbb{R}_{p,q,r}$, $i = 1, \ldots ,n$,

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

The GA $\mathbb{R}_{p,q,r}$ is called Euclidean for $n = p$ and pseudo-Euclidean for $n = p + q$. In the case of $r \neq 0$, its metric is degenerate. The signature (p,q, r) is the key for selecting certain geometric properties of the GA. The geometric product is linear and associative but not commutative. The linear space of a GA is split into a rich subspace structure represented by a set of blades $\boldsymbol{B}_k$ of grade k. Given k independent vectors[1] a_i, $i = 1, \ldots, k$, a k-blade is defined for $k = 1,\ldots,n$ by

$$\boldsymbol{B}_k = \boldsymbol{a}_1 \, \boldsymbol{a}_2 \ldots \boldsymbol{a}_k = \boldsymbol{a}_1 \wedge \boldsymbol{a}_2 \wedge \ldots \wedge \boldsymbol{a}_k \tag{2}$$

Here $(\wedge)$ indicates the outer product. There are $l_k = \binom{n}{k}$ different k-blades, each having its own direction given by $I_k = \mathbf{e}_{i_1} \wedge \mathbf{e}_{i_2} \wedge \ldots \wedge \mathbf{e}_{i_k}$. Hence, k-blades constitute directed linear subspaces of $\mathbb{R}_{p,q,r}$. In figure 1 we visualize the blade structure of $\mathbb{R}_3$, that is the GA of $\mathbb{R}^3$. By considering next the simple example of the geometric product of two vectors a,b $\in \mathbb{R}_{p,q,r}$ we will get an inductive access to the construction rule of multi-vectors as the algebraic entities of a geometric algebra.

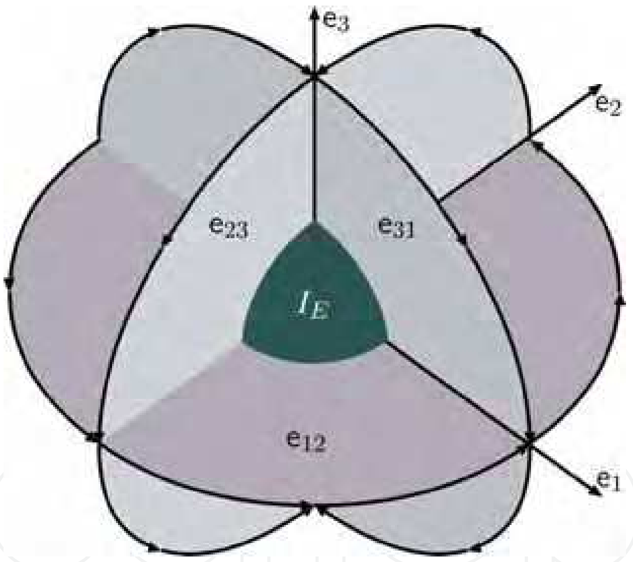

Figure 1. Blade structure of $\mathbb{R}_3$

Here $\langle \cdot \rangle_k$ means the grade-operator which indicates the separation of the linear space $\mathbb{R}_{p,q,r}$ into grade-k entities. Obviously, vectors are of grade one and $\langle \mathbb{R}_{p,q,r} \rangle_1 \equiv \mathbb{R}^{p,q,r}$. Then we get with

$$\boldsymbol{A} = \boldsymbol{a}\,\boldsymbol{b} = \boldsymbol{a} \cdot \boldsymbol{b} + \boldsymbol{a} \wedge \boldsymbol{b} \tag{3}$$

[1] We use lower case letters, as , for algebra vectors or for vector space elements.

a separation of the geometric product into the sum of the inner product

$$\boldsymbol{a} \cdot \boldsymbol{b} = \frac{1}{2}(\boldsymbol{a}\,\boldsymbol{b} + \boldsymbol{b}\,\boldsymbol{a}) = \langle \boldsymbol{a}\,\boldsymbol{b} \rangle_0 \tag{4}$$

and the outer product

$$\boldsymbol{a} \wedge \boldsymbol{b} = \frac{1}{2}(\boldsymbol{a}\,\boldsymbol{b} - \boldsymbol{b}\,\boldsymbol{a}) = \langle \boldsymbol{a}\,\boldsymbol{b} \rangle_2 \,. \tag{5}$$

The geometric product, ***a b***, results in the sum of a scalar, $(\boldsymbol{ab})_0$, and a bivector, $(\boldsymbol{ab})_2$.
In contrast to the scalar product of vector algebra, the geometric product of geometric algebra is both grade-decreasing and grade-increasing. In general the multi-vector ***A*** is a mixture of *k*-vectors, $\boldsymbol{A}_k$,

$$\boldsymbol{A} = \sum_{k=0}^{n} \boldsymbol{A}_k \tag{6}$$

with

$$\boldsymbol{A}_k = \langle \boldsymbol{A} \rangle_k = \sum_{j=1}^{l^*} \beta \boldsymbol{B}_{kj}, \tag{7}$$

$l^* \leq l_k$. For the geometric product of homogeneous multi-vectors of grades *s* and *r* we get a multi-vector **C** with a certain spectrum of different *k*-vectors,

$$\boldsymbol{C} = \boldsymbol{A}_r \boldsymbol{B}_s = \langle \boldsymbol{A}_r \boldsymbol{B}_s \rangle_{|r-s|} + \langle \boldsymbol{A}_r \boldsymbol{B}_s \rangle_{|r-s|+2} + \ldots + \langle \boldsymbol{A}_r \boldsymbol{B}_s \rangle_{r+s} \tag{8}$$

with the pure inner product $\boldsymbol{A}_r \cdot \boldsymbol{B}_s$ = $(\boldsymbol{A}_r\boldsymbol{B}_s)_{|\mathrm{r-s}|}$ and the pure outer product $\boldsymbol{A}_r \wedge \boldsymbol{B}_s$ = $(\boldsymbol{A}_r\boldsymbol{B}_s)_{r+s}$. Hence, the other components result from mixing the inner and outer product.
The blades of grade n are called pseudoscalar, ***P*** ,

$$\boldsymbol{P} = \lambda \boldsymbol{I} \tag{9}$$

with ***I*** being the unit pseudoscalar with $\boldsymbol{I}^2 = \pm 1$ if *r* = 0 and λ being a scalar which equals the determinant of matrix algebra. Because $\boldsymbol{I} = \boldsymbol{I}_k\boldsymbol{I}_{n-k}$, a blade $\boldsymbol{B}_k$ is related to its dual one, $\boldsymbol{B}_{n-k}$, by

$$\boldsymbol{B}_k^* = \boldsymbol{B}_{n-k} = \boldsymbol{B}_k \boldsymbol{I}^{-1} \tag{10}$$

This is a useful operation for switching between different representations of a multi-vector.
There are several main algebra involutions in GA, like in case of complex numbers the only existing one is conjugation. Let us mention as an example the reversion. If $\boldsymbol{A}_k \in \langle \mathbb{R}_{p,q} \rangle_k$ is a *k*-vector, then its reverse is defined as

$$\widetilde{\boldsymbol{A}}_k = \boldsymbol{a}_k \wedge \boldsymbol{a}_{k-1} \wedge \ldots \wedge \boldsymbol{a}_1 \tag{11}$$

and the reserve of a multi-vector $\boldsymbol{A} \in \mathbb{R}_{p,q}$ defined as

$$\widetilde{\boldsymbol{A}} = \sum_{k=0}^{n} (-1)^{\frac{k(k-1)}{2}} \boldsymbol{A}_k. \tag{12}$$

The reverse of a *k*-vector is needed for computing its magnitude,

$$|\boldsymbol{A}_k| = \sqrt{\boldsymbol{A}_k \cdot \widetilde{\boldsymbol{A}}_k} \tag{13}$$

and its inverse,

$$\boldsymbol{A}_k^{-1} = \frac{\widetilde{\boldsymbol{A}}_k}{|\boldsymbol{A}_k|^2} \tag{14}$$

Besides, it should be mentioned that any GA may be decomposed by

$$\mathbb{R}_n = \mathbb{R}_n^- + \mathbb{R}_n^+ \tag{15}$$

into two partial spaces with $\mathbb{R}_n$ representing the odd grade blades and $\mathbb{R}_n^+$ representing the even grade blades and $\mathbb{R}_n^-$ being a GA itself again.

There exist several isomorphisms of algebras. The most important statement is the existence of a certain matrix algebra for every GA (Porteous, 1995). In addition, the following isomorphisms are of practical importance:

$$\mathbb{R}_{p+1,q} \simeq \mathbb{R}_{q+1,p} \tag{16}$$

and

$$\mathbb{R}_{p,q}^+ \simeq \mathbb{R}_{q,p-1}. \tag{17}$$

Examples of the last one are $\mathbb{C} \simeq \mathbb{R}_{0,1} \simeq \mathbb{R}_{2,0}^+$ and $\mathbb{H} \simeq \mathbb{R}_{0,2} \simeq \mathbb{R}_{3,0}^+$ with $\mathbb{C}$ being the algebra of complex numbers and $\mathbb{H}$ being the quaternion algebra.

2.3 Geometric Algebra and its Tensor Notation

We take a look beyond the symbolic level and question how we can realize the structure of geometric algebra numerically. We show a way that makes direct use of the tensor representation inherent in GA.

If $\{\boldsymbol{E}_{1..2^n}\}$ denotes the 2^n -dimensional algebra basis of $\mathbb{R}_n$, then a multi-vector $\boldsymbol{A} \in \mathbb{R}_n$ can be written as $\boldsymbol{A} = \mathsf{a}^i \boldsymbol{E}_i$, where a^i denotes the i^{th} component of a vector[2] $\mathsf{a} \in \mathbb{R}^{2^n}$ and a sum over the repeated index *i* is implied. We use this Einstein summation convention also in the following. If $\boldsymbol{B} = \mathsf{b}^i \boldsymbol{E}_i$ and $\boldsymbol{C} = \mathsf{c}^i \boldsymbol{E}_i$, then the components of $\boldsymbol{C}$ in the algebra equation $\boldsymbol{C} = \boldsymbol{A} \circ \boldsymbol{B}$ can be evaluated via $\mathsf{c}^k = \mathsf{a}^i\, \mathsf{b}^j\, \mathsf{G}^k{}_{ij}$. Here $\circ$ is a placeholder for the algebra product and $\mathsf{G}^k{}_{ij} \in \mathbb{R}^{2^n \times 2^n \times 2^n}$ is a tensor encoding this product (we use sans serif letters as a, g or G to denote vectors, matrices, tensors or generally any regular arrangement of numbers). If we define the matrices $\mathsf{U}, \mathsf{V} \in \mathbb{R}^{2^n \times 2^n}$, as $\mathsf{U}(\mathsf{a}) := \mathsf{a}^i\, \mathsf{G}^k{}_{ij}$ and $\mathsf{V}(\mathsf{b}) := \mathsf{b}^j\, \mathsf{G}^k{}_{ij}$, then $\mathsf{c} = \mathsf{U}(\mathsf{a})\, \mathsf{b} = \mathsf{V}(\mathsf{b})\, \mathsf{a}$. This perfectly reveals the bilinearity of algebra products.

We define a mapping Φ and can then write $\Phi(\boldsymbol{A}) = \mathsf{a}$, $\Phi(\boldsymbol{A}\circ) = \mathsf{U}$, $\Phi(\boldsymbol{A} \circ \boldsymbol{B}) = \mathsf{a}^i\, \mathsf{b}^j\, \mathsf{G}^k{}_{ij}$ or if $\boldsymbol{a} = \mathsf{a}^i \mathbf{e}_i$ is an element of a Euclidian vector space, $\Phi(\boldsymbol{a}) = \mathsf{a}$ as well. Note that we reduce the complexity of equations considerably by only mapping those components of multi-vectors that are actually needed. As an example, a vector in $\mathbb{R}_n$ can have at least *n* non-zero components. Also, the outer product of two vectors will not produce 3-vector components, which can thus be disregarded. In the following we assume that Φ maps to the minimum number of components necessary.

2.4 Conformal Geometric Algebra

Recently it has been shown (Rosenhahn & Sommer, 2005a and 2005b) that the conformal geometry (Needham, 1997) is very attractive for most of the problems in robot vision, which

[2] At least numerically, there is no other way than representing multi-vectors as vectors.

are related to shape modeling, projective geometry and kinematic. Conformal geometric algebra (CGA) delivers a non-linear representation of a Euclidean space with remarkable features:
First, CGA constitutes a unique framework for affine, projective and Euclidean geometry. Because the special Euclidean transformation (RBM) is a special affine transformation, we can handle either kinematic, projective or metric aspects of the problem at hand in the same algebraic frame. Second, the basic geometric entities of conformal geometry are spheres of dimension *n*. Other geometric entities as points, planes, lines, circles,... may be easily constructed. These entities are no longer set concepts of a vector space but elements of CGA. Third, the special Euclidean group is a subgroup of the conformal group, which is in CGA an orthogonal group. Therefore, its action on the above mentioned geometric entities will be a linear operation. Fourth, the inversion operation is another subgroup of the conformal group which can be advantageously used in robot vision. Fifth, CGA generalizes the incidence algebra of projective geometry with respect to the above mentioned geometric entities.
Before we enlighten the structure and features of CGA in more detail, we will have a short look on $\mathbb{R}_3$, the geometric algebra of the Euclidean 3D-space $\mathbb{R}^3$. This will be the starting point for the mentioned non-linear representation in CGA. Additionally, $\mathbb{R}_3$ is the embedding framework for image analysis, which will be described in section 3. The basis of its 8- dimensional space is given by

$$\mathrm{basis}(\mathbb{R}_3) : \{\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_{23}, \mathbf{e}_{31}, \mathbf{e}_{12}, \mathbf{e}_{123}\} \tag{18}$$

with $\mathbf{e}_0 \equiv 1$ and $\mathbf{e}_i$ being the basis vectors of $\mathbb{R}^3$ with $\mathbf{e}_0^2 = \mathbf{e}_i^2 = 1$. Here the $\mathbf{e}_i$ constitute the unit 1-blades and the $\mathbf{e}_{ij}$ constitute the unit 2-blades with $\mathbf{e}_{ij}^2 = -1$, see figure 1. The unit pseudoscalar $\mathbf{e}_{123} \equiv I_E$ squares according to $\mathbf{e}_{123}^2 = -1$.
The even subalgebra $\mathbb{R}_3^+$ is isomorphic to the quaternion algebra $\mathbb{H}$ according to equation (17). Its dimension is four and the basis is given by

$$\mathrm{basis}(\mathbb{R}_3^+) : \{\mathbf{e}_0, \mathbf{e}_{23}, \mathbf{e}_{31}, \mathbf{e}_{12}\} \simeq \{1, i, j, -k\}\,, \tag{19}$$

where *i*, *j*, *k* are the imaginary unit vectors of a quaternion.
The conformal geometric algebra $\mathbb{R}_{4,1}$ of $\mathbb{R}^3$, is built by extending $\mathbb{R}^3$ with a so-called Minkowski plane $\mathbb{R}^{1,1}$, resulting in $\mathbb{R}^{4,1}$. Originally, this construction of the CGA of a pseudo-Euclidean space $\mathbb{R}^{p,q}$ which results in $\mathbb{R}_{p+1,q+1}$, was proposed and analyzed by (Angles, 1980). Only the work of (Li et al., 2001a) has been recognized by the robot vision community as valuable access to the interesting phenomena in a unique framework. The same authors presented also a CGA for spherical geometry (Li et al., 2001b) and a further generalization to cope with Euclidean, spherical and hyperbolic geometry (Li et al., 2001c). But the last two cases have not yet been studied in robot vision.
The basis of the Euclidean CGA $\mathbb{R}_{4,1}$, is of dimension 32. That one of the extended space $\mathbb{R}^{4,1}$ contains as additional basis vectors $\mathbf{e}_+$ and $\mathbf{e}_-$ with $\mathbf{e}_+^2 = 1$, $\mathbf{e}_-^2 = -1$, $\mathbf{e}_+ \cdot \mathbf{e}_- = 0$. Both basis vectors constitute the so-called orthonormal basis of the Minkowski plane. More attractive is to switch to the so-called null-basis $\{\mathbf{e}_o, \mathbf{e}_\infty\}$ with $\mathbf{e}_\infty^2 = \mathbf{e}_o^2 = 0$ and $\mathbf{e}_\infty \cdot \mathbf{e}_o = -1$. This has two reasons. First, both the origin of $\mathbb{R}^3$, represented by $\mathbf{e}_o = \frac{1}{2}(\mathbf{e}_- - \mathbf{e}_+)$, and the point at infinity, represented by $\mathbf{e}_\infty = \mathbf{e}_- + \mathbf{e}_+$, are explicitly accessible. Second, a point *x* 4,1 of the Euclidian 3D-vector space $\mathbb{R}^3$ is mapped to a

conformal point (null vector) $\boldsymbol{X} \in \mathbb{R}^{4,1}$, with $\boldsymbol{X}^2 = 0$ and $\boldsymbol{X} \cdot \mathbf{e}_\infty = -1$, by the embedding function

$$\mathcal{K} : \boldsymbol{x} \longmapsto \boldsymbol{X} := \boldsymbol{x} + \tfrac{1}{2}\,\boldsymbol{x}^2\,\mathbf{e}_\infty + \mathbf{e}_o\,. \tag{20}$$

We denote these special vectors by capital letters as well. The mapping $\mathcal{K}$ builds a homogeneous representation of a stereographically projected point (Rosenhahn & Sommer, 2005a). As a grade-1 entity, a point is a special sphere, *S*, (also of grade one) with radius zero. The dual representation of a sphere

$$\boldsymbol{S}^* = \boldsymbol{A} \wedge \boldsymbol{B} \wedge \boldsymbol{C} \wedge \boldsymbol{D} \tag{21}$$

is of grade four and is defined by the outer product of four points. A circle as a 2-dimensional sphere, $\boldsymbol{Z} \in \langle \mathbb{R}_{4,1} \rangle_2$ or $\boldsymbol{Z}^* \in \langle \mathbb{R}_{4,1} \rangle_3$ is defined by

$$\boldsymbol{Z} = \boldsymbol{S}_1 \wedge \boldsymbol{S}_2\,, \quad \boldsymbol{Z}^* = \boldsymbol{A} \wedge \boldsymbol{B} \wedge \boldsymbol{C}\,. \tag{22}$$

By replacing one point in the defining equations (21) or (22) by the point at infinity, $\mathbf{e}_\infty$, a plane, a line or a point pair (a one-dimensional sphere) may be derived. Most interesting for robot vision is the orthogonal representation in $\mathbb{R}_{4,1}$ of the elements of the conformal group *C*(3). All transformations belonging to the conformal group are linear ones and the null cone, that is the set of all null vectors, is invariant with respect to them. Let $\boldsymbol{G} \in \mathbb{R}_{4,1}$ be an element of the conformal group and $\boldsymbol{U} \in \mathbb{R}_{4,1}$ any entity which has to be transformed by $\boldsymbol{G}$ to $\boldsymbol{U}' \in \mathbb{R}_{4,1}$. Then

$$\boldsymbol{U}' \;=\; \boldsymbol{G}\,\boldsymbol{U}\,\widetilde{\boldsymbol{G}} \tag{23}$$

describes this transformation as a (bi-)linear mapping. In general, all algebraic entities with such sandwich product are called versors (Hestens et al., 2001). Given some conditions, certain versors are called spinors (representing rotation and dilation) and normalized spinors are called rotors (representing pure rotation). Interestingly, also translation has a rotor representation (called translator) in CGA. But the most interesting transformation belonging to the conformal group is inversion, see (Needham, 1997), because all other transformations can be derived from it. Let $\boldsymbol{S} = \mathbf{e}_o - \frac{1}{2}\mathbf{e}_\infty = -\mathbf{e}_+$ be a unit sphere located at the origin $\mathbf{e}_o$, then the inversion of any conformal point $\boldsymbol{X} \in \langle \mathbb{R}_{4,1} \rangle_1$ in the unit sphere is written

$$\boldsymbol{X}' = \boldsymbol{S}\,\boldsymbol{X}\,\boldsymbol{S}\,. \tag{24}$$

The elements of the rigid body motion in CGA are called motors, $\boldsymbol{M} \in \mathbb{R}^+_{4,1}$. They connect rotation, represented by a rotor ***R***, and translation, represented by a translator ***T***, in a multiplicative way,

$$\boldsymbol{M} = \;\boldsymbol{T}\;\boldsymbol{R}\;\widetilde{\boldsymbol{T}} \tag{25}$$

and can be interpreted as a general rotation (Rosenhahn & Sommer, 2005a). As all versors, they are concatenated multiplicatively. Let $\boldsymbol{M} = \boldsymbol{M}_2\boldsymbol{M}_1$ be a sequence of two motors, then

$$\boldsymbol{U}'' = \boldsymbol{M}\,\boldsymbol{U}\,\widetilde{\boldsymbol{M}} \;=\; \boldsymbol{M}_2\,\boldsymbol{U}'\,\widetilde{\boldsymbol{M}}_2 \;=\; \boldsymbol{M}_2\,\boldsymbol{M}_1\,\boldsymbol{U}\,\widetilde{\boldsymbol{M}}_1\,\widetilde{\boldsymbol{M}}_2 \tag{26}$$

for all $\boldsymbol{U} \in \mathbb{R}_{4,1}$. Another important feature of linear operations in GA also applies for versors in CGA. It is the preservation of the outer product under linear transformation, which is called outermorphism (Heestens, 1991). Let $\boldsymbol{S}_1, \boldsymbol{S}_2 \in \langle \mathbb{R}_{4,1} \rangle_1$ be two spheres and $\boldsymbol{Z} \in \langle \mathbb{R}_{4,1} \rangle_2$ a circle. Then according to equations (22) and (23) the circle transforms under the action of a motor $\boldsymbol{M} \in \mathbb{R}_{4,1}^{+}$ as

$$\begin{aligned} \boldsymbol{Z}' &= \boldsymbol{M}\boldsymbol{Z}\widetilde{\boldsymbol{M}} = \boldsymbol{M}(\boldsymbol{S}_1 \wedge \boldsymbol{S}_2)\widetilde{\boldsymbol{M}} = \langle \boldsymbol{M}(\boldsymbol{S}_1\boldsymbol{S}_2)\widetilde{\boldsymbol{M}} \rangle_2 \\ &= \langle \boldsymbol{M}\boldsymbol{S}_1\widetilde{\boldsymbol{M}}\boldsymbol{M}\boldsymbol{S}_2\widetilde{\boldsymbol{M}} \rangle_2 = \boldsymbol{M}\boldsymbol{S}_1\widetilde{\boldsymbol{M}} \wedge \boldsymbol{M}\boldsymbol{S}_2\widetilde{\boldsymbol{M}} = \boldsymbol{S}_1' \wedge \boldsymbol{S}_2' \,. \end{aligned} \tag{27}$$

These last features of CGA turn out to be very important for robot vision applications as pose estimation, see (Rosenhahn & Sommer, 2005b) and (Gebken et al., 2006). Another important feature of CGA is the stratification of spaces according to (Faugeras, 1995) in one algebraic framework. Because

$$\mathbb{R}_{4,1} \supset \mathbb{R}_{3,1} \supset \mathbb{R}_{3,0} \tag{28}$$

with $\mathbb{R}_{3,1}$ being one possible representation of the projective space in GA, the change of the representations with the respective geometric aspects is a simple task, see (Rosenhahn & Sommer, 2005a).

2.5 Conformal Embedding - the Stochastic Supplement

We have to obey the rules of error propagation when we embed points by means of function $\mathcal{K}$, equation (20). Assume that point x is a random vector with a Gaussian distribution and $\bar{\boldsymbol{x}}$ is its mean value. Furthermore, we denote the 3×3 covariance matrix of x by Σ_x. Let $\mathcal{E}$ denote the expectation value operator, such that $\mathcal{E}[\boldsymbol{x}] = \bar{\boldsymbol{x}}$. The uncertain representative in conformal space, i.e. the stochastic supplement for $\boldsymbol{X} = \mathcal{K}(\bar{\boldsymbol{x}})$, is determined by a sphere with imaginary radius

$$\mathcal{E}\Big[\mathcal{K}(\boldsymbol{x})\Big] = \bar{\boldsymbol{x}} + \tfrac{1}{2}\,\bar{\boldsymbol{x}}^2\,\mathbf{e}_\infty + \mathbf{e}_o + \tfrac{1}{2}\mathrm{trace}\Big(\Sigma_{\boldsymbol{x}}\Big)\mathbf{e}_\infty \tag{29}$$

rather than the pure conformal point $\mathcal{K}\Big(\mathcal{E}[\boldsymbol{x}]\Big)$. However, observing that $\mathcal{K}\Big(\mathcal{E}[\boldsymbol{x}]\Big) \approx \mathcal{E}\Big[\mathcal{K}(\boldsymbol{x})\Big]$ shows why our algorithms do not noticeably differ in the output when using an exact embedding or its approximation. We evaluate the corresponding 5×5 covariance matrix $\Sigma_{\boldsymbol{X}}$ for $\boldsymbol{X} = \mathcal{K}(\bar{\boldsymbol{x}})$ by means of error propagation and find

$$\Sigma_{\boldsymbol{X}} = \mathsf{J}_{\mathcal{K}}(\bar{\boldsymbol{x}})\;\Sigma_{\boldsymbol{x}}\;\mathsf{J}_{\mathcal{K}}^{\mathsf{T}}(\bar{\boldsymbol{x}})\,, \tag{30}$$

where we used the Jacobian of $\mathcal{K}$.

$$\mathsf{J}_{\mathcal{K}}(\bar{\boldsymbol{x}}) := \frac{\partial \mathcal{K}}{\partial \boldsymbol{X}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \bar{x}_1 & \bar{x}_2 & \bar{x}_3 \\ 0 & 0 & 0 \end{bmatrix} . \tag{31}$$

3. Monogenic Curvature Tensor as Image Representation

In this section we will describe how the embedding of local image analysis into a geometric algebra extends the representation in such a way that a rich set of local features will emerge.

3.1 Overview: Local Spectral Representations

Image analysis is a central task of robot vision systems. It is to a main portion local analysis. Image analysis based on local spectral representations (Granlund & Knutsson, 1995), that is amplitude and phase, has been a well-known method of signal processing for years. The aim is to assign a structural or/and geometric interpretation to an image point. That task of computing is called split of identity. In practice, a set of oriented bandpass operators are applied, each consisting of a pair of quadrature filters. The most well-known representative is the complex valued Gabor filter (Gabor, 1946). It delivers a complex valued signal representation, the analytic signal, from which for each chosen orientation at position $\boldsymbol{x} \in \mathbb{R}^2$ a local amplitude and a local phase can be derived. The local amplitude can be considered as a confidence measure of estimates of the local parity symmetry of the signal derived from local phase. Parity symmetry is a measure, which describes the type of structure. The method can be used for detecting lines and edges, analyzing textures, and with some restrictions for detecting corners and junctions.

Regrettably, the analytic signal is neither rotation invariant nor sensitive to discriminate intrinsically 1D and 2D (i1D and i2D) structures. This has its reason in the fact that the analytic signal is indeed only a reasonable complex valued extension of one-dimensional functions. Therefore, with great endeavour the problems of orientation steerability (Freeman & Adelson, 1991) and of generalizing the Hilbert transform (Hahn, 1996) have been attacked.

Only the consequent use of Clifford analysis (Brackx et al., 1982) led us to a multi-dimensional generalization of the analytic signal, called monogenic signal (Felsberg & Sommer, 2001) which overcomes the missing rotation invariance. But also that representation is incomplete with respect to represent intrinsically 2D structures, see the survey paper (Sommer & Zang, 2007).

The monogenic curvature tensor (Zang & Sommer, 2007) further generalizes the monogenic signal. It delivers a local signal representation with the following features:

- It enables classification of intrinsic dimension.
- It delivers two curvature based signal representations which distinctly separate represent intrinsically 1D and 2D structures. One of these is identical to the monogenic signal. Two specific but comparable types of local amplitude and phase can be described.
- In both cases the local phase constitutes a vector that includes also the orientation as a geometric feature.
- In case of i2D structures, an angle of intersection can be derived from the derivations of phase angles.
- Both curvature based signal representations can be embedded in a novel scale-space concept, the monogenic scale-space (Feldsberg & Sommer, 2004), in which local amplitude, phase and orientation become inherent features of a scale-space theory. This enables scale adaptive local image analysis.

All these efforts have been made because of the advantages of phase based image analysis for getting access to geometry and because of the illumination invariance of phase information.

3.2 Monogenic Curvature Tensor

The image representation we want to model should have some invariances:

- Invariance with respect to intrinsic dimension: Both i1D and i2D structures can be modeled. This is possible by the curvature tensor of differential geometry (Koenderink & van Doorn, 1987).
- Invariance with respect to parity symmetry: Both even and odd symmetric structures can be represented. This is possible by designing quadrature phase filters, whose harmonic conjugate component is in quadrature phase relation to the real valued component (Sommer & Zang, 2007). The way to get this is applying a (generalized) holomorphic extension of a real valued multi-dimensional function by a (generalized) Hilbert transform.
- Invariance with respect to rotation: This becomes possible by specifying the generalized holomorphic extension by a monogenic extension (Felsberg & Sommer, 2001), whose operator realization is given by the Riesz transform (Stein & Weiss, 1971).
- Invariance with respect to angle of intersection: Because of the involved differential geometric model, a local structure model for i2D structures is considered for i1D structures intersecting at arbitrary angles.
- Invariance with respect to scale: This requires embedding of the image representation, respective of the operator which derives it into a monogenic scale-space (Felsberg & Sommer, 2004).

Having these invariances in the image representation, in a second step of analysis the corresponding variances can be computed. These are intrinsic dimension, parity symmetry, rotation angle, angle of intersection and intrinsic scale at which these features exist.

We will interpret a 2D-image as a surface in $\mathbb{R}^3$. Let be C the curvature tensor of the second fundamental theorem of differential geometry. Its Monge patch representation is given by

$$\mathsf{C}(\boldsymbol{x}) = \left((1 + {f_x}^2 + {f_y}^2)^{-\frac{1}{2}}\ \mathsf{B}\right)(\boldsymbol{x}) \tag{32}$$

with the Hesse matrix

$$\mathsf{B}(\boldsymbol{x}) = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix}\left(\boldsymbol{x}\right). \tag{33}$$

Then the Gaussian curvature, $\kappa(\boldsymbol{x}) = \det(\mathsf{B})$, and the mean curvature, $\mu(\boldsymbol{x}) = \mathrm{trace}(\mathsf{B})$, are spanning a basis in which the local signal *f*(*x*) can be classified according to its intrinsic dimension according to table 1.

Type	μ (Mean Curvature)	κ (Gaussian Curvature)
Elliptic (i2D)		$\kappa > 0$
Hyperbolic (i2D)		$\kappa < 0$
Parabolic (i1D)	$\lvert\mu\rvert \neq 0$	$\kappa = 0$
Planar (i0D)	$\lvert\mu\rvert = 0$	$\kappa = 0$

Table 1. Surface type classification based on Gaussian and mean curvature

The signal representation we want to get is a kind of Hesse matrix in a monogenic representation. This requires two steps. First, according to (Felsberg & Sommer, 2001) we are embedding the originally scalar valued signal *f*(*x*) as a vector field ***f***(*x*) with values directed to the unit vector $\mathbf{e}_3$,

$$\begin{aligned} f(\boldsymbol{x}) \,:\, \mathbb{R}^2 \rightarrow \mathbb{R} \quad &\longrightarrow \quad \boldsymbol{f}(\boldsymbol{x}) \,:\, \mathbb{R}^2 \rightarrow \mathbb{R}\,\mathbf{e}_3 \\ \boldsymbol{f}(\boldsymbol{x}) \;=\; \boldsymbol{f}(x\,\mathbf{e}_1 + y\,\mathbf{e}_2) \;&=\; f(x,y)\,\mathbf{e}_3\,. \end{aligned} \tag{34}$$

Second, we are switching from the vector space $\mathbb{R}^3$ to the Euclidean geometric algebra $\mathbb{R}_3 \equiv \mathbb{G}(\mathbb{R}^3)$ and are applying a monogenically extended Hessian operator, $\mathsf{h}_M \in M(2, \mathbb{R}_3)$, which is a 2×2 matrix with monogenic elements. The convolution of the signal ***f*** with all elements of the operator matrix results in the monogenic curvature tensor $\mathsf{T}(x) \in M(2, \mathbb{R}_3)$ as signal representation. To be more specific, see (Zang & Sommer, 2007), the monogenic Hessian operator may be splitted into an even operator, $\mathsf{h}_e \in M(2, \mathbb{R}_3^+)$, with spinor valued elements and an odd operator, $\mathsf{h}_o \in M(2, \mathbb{R}_3^-)$ which results from the even operator by applying the Riesz transform ***h****R*,

$$\mathsf{h}_M \;=\; \mathsf{h}_e + \mathsf{h}_o \;=\; \mathsf{h}_e \,+\, \boldsymbol{h}_R * \mathsf{h}_e \tag{35}$$

with

$$\mathsf{h}_e(\boldsymbol{x}) \;=\; \begin{bmatrix} \partial_{xx} & -\partial_{xy}\,\mathbf{e}_{12} \\ \partial_{xy}\,\mathbf{e}_{12} & \partial_{yy} \end{bmatrix} (\boldsymbol{x}) \tag{36}$$

and

$$\boldsymbol{h}_R(\boldsymbol{x}) \;=\; \frac{\boldsymbol{x}\,\mathbf{e}_3}{2\pi|\boldsymbol{x}|^3}\,. \tag{37}$$

The monogenic Hessian operator may be interpreted as a rotation invariant and parity symmetry invariant detector of two i1D structures crossing invariant with respect to the angle of intersection. This involved structure model is the most general that could be developed. Nevertheless, it is limited by the model of differential geometry which does not consider derivatives of order higher than two. The structure of the monogenic Hessian operator reveals if we are going to the Fourier domain, take advantage of the derivative theorem of Fourier theory, and are modeling the operator in terms of circular harmonics of order *n*, $\boldsymbol{C}_n \in \mathbb{R}_3^+$, in polar coordinates ***u*** = (*ρ*, *α*),

$$\boldsymbol{C}_n(\varrho, \alpha) \;=\; \boldsymbol{C}_n(\varrho)\exp(n\alpha\mathbf{e}_{12})\,. \tag{38}$$

Then we recognize that our model involves circular harmonics of orders n ∊ {0, 1, 2,3},

$$\mathsf{H}_e(\boldsymbol{u}) \;=\; \frac{1}{2}\begin{bmatrix} \boldsymbol{C}_0 + \langle \boldsymbol{C}_2 \rangle_0 & -\langle \boldsymbol{C}_2 \rangle_2 \\ \langle \boldsymbol{C}_2 \rangle_2 & \boldsymbol{C}_0 - \langle \boldsymbol{C}_2 \rangle_0 \end{bmatrix} (\boldsymbol{u}) \tag{39}$$

$$\mathsf{H}_o(\boldsymbol{u}) \;=\; \frac{1}{2}\begin{bmatrix} \boldsymbol{C}_1(\boldsymbol{C}_0 + \langle \boldsymbol{C}_2 \rangle_0) & \boldsymbol{C}_1(-\langle \boldsymbol{C}_2 \rangle_2) \\ \boldsymbol{C}_1(\langle \boldsymbol{C}_2 \rangle_2) & \boldsymbol{C}_1(\boldsymbol{C}_0 - \langle \boldsymbol{C}_2 \rangle_0) \end{bmatrix} (\boldsymbol{u})\,. \tag{40}$$

As equations (35) and (40) reveal, the Riesz transform is identic to the first order circular harmonic,

$$\boldsymbol{H}_R(\boldsymbol{u}) \;=\; \frac{\boldsymbol{u}}{|\boldsymbol{u}|}\,\mathbf{e}_{12}^{-1} \;\equiv\; \boldsymbol{C}_1(\boldsymbol{u})\,. \tag{41}$$

What remains for fulfilling the scale invariance requirement is embedding the monogenic Hessian operator into the monogenic scale-space (Felsberg & Sommer, 2004). This is achieved by replacing the radial component of circular harmonics, $\boldsymbol{C}_n(\rho)$, by a Difference-of-Poisson kernel, $\boldsymbol{H}_{\mathrm{DOP}}$,

$$\boldsymbol{H}_{\mathrm{DOP}}(\varrho; s_1, s_2) \;=\; \exp(-2\pi\varrho\, s_1) \;-\; \exp(-2\pi\varrho\, s_2) \tag{42}$$

with $s_1 < s_2$ being two different scale parameters. This results in circular harmonic bandpass functions

$$\boldsymbol{C}_n(\varrho, \alpha; s_1, s_2) \;=\; \boldsymbol{H}_{\mathrm{DOP}}(\varrho; s_1, s_2)\,\boldsymbol{C}_n(\alpha) \tag{43}$$

Finally, we get the monogenic curvature tensor $\mathsf{T}(\boldsymbol{x})$ as

$$\mathsf{T}(\boldsymbol{x}) \;=\; \mathsf{T}_e(\boldsymbol{x}) + \mathsf{T}_o(\boldsymbol{x}) \;=\; \Big((\mathsf{h}_e + \mathsf{h}_o) * \boldsymbol{f}\Big)(\boldsymbol{x})\,, \tag{44}$$

respectively its representation in frequency domain.

3.3 Analysis of the Monogenic Curvature Tensor

Having the monogenic curvature tensor (in a scale-space embedding), it will now be analyzed with respect to the represented curvature information (Zang & Sommer, 2007).

By computing the trace of $\mathsf{T}(\boldsymbol{x})$, we get the monogenic mean curvature signal, $\boldsymbol{f}_{\mathrm{i1D}}(\boldsymbol{x})$: $\mathbb{R}^2 \to \mathbb{R}_3$, which is specific with respect to i1D structures. It may be written as a vector field

$$\boldsymbol{f}_{\mathrm{i1D}}(\boldsymbol{x}) \quad=\quad \boldsymbol{t}_e(\boldsymbol{x}) + \boldsymbol{t}_o(\boldsymbol{x}) \;=\; \mathrm{trace}(\mathsf{T}_e)(\boldsymbol{x}) + \mathrm{trace}(\mathsf{T}_o)\,\mathbf{e}_2(\boldsymbol{x}) \tag{45}$$

$$=\quad \boldsymbol{f}(\boldsymbol{x}) + \Big(\boldsymbol{h}_R * f\Big)(\boldsymbol{x}) \;\equiv\; \boldsymbol{f}_M(\boldsymbol{x})\,, \tag{46}$$

which turns out to be identical to the monogenic signal (Felsberg & Sommer, 2001).

By computing the determinant of $\mathsf{T}(\boldsymbol{x})$, we get the generalized monogenic Gaussian curvature signal, $\boldsymbol{f}_{\mathrm{i2D}}(\boldsymbol{x})$: $\mathbb{R}^2 \to \mathbb{R}_3$, which is specific with respect to i2D structures. In similar way as $\boldsymbol{f}_{\mathrm{i1D}}$, it may be written as a vector field

$$\boldsymbol{f}_{\mathrm{i2d}}(\boldsymbol{x}) \quad=\quad \boldsymbol{d}_e(\boldsymbol{x}) + \boldsymbol{d}_o(\boldsymbol{x}) \;=\; \det{}_R(\mathsf{T}_e)\mathbf{e}_3(\boldsymbol{x}) + \mathbf{e}_1 \det{}_R(\mathsf{T}_o)(\boldsymbol{x}) \tag{47}$$

$$=\quad \boldsymbol{d}_e(\boldsymbol{x}) + \Big(\mathbf{e}_1\boldsymbol{c}_2\mathbf{e}_3 \,*\, \boldsymbol{d}_e\Big)(\boldsymbol{x}) \;\equiv\; \boldsymbol{f}_{\mathrm{MC}}(\boldsymbol{x})\,. \tag{48}$$

We call it ’generalized monogenic’ because its conjugate harmonic part results from the real part by applying $\boldsymbol{c}_2$ as generalized Hilbert transform with the result that the relations between $\boldsymbol{d}_{\mathrm{e}}$ and $\boldsymbol{d}_{\mathrm{o}}$ are different to those of $\boldsymbol{t}_{\mathrm{e}}$ and $\boldsymbol{t}_{\mathrm{o}}$. Both signal representations can be interpreted as the result of a spinor valued operator, $\boldsymbol{s}$, which rotates and scales the original vector field $\boldsymbol{f}(\boldsymbol{x}) \;=\; f(x, y)\,\mathbf{e}_3$ so that it will be supplemented by a conjugate harmonic component which projects to the plane $\mathbf{e}_1 \wedge \mathbf{e}_2$ and fulfills the conditions $\boldsymbol{t}_e^2 \;=\; \boldsymbol{t}_o^2$ and $\boldsymbol{d}_e^2 \;=\; \boldsymbol{d}_o^2$. The scaling-rotation is performed in the ’phase plane’ $\boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{x}) \wedge \mathbf{e}_3 \;=\; \langle \mathbf{e}_3 \boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{x})\rangle_2$

with $\boldsymbol{s}(\boldsymbol{x}) = \mathbf{e}_3 \boldsymbol{f_s}(\boldsymbol{x})$ being the respective spinor and $\boldsymbol{f_s} \equiv \boldsymbol{f}_{\mathrm{i1D}}$ or $\boldsymbol{f_s} \equiv \boldsymbol{f}_{\mathrm{i2D}}$. By evaluating the exponential representation of $\boldsymbol{s}$ with respect to the $\mathbb{R}_3^+$-logarithm, see (Felsberg, 2002), the local spectral representations can be computed. These are the local amplitude

$$a(\boldsymbol{x}) \;=\; |\boldsymbol{f_s}(\boldsymbol{x})| \;=\; \exp(\langle \log(\mathbf{e}_3 \boldsymbol{f_s}(\boldsymbol{x}))\rangle_0) \tag{49}$$

and the (generalized) monogenic local phase bivector

$$\Phi(\boldsymbol{x}) \;=\; \arg(\boldsymbol{f_s}(\boldsymbol{x})) \;=\; \langle \log(\mathbf{e}_3 \boldsymbol{f_s}(\boldsymbol{x}))\rangle_2 \,. \tag{50}$$

From $\Phi(x)$ follow the local phase $\phi(x)$ as rotation angle within the phase plane,

$$\phi(\boldsymbol{x}) \;=\; |(\Phi(\boldsymbol{x}))^\star| \;=\; \operatorname{atan}\left(\frac{|\langle \mathbf{e}_3 \boldsymbol{f_s}(\boldsymbol{x})\rangle_2|}{|\langle \mathbf{e}_3 \boldsymbol{f_s}(\boldsymbol{x})\rangle_0|} \right), \tag{51}$$

and the orientation angle θ(x) of the phase plane within the plane $\mathbf{e}_1 \wedge \mathbf{e}_2$,

$$\boldsymbol{\theta}(\boldsymbol{x}) \;=\; \frac{\langle \mathbf{e}_3 \boldsymbol{f_s}(\boldsymbol{x})\rangle_2}{|\langle \mathbf{e}_3 \boldsymbol{f_s}(\boldsymbol{x})\rangle_2|} \,. \tag{52}$$

In the case of f_{i1D}, $\boldsymbol{\theta}(x)$ is indicating the orientation of the i1D structure within the image plane and in the case of f_{i2D}, $2\boldsymbol{\theta}(x)$ represents the local main orientation of the i2D structure in a double angle representation which results from the eigenvector decomposition of the structure tensor (Felsberg, 2002). Hence, phase analysis delivers also the orientation information as a consequence of the monogenic representation of the curvature tensor.

In Figure 2, an example signal is analyzed with respect to its local spectral representations. The monogenic curvature tensor is obviously invariant with respect to rotation. In figure 3, two patterns of even and odd symmetric structures are analyzed with respect to local amplitudes and local phases for f_{i1D} and f_{i2D}, respectively. Clearly can be seen the invariances of the monogenic curvature tensor with respect to the intrinsic dimension, parity symmetry and angle of intersection.

We will not discuss in detail the scale-space properties (Zang & Sommer, 2006a). It should only be mentioned that the embedding of the curvature tensor into a monogenic scale-space results in an improved corner detection based on a novel two-dimensional phase congruency method (Zang & Sommer, 2006b) and delivers superior estimates of the optical flow field based on a phase constrained variational approach (Zang et al., 2007).

4. Parameter Estimation from Uncertain Data

Uncertain data occurs almost invariably, especially in computer vision applications. It is hence a necessity to develop and use methods, which account for the errors in observational data. Here, we discuss a parameter estimation from uncertain data in the unified mathematical framework of geometric algebra.

We use conformal geometric algebra (CGA) as introduced in section 2.4. Consequently, the estimation is applicable to (parameterizations of) geometric entities and geometric operators; points, lines, planes, circles or spheres can be treated in very much the same way as rotations or rigid body motions (RBM). In general, our aim is to find multi-vectors that satisfy a particular condition equation, which depends on a set of uncertain measurements. The specific problem and the type of multi-vector, representing a geometric entity or a geometric operator, determine the condition. In the language of CGA we obtain succinct expressions and thanks to the bilinearity of the always involved geometric product, the corresponding equations are linear or at most quadratic in the multi-vector components. In section 2.3 we have introduced a simple way to represent geometric algebra operations in terms of a tensor notation, where the term tensor denotes the classical extension of matrix theory to higher dimensions. This allows us to use well-tried and efficient algorithms without leaving the algebra. Moreover, it paves the way for using the stochastic: standard error propagation, for example, is exact for the geometric product and makes it easily possible to keep track of the uncertainties while doing operations like an intersection.

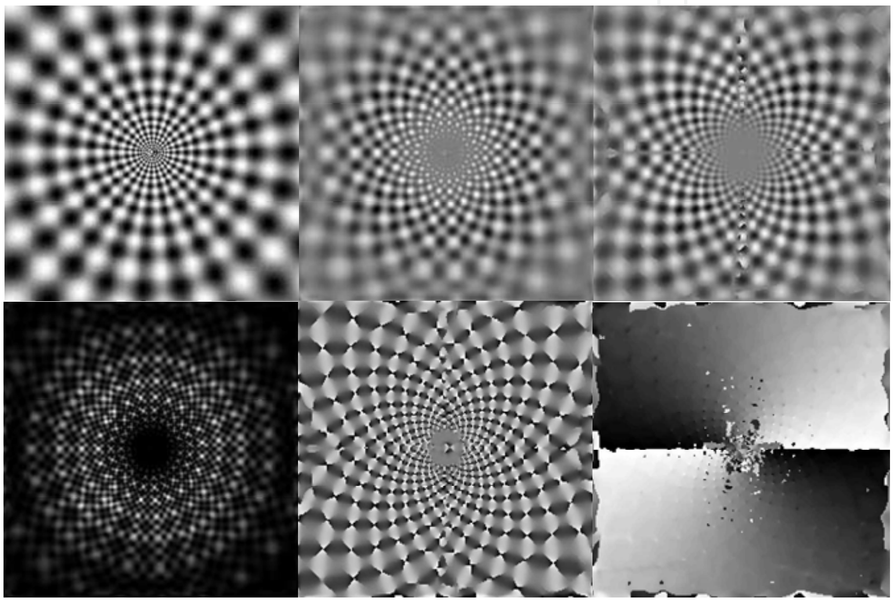

Figure 2. Top: original image (left), even and odd components of f_{i2D} (middle and right). Bottom: local amplitude (left), local phase (middle) and local orientation (right)

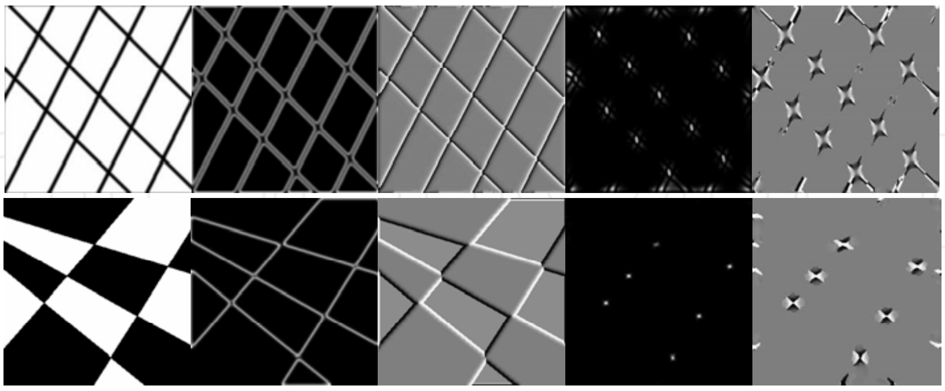

Figure 3. From left to right: original images, local amplitudes and local phases of the monogenic signal f_{i1D}, local amplitudes and local phases of the generalized monogenic Gaussian curvature signal f_{i2D}

The stochastic is one of the fundamental aspects of this section. To account for the uncertainties in observational data we consequently decided on a least squares adjustment parameter estimation. We use the Gauss-Markov and the Gauss-Helmert method. Each of them provides an estimate together with a suitable covariance matrix. Hence, further calculations can be carried out with these uncertain elements, as mentioned above.

This text builds on previous works by (Heuel, 2004) where uncertain points, lines and planes were treated in a unified manner, but not in GA. The linear estimation of rotation operators in GA was previously discussed in (Perwass & Sommer, 2002), albeit without taking account of uncertainty. In (Perwass et al., 2005) the estimation of uncertain general operators was introduced.

The structure of this section is as follows: first, we explain the underlying parameter estimation methods. We then present two applications. For each, we demonstrate in which way we profit from the expressiveness of CGA and we explain how our method can be applied within that framework.

4.1 Stochastic Estimation Method

In the field of parameter estimation one usually parameterizes some physical process $\mathcal{P}$ in terms of a model $\mathcal{M}$ and a suitable parameter vector p. The components of p are then to be estimated from a set of observations originating from $\mathcal{P}$.

Here, we introduce our two parameter estimation methods, the common Gauss-Markov method and the most generalized case of least squares adjustment, the Gauss-Helmert method. Both are founded on the respective homonymic linear models, cf. (Koch, 1997). The word 'adjustment' puts emphasis on the fact that an estimation has to handle redundancy in observational data appropriately, i.e. to weight unreliable data to a lesser extend. In order to overcome the inherent noisiness of measurements one typically introduces a redundancy by taking much more measurements than necessary to describe the process. Each observation must have its own covariance matrix describing the corresponding Gaussian probability density function that is assumed to model the observational error. The determination of which is inferred from the knowledge of the underlying measurement process. The matrices serve as weights and thereby introduce a local error metric.

The principle of least squares adjustment, i.e. to minimize the sum of squared weighted errors $\Delta\mathsf{y}_i$, is often denoted as

$$\sum_i \Delta\mathsf{y}_i^{\mathsf{T}} \Sigma_{\mathsf{y}_i}{}^{-1} \Delta\mathsf{y}_i \;\longrightarrow\; \min\,, \tag{53}$$

where Σ_{y_i} is a covariance matrix assessing the confidence of y_i.

Let $\{\mathsf{b}_1, \mathsf{b}_2, \ldots, \mathsf{b}_N\}$ be a set of N observations, for which we introduce the abbreviation $\{\mathsf{b}_{1..N}\}$. Each observation b_i is associated with an appropriate covariance matrix Σ_{b_i}. An entity, parameterized by a vector p, is to be fitted to the observational data. Consequently, we define a condition function $\mathsf{g}(\mathsf{b}_i,\, \mathsf{p})$ which is supposed to be zero if the observations and the entity in demand fit algebraically. Besides, it is often inevitable to define constraints $\mathsf{h}(\mathsf{p}) \;=\; \mathsf{0}$ on the parameter vector p. This is necessary if there are functional dependencies within the parameters. Consider, for example, the parameterization of a Euclidian normal vector n using three variables $\mathsf{n} = [n_1, n_2, n_3]^{\mathrm{T}}$. A constraint $\mathsf{n}^{\mathrm{T}}\ \mathsf{n} = 1$ could be avoided using spherical coordinates θ and ϕ, i.e. $\mathsf{n} = [\cos\theta\cos\phi, ,\cos\theta\sin\phi, \sin\theta]$. In the following sections, we refer to the functions g and h as G-constraint and H-constraint, respectively.

Note that most of the fitting problems in these sections are not linear but quadratic, i.e. the condition equations require a linearization and estimation becomes an iterative process. An important issue is thus the search for an initial estimate (starting point). If we know an already good estimate $\hat{\mathsf{p}}$, we can make a linearization of the G-constraint yielding $(\partial_{\mathsf{p}}\,\mathsf{g})(\mathsf{b}_i,\hat{\mathsf{p}})\Delta\mathsf{p} + \mathsf{g}(\mathsf{b}_i,\,\hat{\mathsf{p}}) \approx 0$. Hence, with $\mathsf{U}_i = (\partial_{\mathsf{p}}\,\mathsf{g})(\mathsf{b}_i,\hat{\mathsf{p}})$ and $\mathsf{y}_i = -\mathsf{g}(\mathsf{b}_i,\,\hat{\mathsf{p}})$: $\mathsf{U}_i\Delta\mathsf{p} = \mathsf{y}_i + \Delta\mathsf{y}_i$, which exactly matches the linear Gauss-Markov model. The minimization of equation (53) in conjunction with the Gauss-Markov model leads to the best linear unbiased estimator. Note that we have to leave the weighting out in equation (53), since our covariance matrices Σ_{b_i} do not match the Σ_{y_i}. Subsequently, we consider a model which includes the weighting.

If we take our observations as estimates, i.e. $\{\hat{\mathsf{b}}_{1..N}\} = \{\mathsf{b}_{1..N}\}$, we can make a Taylor series expansion of first order at $(\hat{\mathsf{b}}_i,\,\hat{\mathsf{p}})$ yielding

$$(\partial_{\mathsf{p}}\,\mathsf{g})(\hat{\mathsf{b}}_i,\hat{\mathsf{p}})\Delta\mathsf{p} + (\partial_{\mathsf{b}}\,\mathsf{g})(\hat{\mathsf{b}}_i,\hat{\mathsf{p}})\Delta\mathsf{b}_i + \mathsf{g}(\hat{\mathsf{b}}_i,\,\hat{\mathsf{p}}) \approx 0\,. \tag{54}$$

Similarly, with $\mathsf{V}_i = (\partial_{\mathsf{b}}\,\mathsf{g})(\hat{\mathsf{b}}_i,\hat{\mathsf{p}})$ we obtain $\mathsf{U}_i\Delta\mathsf{p} + \mathsf{V}_i\Delta\mathsf{b}_i = \mathsf{y}_i$, which exactly matches the linear Gauss-Helmert model. Note that the error term $\Delta \mathrm{y_i}$ has been replaced by the linear combination $\Delta\mathsf{y}_i = -\mathsf{V}_i\Delta\mathsf{b}_i$; the Gauss-Helmert differs from the Gauss-Markov model in that the observations have become random variables and are thus allowed to undergo small changes $\Delta\mathsf{b}_i$ to compensate for errors. But changes have to be kept minimal, as observations represent the best available. This is achieved by replacing equation (53) with

$$\sum_i \Delta\mathsf{b}_i^{\mathsf{T}}\Sigma_{\mathsf{b}_i}{}^{-1}\Delta\mathsf{b}_i \;\longrightarrow\; \min\,, \tag{55}$$

where $\Delta\mathsf{b}_i$ is now considered as error vector.

The minimization of (55) subject to the Gauss-Helmert model can be done using Lagrange multipliers. By introducing $\Delta\mathsf{b} = [\Delta\mathsf{b}_1^{\mathsf{T}}, \Delta\mathsf{b}_2^{\mathsf{T}}, \ldots, \Delta\mathsf{b}_N^{\mathsf{T}}]^{\mathsf{T}}$, $\Sigma_{\mathsf{b}} = \mathtt{diag}([\Sigma_{\mathsf{b}_1}, \Sigma_{\mathsf{b}_2}, \ldots, \Sigma_{\mathsf{b}_N}])$, $\mathsf{U} = [\mathsf{U}_1^{\mathsf{T}}, \mathsf{U}_2^{\mathsf{T}}, \ldots, \mathsf{U}_N^{\mathsf{T}}]^{\mathsf{T}}$, $\mathsf{V} = \mathtt{diag}([\mathsf{V}_1, \mathsf{V}_2, \ldots, \mathsf{V}_N])$ and $\mathsf{y} = [\mathsf{y}_1^{\mathsf{T}}, \mathsf{y}_2^{\mathsf{T}}, \ldots, \mathsf{y}_N^{\mathsf{T}}]^{\mathsf{T}}$ the Lagrange function Ψ, which is now to be minimized, becomes

$$\Psi(\Delta\mathsf{p},\,\Delta\mathsf{b},\,\mathsf{u},\,\mathsf{v}) \;=\; \frac{1}{2}\,\Delta\mathsf{b}^{\mathsf{T}}\,\Sigma_{\mathsf{b}}^{-1}\,\Delta\mathsf{b} \;-\; \Big(\,\mathsf{U}\,\Delta\mathsf{p} + \mathsf{V}\,\Delta\mathsf{b} - \mathsf{y}\,\Big)^{\mathsf{T}}\mathsf{u} \;+\; \Big(\,\mathsf{H}\,\Delta\mathsf{p} \;-\; \mathsf{z}\,\Big)^{\mathsf{T}}\mathsf{v} \tag{56}$$

The last summand in Ψ corresponds to the linearized H-constraint, where $\mathsf{H} = (\partial_{\mathsf{p}}\mathsf{h})(\hat{\mathsf{p}})$ and $\mathsf{z} = -\,\mathsf{h}(\hat{\mathsf{p}})$ was used. That term can be omitted, if p has no functional dependencies. A differentiation of Ψ with respect to all variables gives an extensive matrix equation, which could already be solved. Nevertheless, it can be considerably reduced with the substitutions $\mathsf{N} = \sum_i \mathsf{U}_i^{\mathsf{T}}\left(\mathsf{V}_i\;\Sigma_{\mathsf{b}_i}\;\mathsf{V}_i^{\mathsf{T}}\right)^{-1}\mathsf{U}_i$ and $\mathsf{z}_N = \sum_i \mathsf{U}_i^{\mathsf{T}}\left(\mathsf{V}_i\;\Sigma_{\mathsf{b}_i}\;\mathsf{V}_i^{\mathsf{T}}\right)^{-1}\mathsf{y}$. The resultant matrix equation is free from $\Delta\mathsf{b}$ and can be solved for $\Delta\mathsf{p}$

$$\begin{bmatrix} \mathsf{N} & \mathsf{H}^{\mathsf{T}} \\ \mathsf{H} & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathsf{p} \\ \mathsf{v} \end{bmatrix} = \begin{bmatrix} \mathsf{z}_N \\ \mathsf{z} \end{bmatrix}. \tag{57}$$

For the corrections $\{\Delta\mathsf{b}_{1..N}\}$, which are now minimal with respect to the Mahalanobis distance (55), we compute

$$\Delta \mathsf{b}_i = \Sigma_{\mathsf{b}_i} \mathsf{V}_i^{\mathsf{T}} \left(\mathsf{V}_i \Sigma_{\mathsf{b}_i} \mathsf{V}_i^{\mathsf{T}} \right)^{-1} \left(\mathsf{y}_i - \mathsf{U}_i \, \Delta \mathsf{p} \right). \tag{58}$$

It is an important by-product that the (pseudo-) inverse of the quadratic matrix in equation (57) contains the covariance matrix $\Sigma_{\Delta \mathsf{p}} = \Sigma_{\mathsf{p}}$ belonging to p. The similar solution for the Gauss-Markov model and the corresponding proofs and derivations can be found in (Koch, 1997). Due to outstanding convergence properties we start iterating with the Gauss-Markov method. At the optimum we start the slower Gauss-Helmert method, which ultimately adjusts the estimate according to the uncertainties Σ_{b_i}.

4.2 Fitting a Circle in 3D

Now we show how the estimation method can be used in CGA to fit a circle in 3D-space to a set of *N* data points $\{\boldsymbol{b}_{1..\mathrm{N}}\}$. Each data point is given with its mean $\boldsymbol{b}_i$ and covariance matrix Σ_{b_i}. In order to apply the estimation methods as described, we need a G-constraint and possibly an H-constraint. We therefore give an introduction to circles in CGA.

We represent a circle by the inner product null space $\mathbb{X}$ of a 2-blade $\boldsymbol{C}$. That space consists of all conformal points **X**, the inner product of which with the circle $\boldsymbol{C}$ is zero, i.e. $\mathbb{X}$ =.$\{\boldsymbol{X} = \mathcal{K}(\boldsymbol{x}) \,|\, \boldsymbol{X} \cdot \boldsymbol{C} = 0\}$ To understand this relationship, consider the inner product null space of a sphere $\boldsymbol{S}_\mathrm{r}$ with radius r and center $\boldsymbol{m}$. It can be created from a point $\boldsymbol{S}_0 = \mathcal{K}(\boldsymbol{m}) = \boldsymbol{m} + \frac{1}{2}\boldsymbol{m}^2 \mathbf{e}_\infty + \mathbf{e}_o$ by subtracting the term '$\frac{1}{2} r^2 \mathbf{e}_\infty$'. The sphere is thus given by $\boldsymbol{S}_r = \boldsymbol{m} + \frac{1}{2}(\boldsymbol{m}^2 - r^2)\mathbf{e}_\infty + \mathbf{e}_o$. For some vector $\boldsymbol{x}$ it can be verified that $\mathcal{K}(\boldsymbol{x}) \cdot \boldsymbol{S}_r = 0 \in \mathbb{R}$ *iff* $\|\boldsymbol{x} - \boldsymbol{m}\|_2 = r$. Now, consider two intersecting spheres $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. A circle intuitively consists of all points $\boldsymbol{X}$ lying on $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. Intersection can be expressed by the outer product and in fact the circle definition is $\boldsymbol{C} = \boldsymbol{S}_1 \wedge \boldsymbol{S}_2$. For a justification examine the inner product $\boldsymbol{X} \cdot \boldsymbol{C}$

$$\boldsymbol{X} \cdot (\boldsymbol{S}_1 \wedge \boldsymbol{S}_2) = (\boldsymbol{X} \cdot \boldsymbol{S}_1)\boldsymbol{S}_2 - (\boldsymbol{X} \cdot \boldsymbol{S}_2)\boldsymbol{S}_1 \,. \tag{59}$$

The terms cannot cancel each other if S1 $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are linearly independent, i.e. if they do not represent the same sphere. The upper equation is therefore zero *iff* $\boldsymbol{X}$ is located on $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ as well.

Remarkably, we have found an appropriate G-constraint right from the definition of the circle's inner product null space itself. It remains to transfer the inner product expression $\boldsymbol{X} \cdot \boldsymbol{C}$ to an equivalent matrix expression. As there are ten basis blades of grade two in $\mathbb{R}_{4,1}$, we have $\Phi(\boldsymbol{C}) = \mathsf{p} \in \mathbb{R}^{10}$. The points $\{\boldsymbol{b}_{1..N}\}$ are embedded and mapped as follows: $\Phi(\mathcal{K}(\boldsymbol{b}_i) = \boldsymbol{B}_i) = \mathsf{b}_i \in \mathbb{R}^5$. Note that our condition equation (59) yields a vector, being defined by five components in $\mathbb{R}_{4,1}$. Consequently, we obtain

$$\Phi(\boldsymbol{B}_i \cdot \boldsymbol{C}) = \mathsf{U}(\mathsf{b}_i)\,\mathsf{p} = \mathsf{V}(\mathsf{p})\,\mathsf{b}_i = \mathsf{g}(\mathsf{b}_i, \mathsf{p}) \in \mathbb{R}^5, \tag{60}$$

which can be differentiated easily. Thus, the required Jacobians $\{\mathsf{U}_{1..N}\}$ and V follow from the bilinearity of geometric algebra products in an implicit manner.

Because a circle in 3D-space can be described by a minimum number of six parameters, we face a functional dependency of grade 4 = 10 – 6 within p. As mentioned in section 4.1, we have to introduce constraints on the parameters, namely the H-constraint $\mathsf{h}(\mathsf{p})$. We enforce $\boldsymbol{C}$ to be a circle by requiring that $\boldsymbol{C} \wedge \boldsymbol{C} = \mathbf{0}$, which can be shown to be sufficient. In almost the same way as for the G-constraint, the usage of Φ allows us to derive the H-matrix. Being in the possession of all necessary matrices, we are able to run the estimation in order to solve for the corrections $\Delta \mathsf{p}$ and $\{\Delta \mathsf{b}_{1..N}\}$.

We remain with this stage and refer the reader to the next estimation example. There, we explicitly derive the constraint functions in terms of the tensor notation.

As mentioned earlier, our method provides the covariance matrix Σ_{p} of the estimated entity p as well. It shows up to which degree the model fits the observations and how advantageously they were initially distributed. It does not reflect to which extend the estimate deviates from a potentially perfect fit, i.e. it is no quality measure for our method. Figure 4 exemplarily shows the uncertainty of an estimated circle. The surrounding tubes, indicated by slices, show the standard deviation of the estimates.

4.3 Fitting two Point Clouds in 3D

In this part, we describe how the proposed methods can be used to estimate an RBM; it extends a rotation, given by a rotor, by a translational component along the axis of rotation. Hence, we can think of it as a screw motion, cf. (Rosenhahn, 2003). In geometric algebra an RBM is represented by an operator called motor. In the scope of pose estimation, the pose is uniquely characterized by an RBM. The estimation of motors is thus a first step towards the perspective pose estimation problem.

Let $\{a_{1..N}\}$ and $\{\mathrm{b}_{1..N}\}$ be two sets of N Euclidian points each. The latter represent the observations for which we have the covariance matrices $\{\Sigma_{\boldsymbol{b}_{1..N}}\}$. The set $\{\boldsymbol{a}_{1..N}\}$ is assumed to have no uncertainty. Let $\boldsymbol{A}_i = \mathcal{K}(\boldsymbol{a}_i)$ and $\boldsymbol{B}_i = \mathcal{K}(\boldsymbol{b}_i)$ denote the conformal embedding of $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$, respectively. We search for the motor $\boldsymbol{M}$, which best transforms all points in $\{\boldsymbol{A}_{1..N}\}$ to the respective points in $\{\boldsymbol{B}_{1..N}\}$. The scenario is shown in figure 5.

Using geometric algebra, we can easily write $\boldsymbol{M}\boldsymbol{A}_i\widetilde{\boldsymbol{M}} = \boldsymbol{B}_i$ cf. (Perwass & Sommer, 2002). Note that a motor is a unitary versor, i.e. it has to satisfy $\boldsymbol{M}\widetilde{\boldsymbol{M}} = 1$. Exploiting this fact, we rearrange the previous formula and obtain the G- constraint

$$\begin{array}{cccccccc} (\boldsymbol{M} & & \boldsymbol{A}_i) & - & (\boldsymbol{B}_i & & \boldsymbol{M}) & = & 0 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \mathsf{p}^k & \mathsf{G}^t{}_{kl} & \mathsf{a}_i{}^l & - & \mathsf{b}_i{}^l & \mathsf{G}^t{}_{lk} & \mathsf{p}^k & = & 0^t \end{array}, \tag{61}$$

where we used $\Phi(\boldsymbol{A}_i) = \mathsf{a}_i$, $\Phi(\boldsymbol{B}_i) = \mathsf{b}_i$ and $\Phi(\boldsymbol{M}) = \mathsf{p} \in \mathbb{R}^8$. The tensor G encodes the geometric product. In order to evaluate the matrices U and V, we differentiate equation (61) with respect to p and b, respectively. Hence, we get $\mathsf{U}(\mathsf{b}_i) = \mathsf{G}^t{}_{kl}\mathsf{a}_i{}^l - \mathsf{b}_i{}^l\mathsf{G}^t{}_{lk}$ and $\mathsf{V}(\mathsf{p}) = -\,\mathsf{p}^k\mathsf{G}^t{}_{lk}$.

Since an RBM is defined by six rather than eight parameters, we need the H-constraint. We again exploit unitarity and choose $\mathsf{h}(\mathsf{p}) = \Phi(\boldsymbol{M}\widetilde{\boldsymbol{M}} - 1) = \mathsf{p}^k\,\mathsf{p}^l\,\mathsf{R}^m{}_l\,\mathsf{G}^t{}_{km} - \delta^{t,1}$. The tensor R encodes the reverse operation and $\delta^{t,1}$ is zero, except for t = 1. Differentiation $\partial_{\mathsf{p}}\mathsf{h}$ yields $\mathsf{H}(\mathsf{p}) = \mathsf{p}^l\,(\mathsf{R}^m{}_l\,\mathsf{G}^t{}_{km} + \mathsf{R}^m{}_k\,\mathsf{G}^t{}_{lm})$. The estimate for $\boldsymbol{M}$ can now be computed by simply substituting the matrices $\{\mathsf{U}_{1..N}\}$, V and H into the respective equations given in the theoretical part.

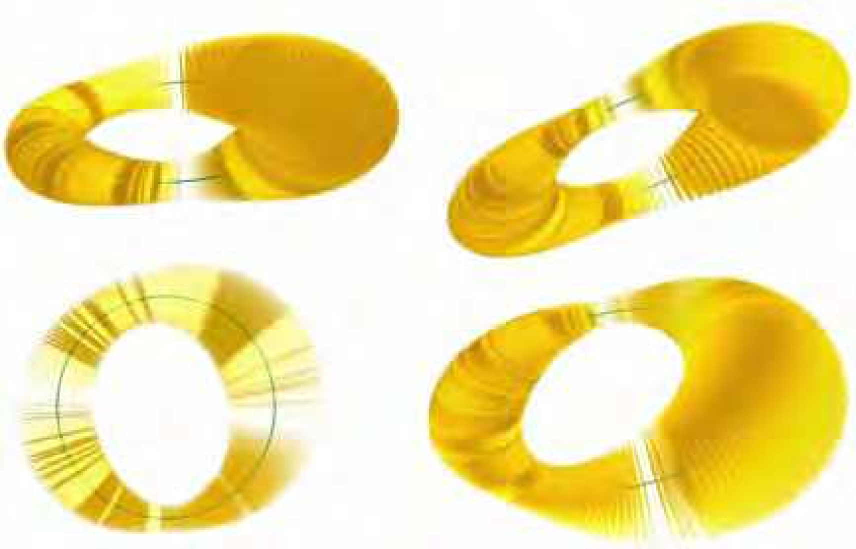

Figure 4. Fitting a circle: four views of a circle's uncertainty (standard deviation)

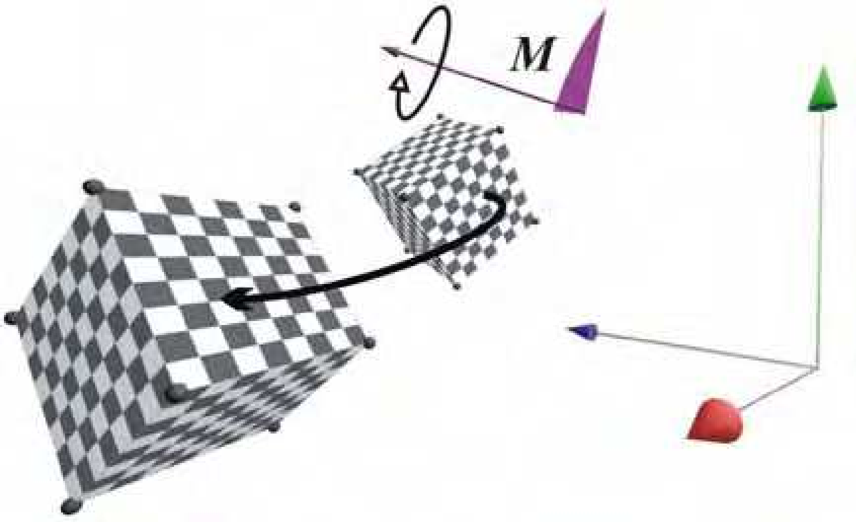

Figure 5. Fitting two point clouds: the rotation of the motor $\boldsymbol{M}$ is indicated by the partial disc. The translational part is specified by the arrow attached to it

5. Pose Estimation from Uncertain Omnidirectional Image Data

We present a sophisticated application of the parameter estimation from uncertain data as depicted in the previous section. It reads ′perspective 2D-3D pose estimation for omnidirectional vision using line-plane correspondences′ and has strong geometrical streaks, which is why we spend an extra section. In this context, we introduce the ′inversion camera model', which has the ability to model a variety of distinct camera systems thereby taking image distortions into account.

Pose estimation certainly is a well-studied subject, but not in case of an omnidirectional vision system. Hence, our objective was to develop accurate pose estimation for omnidirectional vision, given imprecise image features, i.e. 2D-sensory data. Note that these features can readily be detected by the method proposed in section 3.

Comparable to triangulation, the accuracy of an estimated pose benefits when landmarks can be seen in clearly different directions. But the most significant advantages of omnidirectional vision are related to navigation, since the objects remain on the image plane under most camera movements. We consider a single viewpoint catadioptric vision sensor. It combines a customary camera with a parabolic mirror and provides a panoramical view of 360°.

We make the assumption to have 3D-models of the interesting objects we observe in the images. Secondly, we assume to know the one-to-one correspondences between the model features and the image features. Note that a model consists of 3D-lines, which mostly represent object edges, which in turn, are likely to generate a line under imaging; consequently, we have lines as image features. We herewith extend our previous work where we had been employing point features and point models.

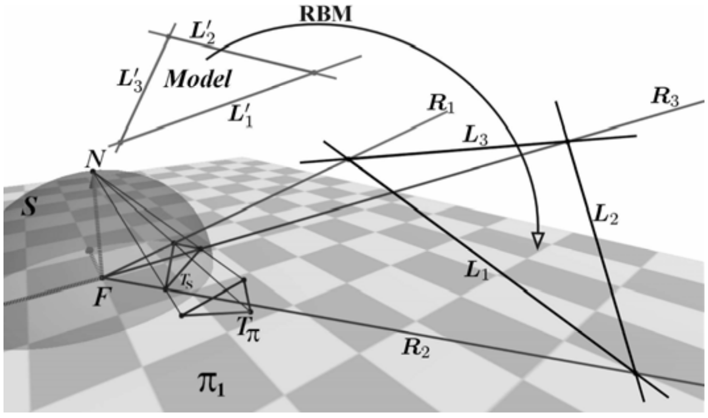

Figure 6. Fitting a triangle model to the projection planes spanned by $\boldsymbol{R}_1$, $\boldsymbol{R}_2$ and $\boldsymbol{R}_3$

5.1 Omnidirectional 2D-3D Pose Estimation

Roughly speaking, rigidly moving an object in 3D such that it comes into agreement with 2D-sensory data of a camera is called 2D-3D pose estimation (Grimson, 1990). Specifically, we estimate an RBM in 3D, such that the model lines come to lie on the projection planes of the underlying image lines, see figure 6.

The method to be proposed comprises three steps: from those pixels corresponding to visible model lines, we estimate projection planes with associated uncertainties. In a second step, a simple algorithm is used to do prior rotation estimation being a first and rough guess at the rotational part of the desired RBM. As a result the model will be aligned such that its lines are nearly parallel to the respective projection planes. We finally estimate the entire pose taking the computed plane uncertainties into account as well.

Before we explain those steps in detail, we give a sketch of catadioptric imaging.

5.2 Omnidirectional Imaging

Consider a camera, focused at infinity, which looks upward at a parabolic mirror centered on its optical axis. This setup is shown in figure 7. A light ray emitted from world point $\boldsymbol{P}_w$ that would pass through the focal point $\boldsymbol{F}$ of the parabolic mirror M, is reflected parallel to the central axis of M, to give point $\boldsymbol{p}_2$ on image plane $\boldsymbol{\pi}_2$. Now we use the simplification that a projection to sphere S with a subsequent stereographic projection to $\boldsymbol{\pi}_1$ produces an identical image on $\boldsymbol{\pi}_1$. Accordingly, point $\boldsymbol{P}_w$ maps to $\boldsymbol{P}_S$ and further to $\boldsymbol{p}_1$, see figure 7. Together with the right side of figure 7 it is intuitively clear that infinitely extended lines form great circles on $\boldsymbol{S}$. Moreover, a subsequent stereographic projection, being a conformal mapping, results in circles on the image plane, which then are no more concentric. For details refer to (Geyer & Daniilidis, 2001).

Our approach exploits that the mapping from a projection ray to an image point is bijective and therefore invertible. Moreover, given an image line, we can compute its projection plane.

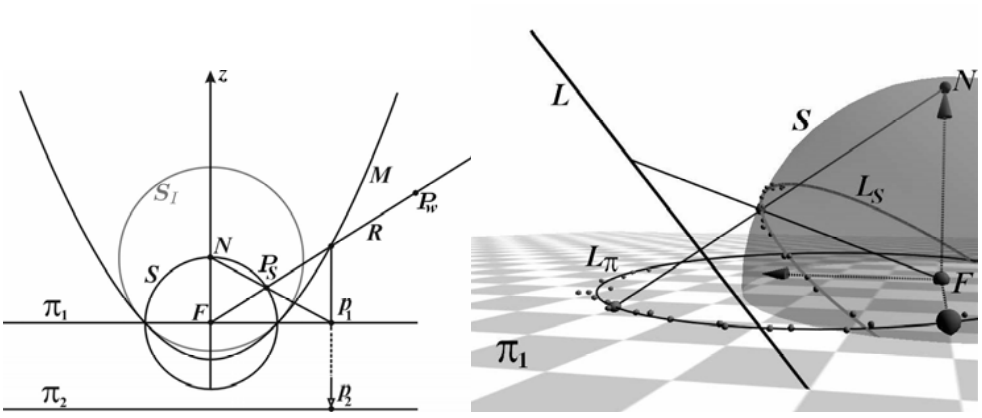

Figure 7. Left: mapping (cross-section) of a world point $\boldsymbol{P}_w$: the image planes $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ are identical. Right: mapping of line $\boldsymbol{L}$ to $\boldsymbol{L}_\pi$ via great circle $\boldsymbol{L}_S$ on $\boldsymbol{S}$. As an example, scattered image data belonging to $\boldsymbol{L}_\pi$ is shown

5.3 Estimating Projection Planes

We must come up with observations in the form of planes for a line-plane fitting; we compute a projection plane for each set of image points that corresponds to a visible model line. To be more specific, we estimate the planes from the stereographically back-projected image points. Hence, the points have to be moved to the projection sphere $\boldsymbol{S}$, see figure 7. This is done by an inversion of the image points in a certain sphere $\boldsymbol{S}_I$. Note that the (uncertain) image points, initially identically 2D-distributed, thereby obtain distinct 3D-uncertainties, which reflect the imaging geometry. The uncertainties are computed using error propagation, where we profit from the inversion being a linear operation in $\mathbb{R}_{4,1}$. The plane estimation can now be done by restricting the circle estimation, see section 4.2, to the three parameters describing the circle's plane. Recall that we obtain a covariance matrix for each estimated plane.

5.4 Prior Model Alignment

The line-plane pose estimation will prove to be a quadratic problem. In such cases, as mentioned in section 4.1, the linearization requires an initial estimate. The prior model alignment provides such a starting point at very low costs. We like to rotate the model such that the set of unit direction vectors $\{\hat{\mathsf{r}}_{1..N}\}$ of its lines lie on the respective planes. Let $\{\hat{\mathsf{n}}_{1..N}\}$ denote the set of normal vectors of all planes, which belong to visible model lines.
We search for a rotation matrix R such that $\hat{}(\forall i):\ \hat{\mathsf{n}}_i^{\mathsf{T}}\,\mathsf{R}\,\hat{\mathsf{r}}_i = 0 \in \mathbb{R}$.
By Rodrigues's formula (1840) we know that the rotation matrix R regarding a rotation of angle θ around unit vector $\hat{\mathsf{a}} = [a_1, a_2, a_3]^{\mathsf{T}}$ can be expressed by an exponential map of $\mathsf{A} = [[0, a_3, -a_2]^{\mathsf{T}}[-a_3, 0, a_1]^{\mathsf{T}}[a_2, -a_1, 0]^{\mathsf{T}}]$: $\mathsf{R} = \exp(\theta\mathsf{A})$ which is $\mathsf{R} = \mathsf{I}_3 + \sin\theta\,\mathsf{A} + (1-\cos\theta)\mathsf{A}^2$, where I_3 denotes the 3×3 identity matrix. For small angles we obtain $\mathsf{R} = \mathsf{I}_3 + \theta\mathsf{A}$. With this relation and due to the skew symmetric structure of $\mathsf{A}' = \theta\mathsf{A}$ it is possible to solve for $\mathsf{a}' = [\theta a_1, \theta a_2, \theta a_3]^{\mathsf{T}}$, where each line-plane pair gives one line $\hat{\mathsf{n}}_i^{\mathsf{T}}\,\mathsf{A}'\,\hat{\mathsf{r}}_i = -\hat{\mathsf{n}}_i^{\mathsf{T}}\,\hat{\mathsf{r}}_i$ in an overdetermined system of linear equations. Every run of this procedure yields a rotation matrix, the concatenation of which gives the desired rotation matrix R. Once, the rotated lines are close enough to the planes w.r.t. some threshold the procedure can be stopped.

5.5 Perspective Line-Plane Pose Estimation

Here we derive geometric constraint equations for the stochastic estimation methods presented in the previous section.
Let $\boldsymbol{P}$ be a projection plane, see section 5.3. For any line $\boldsymbol{L}$ lying on $\boldsymbol{P}$, we have $\boldsymbol{P}\wedge\boldsymbol{L} = 0 \in \mathbb{R}_{4,1}$ A model line $\boldsymbol{L}'$ is transformed by an RBM represented by $\boldsymbol{M}$, say, via the operation $\boldsymbol{M}\,\boldsymbol{L}'\,\widetilde{\boldsymbol{M}}$. Therefore, if we have estimated the correct $\boldsymbol{M}$, a model line $\boldsymbol{L}'$ with corresponding projection plane $\boldsymbol{P}$ has to satisfy $\boldsymbol{P}\wedge(\boldsymbol{M}\,\boldsymbol{L}'\,\widetilde{\boldsymbol{M}}) = 0$
Using Φ from section 2.3, we can identify our elements $\boldsymbol{P}$, $\boldsymbol{L}'$ and $\boldsymbol{M}$ with particular vectors $\mathsf{n} \in \mathbb{R}^3$, $\mathsf{l}' \in \mathbb{R}^6$ and $\mathsf{p} \in \mathbb{R}^8$. For example, n simply denotes the normal vector of the plane represented by $\boldsymbol{P}$. We contract all constituent product tensors to one tensor Q and obtain condition function g for one line-plane pair

$$\mathsf{g}^t(\mathsf{n},\,\mathsf{p},\,\mathsf{l}') \;=\; \sum_{i,j,k,l} \mathsf{p}^i\,\mathsf{p}^j\,\mathsf{n}^k\,\mathsf{l}'^{\,l}\,\mathsf{Q}^t{}_{ijkl} \;=\; 0\,, \tag{62}$$

Algebraically, the constraint $\boldsymbol{P}\wedge\boldsymbol{L}$ may only be non-zero in four of its $2^5 = 32$ components, which is why we have $t \in \{1,\dots,4\}$. The observations and parameters are n and p, respectively. Hence, differentiating would yield the matrices $\{\mathsf{U}_{1..N}\}$ and $\{\mathsf{V}_{1..N}\}$ required in section 4.1. The eight components of $\boldsymbol{M}$ are an overparameterization, again, such that we need to include the H-constraint $\boldsymbol{M}\,\widetilde{\boldsymbol{M}} = 1$ from section 4.3.

5.6 Inversion Camera Model

The inversion camera model can be used for image rectification. Besides, it can readily be incorporated into the previously presented pose estimation methods as inversion embodies the main CGA operation. We briefly discuss both applications.
We go on from section 5.2 in which we dealt with imaging. The considerations were limited to the special case of a parabolic catadioptric imaging system: a stereographic projection had been replaced by an inversion of the projection sphere S in a inversion sphere S_I. This is one case of what the inversion camera model, which was proposed by (Perwass & Sommer,

2006), can handle. It basically expresses a projective mapping in terms of an inversion. It enables a continuous transition between different geometries of imaging, as fisheye optics or the classic pinhole camera, merely by changing two parameters. These determine the constellation of suitable spheres ***S*** and ***S***$_\mathrm{I}$ in respect to the focal point ***F***. In addition to the left side of figure 7, which illustrates a parabolic catadioptric imaging system, figure 8 depicts two further interesting constellations. To demonstrate the versatility of the inversion camera model, recall the imaging principle described in section 5.2. It can equally be applied to the left side of figure 8, where the same operations describe a completely different camera system: ′point $\boldsymbol{P}_w$ maps to $\boldsymbol{P}_S$ and further to ***p***′.

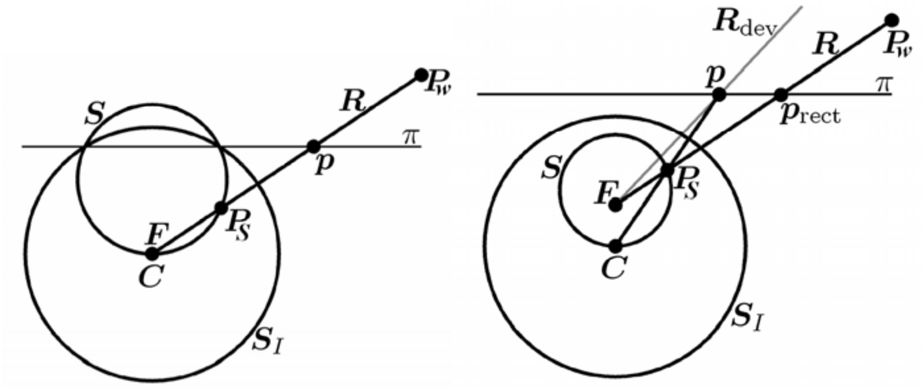

Figure 8. Mapping schemes (cross-section) in terms of the inversion camera model. Left: setup reflecting a pinhole camera. Right: setup modeling a real lens by taking radial distortions into account. Namings are in concordance with figure 7; ***C*** denotes the center of ***S***$_\mathrm{I}$

The aim of image rectification is to undo distortions which originate from a variety of optical imperfections. The right side of figure 8 shows the problem. The ray belonging to world point $\boldsymbol{P}_w$ was subjected to distortion which lead to the ray $\boldsymbol{R}_{\text{dev}}$ that eventually produced ***p***. However, $\boldsymbol{R}_{\text{dev}}$ deviates from the geometrically true ray ***R*** in a non-linear manner depending on the angle to the optical axis. Hence a mapping has to be found that corrects the position of point ***p***, within the image plane, such that it comes to lie on its projection ray ***R*** again.

We denote the rectified point $\boldsymbol{p}_{\text{rect}}$. In (Perwass & Sommer, 2006), the authors discovered that moving off the inversion sphere $\boldsymbol{S}_I$ from ***F*** , which distinguishes the mapping schemes in figure 8, results in a mapping suitable to model distortions. It consists of two parts. First a versor ***K***, which essentially does the inversion of the image point ***p***, is applied. Next, the corresponding ray ***R*** is constructed and intersected with image plane **π** to give $\boldsymbol{p}_{\text{rect}}$.

Our subsequent considerations require a right handed coordinate system. The $\mathbf{e}_3$-axis denotes the optical axis. It points upwards and is incident with ***F***. The $\mathbf{e}_1$-axis points to the right and is aligned with the image plane. Hence, all image points lie on the $\mathbf{e}_1$-$\mathbf{e}_2$-plane.

The inversion sphere $\boldsymbol{S}_I$ of radius ***r*** is defined by $\boldsymbol{S}_I = s_1\mathbf{e}_3 + \frac{1}{2}s_2\mathbf{e}_\infty + \mathbf{e}_o$, where we used the abbreviation $s_2 = s_1^2 - r^2$. One of the simplest forms ***K*** can take on is $\boldsymbol{K} = \mathbf{S}_I\boldsymbol{D}$. In order to handle scaling and for numerically well-balanced equations, the inversion in $\boldsymbol{S}_I$ is preceded by the dilator ***D*** (isotropic scaling operator). The dilation operator ***D*** for a scaling by a factor d is given by D = 1 + γ *E*, where we defined γ = (d – 1)/(d+ 1). The image point transformation operator *K* is then given by

$$\boldsymbol{K} \;=\; \boldsymbol{S}_I \boldsymbol{D} \;=\; k_1 \mathbf{e}_3 + k_2 \mathbf{e}_\infty + k_3\, \mathbf{e}_o + k_4 \mathbf{e}_3 \boldsymbol{E}\,, \tag{63}$$

with $k_1 \;:=\; s_1,\; k_2 \;:=\; \frac{1}{2}(1-\gamma)\, s_2,\; k_3 \;:=\; 1+\gamma$ and $k_4 \;:=\; -\gamma\, s_1$. Let $\boldsymbol{P} \;=\; \mathcal{K}(\boldsymbol{p})$ be the embedding of an image point ***p***. Similar to figure 8, we denote $\boldsymbol{P}_S \;=\; \boldsymbol{K}\,\boldsymbol{P}\,\widetilde{\boldsymbol{K}}$ the point transformed by ***K***. For determining the rectified point $\boldsymbol{p}_{\mathrm{rect}}$, as intended, it remains to intersect the projection ray ***R***, now given by

$$\boldsymbol{R} \;=\; \boldsymbol{P}_S \wedge \boldsymbol{F} \wedge \mathbf{e}_\infty \;=\; \left(\boldsymbol{K}\,\boldsymbol{P}\,\widetilde{\boldsymbol{K}}\right) \wedge \boldsymbol{F} \wedge \mathbf{e}_\infty\,, \tag{64}$$

with the image plane **π**. The intersection is an elementary operation in CGA and yields the conformal point $\boldsymbol{P}_{\mathrm{rect}} = \mathcal{K}(\boldsymbol{p}_{\mathrm{rect}})$. Computing $\boldsymbol{p}_{\mathrm{rect}} = \mathcal{K}^{-1}(\boldsymbol{P}_{\mathrm{rect}})$ yields

$$\boldsymbol{p}_{\mathrm{rect}} \;=\; \frac{\beta}{1\,+\,\alpha \boldsymbol{p}^2}\,\boldsymbol{p}\,, \tag{65}$$

with the two parameters

$$\alpha \;:=\; \frac{1-s_1}{s_1(s_1-s_2)} \qquad \text{and} \qquad \beta \;:=\; \frac{r^2 d}{s_1(s_1-s_2)}\,. \tag{66}$$

It is noteworthy that $\boldsymbol{p}_{\mathrm{rect}}/\beta$ is the respective expression produced by the so-called division model. It was proposed by (Fitzgibbon, 2001) and can be considered equivalent to the camera inversion model. The division model itself was shown in (Claus & Fitzgibbon, 2005) to have a rectification quality comparable to a fourth order radial polynomial approach. The camera inversion model is thus a sufficiently good approximation of lens distortion for many applications.

In (Perwass & Sommer, 2006), the estimation of lens distortion was successfully combined with pose estimation by means of the estimation methods presented in this text. Specifically, the pose, the focal length and the lens distortion were estimated at the same time. For example, in case of a point-line fitting a model point ***P'*** is to be transformed by an RBM ***M*** such that it comes to lie on the corresponding rectified projection ray ***R***. In analogy to section 5.5 and with the help of equation (64) it is required for image point $\boldsymbol{P} = \mathcal{K}(\boldsymbol{p})$ that

$$\left(\left(\boldsymbol{K}\,\boldsymbol{P}\,\widetilde{\boldsymbol{K}}\right) \wedge \boldsymbol{F} \wedge \mathbf{e}_\infty\right) \wedge (\boldsymbol{M}\boldsymbol{P}'\boldsymbol{M}) \;=\; \mathbf{0}\,. \tag{67}$$

A respective tensor representation can be derived easily, and the necessary constraints follow from differentiation. With this impressive example of the unifying nature of geometric algebra we conclude this chapter.

6. References

Angles, P. (1980). Construction de revêtements du groupe conforme d'un espace vectoriel muni d'une "métrique" de type (p,q). Ann. Inst. Henri Poincaré, 33(1):33-51.

Brackx, F.; Delanghe, R. & Sommen F. (1982). Clifford Analysis, volume 76 of *Research Notes in Mathematics*. Pitman Advanced Publishing Program, Boston, MA.

Brannan, D.A.; Esplen, M.F. & Gray, J.J. (1999). *Geometry*. Cambridge University Press, Cambridge.

Buchholz, S. & Le Bihan, N. (2006). Optimal separation of polarized signals by quaternionic neural networks. In *14th European Signal Processing Conference, EUSIPCO 2006*, September 4-8, Florence, Italy.

Buchholz, S. & Sommer, G. (2006). On Clifford Neurons and Clifford Multi-layer Perceptrons. *Neural Networks*, accepted for publication.

Claus, D. & Fitzgibbon, A.W. (2005). A rational function lens distortion model for general cameras. In *Conference on Computer Vision and Pattern Recognition (CVPR05)*, volume 1, pages 213-219, Washington, DC, USA, IEEE Computer Society.

Faugeras, O. (1995). Stratification of three-dimensional vision: projective, affine, and metric representations. *Journal of the Optical Society of America*, 12(3):465-484.

Felsberg, M. (2002). Low-level image processing with the structure multivector. *Technical Report* Number 0203, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik, März.

Felsberg, M. & Sommer, G. (2001). The monogenic signal. *IEEE Trans. Signal Process.*, 49(12):3136-3144

Felsberg, M. & Sommer, G. (2004). The monogenic scale-space: A unifying approach to phase-based image processing in scale-space. *Journal of Mathematical Imaging and Vision*, 21:5-26.

Fitzgibbon, A.W. (2001). Simultaneous linear estimation of multiple view geometry and lens distortion. In *Conference on Computer Vision and Pattern Recognition (CVPR01)*, Hawaii, volume 1, pages 125-132.

Freeman, W.T. & Adelson, E.H. (1991). The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(9):891-906.

Gabor, D. (1946). Theory of communications. *Journal of the IEEE*, 93(3):429-457.

Gebken, C.; Tolvanen, A. & Sommer, G. (2006). Pose estimation from uncertain omnidirectional image data using line-plane correspondences. In *28. Symposium für Mustererkennung, DAGM 2006*, Berlin, 12.9.-14.9.2006, number 4174 in LNCS, pages 587-596, Springer-Verlag, Heidelberg, Berlin.

Geyer, C. & Daniilidis, K. (2001). Catadioptric projective geometry. *International Journal of Computer Vision*, 45(3):223-243.

Granlund, G.H. & Knutsson, H. (1995). Signal *Processing for Computer Vision*. Kluwer Academic Publishers, Norwell, MA, USA.

Grimson, W.E.L. (1990). *Object Recognition by Computer*. MIT Press, Cambridge, MA.

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

Hestenes, D. (1991). The design of linear algebra and geometry. *Acta Applicandae Mathematicae*, 23:65-93.

Hestenes, D.; Li, H. & Rockwood, A. (2001). New algebraic tools for classical geometry. In G. Sommer, editor, *Geometric Computing with Clifford Algebras*, pages 3-26, Springer-Verlag, Heidelberg, Berlin.

Hestenes, D. & Sobczyk, G. (1984). Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics. *Fundamental theories of physics*. D. Reidel Publishing Company, Dordrecht, 1987. First publ.

Heuel, S. (2004). *Uncertain Projective Geometry*, volume 3008 of LNCS, Springer-Verlag, Heidelberg, Berlin.

Koch, K.R. (1997). *Parameter Estimation and Hypothesis Testing in Linear Models*. Springer-Verlag, Heidelberg, Berlin.

Koenderink, J.J. & van Doorn, A.J. (1987). *Representation of local geometry in the visual system*. Biol. Cybern., 55(6):367-375.

Li, H.; Hestenes, D. & Rockwood, A. (2001). Generalized homogeneous coordinates for computational geometry. In G. Sommer, editor, *Geometric Computing with Clifford Algebras*, pages 27-59, Springer-Verlag, Heidelberg, Berlin.

Li, H.; Hestenes, D. & Rockwood, A. (2001). Spherical conformal geometry with geometric algebra. In G. Sommer, editor, *Geometric Computing with Clifford Algebras*, pages 61-75, Springer-Verlag, Heidelberg, Berlin.

Li, H.; Hestenes, D. & Rockwood, A. (2001). A universal model for conformal geometries of euclidean, spherical and double-hyperbolic spaces. In G. Sommer, editor, *Geometric Computing with Clifford Algebras*, pages 77-104, Springer-Verlag, Heidelberg, Berlin.

Needham, T. (1997). Visual Complex Analysis. Clarendon Press, Oxford.

Pauli, J. (2001). Learning-Based Robot Vision, volume 2048 of Lecture Notes in *Computer Science*. Springer-Verlag, Heidelberg, Berlin.

Perwass, C.; Gebken, C. & Sommer, G. (2005). Estimation of geometric entities and operators from uncertain data. In *27. Symposium f¨ur Mustererkennung, DAGM 2005*, Wien, 29.8.-2.9.005, number 3663 in LNCS. Springer-Verlag, Heidelberg, Berlin.

Perwass, C.; Gebken, C. & Sommer, G. (2006). Geometry and kinematics with uncertain data. In A. Leonardis, H. Bischof, and A. Pinz, editors, *9th European Conference on Computer Vision, ECCV 2006*, May 2006, Graz, Austria, number 3951 in LNCS, pages 225-237. Springer-Verlag, Heidelberg, Berlin.

Perwass, C. & Hildenbrand, D. (2003). Aspects of geometric algebra in euclidean, projective and conformal space. *Technical Report* Number 0310, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik, September.

Perwass, C. & Sommer, G. (2002). Numerical evaluation of versors with Clifford algebra. In Leo Dorst, Chris Doran, and Joan Lasenby, editors, *Applications of Geometric Algebra in Computer Science and Engineering*, pages 341-349. Birkh¨auser.

Perwass, C. & Sommer, G. (2006). The inversion camera model. In *28. Symposium für Mustererkennung*, DAGM 2006, Berlin, 12.-14.09.2006, number 4174 in LNCS, pages 647-656. Springer-Verlag, Heidelberg, Berlin.

Porteous, I.R. (1995). Clifford Algebras and the Classical Groups. *Cambridge Stud. Adv. Math.*, Cambridge University Press, Cambridge.

Rosenhahn, B. (2003). Pose Estimation Revisited. Technical Report Number 0308, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik, September.

Rosenhahn, B. & Sommer, G. (2005). Pose estimation in conformal geometric algebra, part I: The stratification of mathematical spaces. *Journal of Mathematical Imaging and Vision*, 22:27-48.

Rosenhahn, B. & Sommer, G. (2005). Pose estimation in conformal geometric algebra, part II: Realtime pose estimation using extended feature concepts. *Journal of Mathematical Imaging and Vision*, 22:49-70.

Sommer, G. (1997). Algebraic aspects of designing behavior based systems. In G. Sommer and J.J. Koenderink, editors, *Algebraic Frames for the Perception and Action Cycle*, volume 1315 of Lecture Notes in Computer Science, pages 1-28. Proc. Int.Workshop AFPAC97, Kiel, Springer-Verlag, Heidelberg, Berlin.

Sommer, G. (1999). The global algebraic frame of the perception-action cycle. In B. Jähne, H. Haussecker, and P. Geissler, editors, *Handbook of Computer Vision and Applications*, volume 3, pages 221-264. Academic Press, San Diego.

Sommer, G. (2004). A geometric algebra approach to some problems of robot vision. In J. Byrnes, editor, *Computational Noncommutative Algebra and Applications*, number 136 in NATO Science Series II, pages 309-338. Kluwer Academic Publishers, Dordrecht.

Sommer, G. (2005). Applications of geometric algebra in robot vision. In H. Li, P.J. Olver, and G. Sommer, editors, *Computer Algebra and Geometric Algebra with Applications*, volume 3519 of LNCS, pages 258-277. 6th International Workshop IWMM 2004, Shanghai, China and International Workshop GIAE 2004, Xian, China, Springer-Verlag, Heidelberg, Berlin.

Sommer, G. & Zang, D. (2007). Parity symmetry in multi-dimensional signals. *Communications in Pure and Applied Analysis*, accepted.

Stein, E.M. & Weiss G. (1971). Introduction to Fourier Analysis on Euclidean Spaces, volume 32 of *Princeton Mathematical Series*. Princeton University Press, Princeton, N.J.

Zang, D. & Sommer, G. (2006). The monogenic curvature scale-space. In R. Reulke, U. Eckardt, B. Flach, U. Knauer, and K. Polthier, editors, *11th International Workshop on Combinatorial Image Analysis, IWCIA06*, Berlin, volume 4040 of LNCS, pages 320-332. Springer-Verlag, Heidelberg, Berlin.

Zang, D. & Sommer, G. (2006). Detecting intrinsically two-dimensional image structures using local phase. In K. Franke, K. M¨uller, B. Nickolay, and R. Schäfer, editors, *28. Symposium f¨ur Mustererkennung, DAGM 2006*, Berlin, 12.9.-14.9.2006, number 4174 in LNCS, pages 222- 231. Springer-Verlag, Heidelberg, Berlin.

Zang, D. & Sommer, G. (2007). Signal modeling for two-dimensional image structures. *Journal of Visual Communication and Image Representation*, 18(1):81-99.

Zang, D.; Wietzke, L.; Schmaltz, C. & Sommer, G. (2007). Dense optical flow estimation from the monogenic curvature tensor. In *Int. Conf. on Scale-Space and Variational Methods (SSVM)*, Ischia, Italy.

Vision Systems: Applications

Edited by Goro Obinata and Ashish Dutta

ISBN 978-3-902613-01-1
Hard cover, 608 pages
Publisher I-Tech Education and Publishing
Published online 01, June, 2007
Published in print edition June, 2007

Computer Vision is the most important key in developing autonomous navigation systems for interaction with the environment. It also leads us to marvel at the functioning of our own vision system. In this book we have collected the latest applications of vision research from around the world. It contains both the conventional research areas like mobile robot navigation and map building, and more recent applications such as, micro vision, etc.The fist seven chapters contain the newer applications of vision like micro vision, grasping using vision, behavior based perception, inspection of railways and humanitarian demining. The later chapters deal with applications of vision in mobile robot navigation, camera calibration, object detection in vision search, map building, etc.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821