Adv. appl. Clifford alg. 18 (2008), 647–664

0188-7009/030647-18, *published online* June 16, 2008
DOI 10.1007/s00006-008-0117-4

**Advances in
Applied Clifford Algebras**

Parameter Estimation from Uncertain Data in Geometric Algebra

Christian Gebken, Christian Perwass and Gerald Sommer

Abstract. We show how standard parameter estimation methods can be applied to Geometric Algebra in order to fit geometric entities and operators to uncertain data. These methods are applied to three selected problems. One of which is the perspective pose estimation problem. We show experiments with synthetic data and compare the results of our algorithm with standard approaches.

In general, our aim is to find multivectors that satisfy a particular constraint, which depends on a set of uncertain measurements. The specific problem and the type of multivector, representing a geometric entity or a geometric operator, determine the constraint. We consider the case of point measurements in Euclidian 3D-space, where the respective uncertainties are given by covariance matrices. We want to find a best fitting circle or line together with their uncertainty. This problem can be expressed in a linear manner, when it is embedded in the corresponding conformal space. In this space, it is also possible to evaluate screw motions and their uncertainty, in very much the same way.

The parameter estimation method we use is a least-squares adjustment method, which is based on the so-called *Gauss-Helmert model*, also known as *mixed model with constraints*. For this linear model, we benefit from the implicit linearization when expressing our constraints in conformal space. The multivector representation of the entities we are interested in also allows their uncertainty to be expressed by covariance matrices. As a by-product, this method provides such covariance matrices.

Keywords. Geometric Algebra, conformal space, parameter estimation, least squares adjustment, pose estimation, fitting, Mahalanobis distance.

1. Introduction

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 a unified mathematical framework, namely Geometric Algebra. We use the Geometric Algebra of the conformal space of Euclidean 3D-space as introduced in [1]. Consequently, the estimation is applicable to geometric entities and geometric operators. In particular, it can be shown that evaluating points, lines, planes, circle, spheres and their covariance matrices from a set of uncertain points can be done in much the same way as the evaluation of rotation, translation and dilation operators with corresponding covariance matrices. Using standard error propagation, further calculations can be performed with these uncertain entities, while keeping track of the uncertainty.

This text builds on previous works by Förstner et al. [3] and Heuel [5] where uncertain points, lines and planes were treated in a unified manner. The linear estimation of rotation operators in Geometric Algebra was previously discussed in [13], albeit without taking account of uncertainty. In the scope of perspective pose estimation Rosenhahn and Sommer [15] also derived an estimation method for rotation/translation operators employing conformal Geometric Algebra. Their approach is mainly based on the stratification hierarchy of Euclidean, projective and affine spaces. In [11] the estimation of uncertain general operators was introduced.

The structure of this text is as follows. First, we give an introduction to parameter estimation using least squares adjustment in an ordinary vector space. Next, we provide a concise overview of Geometric Algebra and explain our respective notation. We then show in which way observations of Euclidian 3D-space can be embedded into 5D conformal space and how smoothly error propagation integrates into that process. We present three fields of application for the proposed Gauss-Helmert method. For each, we make clear in which way we profit from the expressiveness of Geometric Algebra and we explain how our method can be applied within that framework. We conclude the text with synthetic experiments regarding the three examples and give the conclusions.

2. Least Squares Adjustment

In the field of parameter estimation one usually tries to parametrize some physical process $\mathcal{P}$ in terms of a model $\mathcal{M}$ and a corresponding parameter vector p. The components of p are then to be estimated from a set of N observations $\{\mathsf{b}_{1..N}\}$[1], stemming from $\mathcal{P}$. In order to overcome the inherent noisiness of measurements, one typically introduces redundancy by taking much more measurements than necessary to describe the process.

The term 'adjustment' in 'least squares adjustment' emphasizes that this method focuses on the way in which redundant observational data influences the least squares minimization. The prerequisite for this is that each observation b_i is associated with an appropriate covariance matrix Σ_{b_i}. It describes the respective Gaussian probability density function that is assumed to model the origination of

[1]We use the abbreviation $\{\mathsf{b}_{1..N}\}$ for a set $\{\mathsf{b}_1, \mathsf{b}_2, \ldots, \mathsf{b}_N\}$.

the observational error. The matrices are taken as weights and introduce a local error metric.

Classically, the method of least squares consists in finding estimates $\{\hat{\mathsf{b}}_{1..N}\}$ for the observations $\{\mathsf{b}_{1..N}\}$, such that the sum of squared errors is minimized

$$\sum_i \|\mathsf{b}_i - \hat{\mathsf{b}}_i\|^2 = \sum_i \Delta\mathsf{b}_i^{\mathsf{T}}\Delta\mathsf{b}_i \longrightarrow \min . \tag{2.1}$$

The minimization has to be done in conjunction with a condition function $\mathsf{g}(\mathsf{b}_i) = \mathsf{0}$, to avoid the trivial solution $\hat{\mathsf{b}}_i = \mathsf{b}_i$. The condition function reflects the model $\mathcal{M}$ and usually also depends on the parameter vector p: $\mathsf{g}(\mathsf{b}_i, \mathsf{p}) = \mathsf{0}$. 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. As an example consider the parametrization of a Euclidian normal vector n, using three variables $\mathsf{n} = (n_1, n_2, n_3)^{\mathsf{T}}$, in which case a constraint like $\mathsf{n}^{\mathsf{T}}\mathsf{n} = 1$ would be needed. This can be avoided by using the spherical coordinates θ and ϕ, i.e. $\mathsf{n} = (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta)^{\mathsf{T}}$. In the following sections, we refer to the functions g and h as G-constraint and H-constraint, respectively. Next, we give a derivation of our method for a general vector space.

In practice constraint functions are usually nonlinear. In order to work with them, we have to make a linearization at a distinct point. The new linear function will only be valid and applicable within a certain region around that tangent point. This implies two things: we are required to provide an initial point and we are supposed to stay in its vicinity. The latter is unintentionally enforced by equation (2.1). If the initial point was chosen poorly, the procedure has to be iterated. For now, parameter estimation turns into an iterative gradient descent method. Subsequently, we assume to have a fairly good initial point, which we denote by $(\hat{\mathsf{b}}_i, \hat{\mathsf{p}})$. Especially, we take the uncertain observations $\{\mathsf{b}_{1..N}\}$ as initial estimates, i.e. $\{\hat{\mathsf{b}}_{1..N}\} = \{\mathsf{b}_{1..N}\}$. We refer to $\hat{\mathsf{p}}$ and $\hat{\mathsf{b}}_i$ as the *initial estimates.* A Taylor series expansion of first order for $\mathsf{g}(\mathsf{b}_i, \mathsf{p})$ at $(\hat{\mathsf{b}}_i, \hat{\mathsf{p}})$ yields

$$\begin{aligned} & \mathsf{g}(\hat{\mathsf{b}}_i + \Delta\mathsf{b}_i,\ \hat{\mathsf{p}} + \Delta\mathsf{p}) = \mathsf{0} \\ \Longleftrightarrow \quad & \underbrace{(\partial_{\mathsf{p}}\,\mathsf{g})(\hat{\mathsf{b}}_i,\hat{\mathsf{p}})}_{\mathsf{U}_i(\hat{\mathsf{b}}_i,\hat{\mathsf{p}})}\,\Delta\mathsf{p} + \underbrace{(\partial_{\mathsf{b}_i}\,\mathsf{g})(\hat{\mathsf{b}}_i,\hat{\mathsf{p}})}_{\mathsf{V}_i(\hat{\mathsf{b}}_i,\hat{\mathsf{p}})}\,\Delta\mathsf{b}_i + \underbrace{\mathsf{g}(\hat{\mathsf{b}}_i,\hat{\mathsf{p}})}_{-\,\mathsf{c}_{g_i}} \approx \mathsf{0} \quad , \end{aligned} \tag{2.2}$$

where $\mathsf{U}_i(\hat{\mathsf{b}}_i, \hat{\mathsf{p}})$ and $\mathsf{V}_i(\hat{\mathsf{b}}_i, \hat{\mathsf{p}})$ are the Jacobians with respect to the parameters and the observations, respectively. For each observation $\hat{\mathsf{b}}_i$ we obtain a matrix equation

$$\mathsf{U}_i\,\Delta\mathsf{p} + \mathsf{V}_i\,\Delta\mathsf{b}_i = \mathsf{c}_{g_i} . \tag{2.3}$$

If we define $\mathsf{b} = (\mathsf{b}_1^\mathsf{T}, \mathsf{b}_2^\mathsf{T}, \ldots, \mathsf{b}_n^\mathsf{T})^\mathsf{T}$, we can combine all observational condition equations into

$$\underbrace{\begin{pmatrix} \mathsf{U}_1 \\ \vdots \\ \mathsf{U}_n \end{pmatrix}}_{\mathsf{U}} \Delta\mathsf{p} \;+\; \underbrace{\begin{pmatrix} \mathsf{V}_1 & & 0 \\ & \ddots & \\ 0 & & \mathsf{V}_n \end{pmatrix}}_{\mathsf{V}} \underbrace{\begin{pmatrix} \Delta\mathsf{b}_1 \\ \vdots \\ \Delta\mathsf{b}_n \end{pmatrix}}_{\Delta\mathsf{b}} = \underbrace{\begin{pmatrix} \mathsf{c}_{g_1} \\ \vdots \\ \mathsf{c}_{g_n} \end{pmatrix}}_{\mathsf{c_g}}$$

and eventually get a block matrix representation

$$\mathsf{U}\,\Delta\mathsf{p} \;+\; \mathsf{V}\,\Delta\mathsf{b} \;=\; \mathsf{c_g}\,. \tag{2.4}$$

Similarly, a Taylor series expansion of first order for the constraint function $\mathsf{h}(\mathsf{p}) = 0$ at $\hat{\mathsf{p}}$ yields

$$\underbrace{(\partial_\mathsf{p}\mathsf{h})(\hat{\mathsf{p}})}_{\mathsf{H}(\hat{\mathsf{p}})}\,\Delta\mathsf{p} + \underbrace{\mathsf{h}(\hat{\mathsf{p}})}_{-\,\mathsf{c}_h} \approx 0 \quad\Longrightarrow\quad \mathsf{H}\Delta\mathsf{p} \;=\; \mathsf{c}_h\,.$$

Condition equation (2.3) matches the *Gauss-Helmert* model, which was introduced by Helmert in 1872 as a general linear model, [4]. In the theory of parameter estimation, it is the starting point for deriving and computing the *best linear unbiased estimator* for the parameter vector p. Besides, it has been shown in [6] that different approaches, namely *least squares*, *maximum likelihood* and the approach using the Gauss-Helmert model, equally lead to the best linear unbiased estimator. The Gauss-Helmert model is also know as *mixed model with constraints*, since the observations (here $\Delta\mathsf{b}_i$) represent unfixed variables, too. It differs from the famous *Gauss-Markov* model, which is commonly used for regression problems, because the observations appear indirectly, i.e. affinely transformed (in our case $\mathsf{V}_i\,\Delta\mathsf{b}_i$). Details can be found in [6, 8].

In the following sections, we refer to our parameter estimation method as the '*Gauss-Helmert method*'.

It is crucial to note that we simultaneously search for corrections $\{\Delta\mathsf{b}_{1..N}\}$ and $\Delta\mathsf{p}$, such that

$$\mathsf{b}_i \;=\; \hat{\mathsf{b}}_i + \Delta\mathsf{b}_i \quad\text{and}\quad \mathsf{p} \;=\; \hat{\mathsf{p}} + \Delta\mathsf{p}\,,$$

where p denotes a valid parameter vector for all $\{\mathsf{b}_{1..N}\}$, i.e. $(\forall i) : \mathsf{g}(\mathsf{b}_i,\, \mathsf{p}) \;=\; 0$ *and* $\mathsf{h}(\mathsf{p}) \;=\; 0$. Moreover, in case of least squares adjustment, the corrections $\{\Delta\mathsf{b}_{1..N}\}$ should be minimal with respect to the covariance matrices $\{\Sigma_{\mathsf{b}_{1..N}}\}$. Using block matrix notation, we thus have the adjustment condition for the correction $\Delta\mathsf{b}$

$$D^2 = \Delta\mathsf{b}^\mathsf{T}\Sigma_\mathsf{b}^{-1}\Delta\mathsf{b} \;\longrightarrow\; \min\,, \tag{2.5}$$

where Σ_b is the appropriate block diagonal matrix, build up of the individual $\{\Sigma_{\mathsf{b}_{1..N}}\}$. Equation (2.5) defines the so-called *Mahalanobis* distance D. To meet all conditions and constraints while minimizing, we make use of the *Lagrange multiplier* method, where the vectors u and v denote the multipliers. We can thus define a new function $\Phi(\Delta\mathsf{p},\, \Delta\mathsf{b},\, \mathsf{u},\, \mathsf{v})$, which has to be minimized

$$\begin{aligned}\Phi(\Delta\mathsf{p},\, \Delta\mathsf{b},\, \mathsf{u},\, \mathsf{v}) \quad := \quad & \tfrac{1}{2}\,\Delta\mathsf{b}^{\mathsf{T}}\,\Sigma_{\mathsf{b}}^{-1}\,\Delta\mathsf{b} \\ - \quad & \left[\,\mathsf{U}\,\Delta\mathsf{p} + \mathsf{V}\,\Delta\mathsf{b} - \mathsf{c_g}\,\right]^{\mathsf{T}}\mathsf{u} \quad . \\ + \quad & \left[\,\mathsf{H}\,\Delta\mathsf{p} - \mathsf{c_h}\,\right]^{\mathsf{T}}\mathsf{v}\end{aligned}$$

Differentiation with respect to all variables gives us the subsequent equation system, which could already be solved

$$\begin{pmatrix} \Sigma_{\mathsf{b}}^{-1} & 0 & \mathsf{V}^{\mathsf{T}} & 0 \\ 0 & 0 & \mathsf{U}^{\mathsf{T}} & -\mathsf{H}^{\mathsf{T}} \\ \mathsf{V} & \mathsf{U} & 0 & 0 \\ 0 & \mathsf{H} & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta\mathsf{b} \\ \Delta\mathsf{p} \\ \mathsf{u} \\ \mathsf{v} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathsf{c_g} \\ \mathsf{c_h} \end{pmatrix} . \tag{2.6}$$

Nevertheless, the system can be considerably reduced by a sequence of substitutions: $\mathsf{N} = \sum\limits_{i=1}^{n} \mathsf{U}_i^{\mathsf{T}} \left(\mathsf{V}_i\ \Sigma_{\mathsf{b}_i}\ \mathsf{V}_i^{\mathsf{T}}\right)^{-1} \mathsf{U}_i$ and $\mathsf{c_n} = \sum\limits_{i=1}^{n} \mathsf{U}_i^{\mathsf{T}} \left(\mathsf{V}_i\ \Sigma_{\mathsf{b}_i}\ \mathsf{V}_i^{\mathsf{T}}\right)^{-1} \mathsf{c}_{\mathsf{g}_i}$. The resultant matrix equation

$$\underbrace{\begin{pmatrix} \mathsf{N} & \mathsf{H}^{\mathsf{T}} \\ \mathsf{H} & 0 \end{pmatrix}}_{\mathsf{Q}} \begin{pmatrix} \Delta\mathsf{p} \\ \mathsf{v} \end{pmatrix} = \begin{pmatrix} \mathsf{c_n} \\ \mathsf{c_h} \end{pmatrix} \tag{2.7}$$

is free of $\Delta\mathsf{b}$ and can be solved for $\Delta\mathsf{p}$. As an important by-product, the (pseudo-) inverse Q^{-1} contains the covariance matrix $\Sigma_{\Delta\mathsf{p}} = \Sigma_{\mathsf{p}}$, belonging to the estimated parameter vector p. This is proven in [6].

The correction $\Delta\mathsf{b}_i$ has to be evaluated by substituting $\Delta\mathsf{p}$ into the equation

$$\Delta\mathsf{b}_i = \Sigma_{\mathsf{b}_i}\mathsf{V}_i^{\mathsf{T}}\ (\mathsf{V}_i\Sigma_{\mathsf{b}_i}\mathsf{V}_i^{\mathsf{T}})^{-1}\ (\mathsf{c}_{\mathsf{g}_i} - \mathsf{U}_i\,\Delta\mathsf{p}).$$

The new estimates for b_i and p are then given by $\hat{\mathsf{p}}' = \hat{\mathsf{p}} + \Delta\mathsf{p}$ and $\hat{\mathsf{b}}_i' = \hat{\mathsf{b}}_i + \Delta\mathsf{b}_i$. If the constraint functions $\mathsf{g}(\mathsf{b}_i,\, \mathsf{p})$ and $\mathsf{h}(\mathsf{p})$ are linear, then these new estimates are the best linear unbiased estimates for b_i and p. If the constraint functions are not linear, then this is a step in an iterative estimation procedure.

3. Geometric Algebra

The Geometric Algebra over an n-dimensional Euclidean vector space $\mathbb{R}^n$ has dimension 2^n and is denoted by $\mathbb{G}(\mathbb{R}^n)$ or simply $\mathbb{G}_n$.

The corresponding 2^n-dimensional basis of $\mathbb{G}_n$ contains the n basis vectors $\{\mathbf{e}_{1..n}\}$ of $\mathbb{R}^n$, representing elements of grade one. An Euclidian vector $\boldsymbol{a} = (a_1, \ldots, a_n)^{\mathsf{T}}$ has therefore an equivalent in $\mathbb{G}_n$ that we denote $\boldsymbol{a} = \sum_i^n a_i\mathbf{e}_i$ as well. Elements of different grade of the algebra can be constructed by the *outer product* of linearly independent vectors. For example, if $\{\boldsymbol{a}_{1..k}\}$ are a set of linearly independent vectors, then $\boldsymbol{A}_{\langle k\rangle} := \boldsymbol{a}_1 \wedge \ldots \wedge \boldsymbol{a}_k$ is an element of $\mathbb{G}_n$ of grade k, which is called a *blade*, where '$\wedge$' denotes the outer product. A general element of the algebra, called *multivector*, can always be expressed as a linear combination of blades of possibly different grades. Blades can be used to represent geometric entities. Apart from representing geometric entities by blades, it is also possible to define operators in Geometric Algebra. The class of operators we are particularly interested in are called *versors*. A versor $\boldsymbol{V} \in \mathbb{G}_n$ is a multivector that satisfies the following condition: for any blade $\boldsymbol{A}_{\langle k \rangle} \in \mathbb{G}_n$, $\boldsymbol{V}\,\boldsymbol{A}_{\langle k \rangle}\,\tilde{\boldsymbol{V}}$ is also of grade k, i.e. a versor is *grade preserving*. Especially, if $\boldsymbol{V}\tilde{\boldsymbol{V}} = 1$ holds, then $\boldsymbol{V}$ is called *unitary*. The expression $\tilde{\boldsymbol{V}}$ denotes the *reverse* of $\boldsymbol{V}$. The reverse operation changes the sign of the constituent blade elements depending on their grade, which has an effect similar to conjugation in quaternions. The most interesting versors for our purposes in conformal space are rotation operators (rotors), translation operators (translators) and scaling operators (dilators). All of them share the property that they can be applied to all geometric entities in the same way. That is, it does not matter whether a blade $\boldsymbol{A}_{\langle k \rangle}$ represents a point, line, plane, circle or sphere; applying, for example, a rotor $\boldsymbol{R}$ to it, as in $\boldsymbol{R}\,\boldsymbol{A}_{\langle k \rangle}\,\tilde{\boldsymbol{R}}$, results in a blade that represents the rotated geometric entity.

3.1. Transferring Algebra Expressions to $\mathbb{R}^{2^n}$

Subsequently, we show how Geometric Algebra expressions can be transferred to a matrix algebra, where we make use of the inherent tensor representation of Geometric Algebra.

If $\{\boldsymbol{E}_{1..2^n}\} \supset \{\mathbf{e}_{1..n}\}$ denotes the 2^n-dimensional algebra basis of $\mathbb{G}_n$, then a multivector $\boldsymbol{A} \in \mathbb{G}_n$ can be written as $\boldsymbol{A} = \mathsf{a}^i\,\boldsymbol{E}_i$, where a^i denotes the i^{th} component of a vector $\mathsf{a} \in \mathbb{R}^{2^n}$ 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 an algebra product and $\mathsf{G}^k{}_{ij} \in \mathbb{R}^{2^n \times 2^n \times 2^n}$ is a tensor encoding this product.

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 $\mathsf{c} = \mathsf{U}(\mathsf{a})\,\mathsf{b} = \mathsf{V}(\mathsf{b})\,\mathsf{a}$. Therefore, we can define an isomorphism Φ, such that for $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{G}_n$, $\Phi(\boldsymbol{A}) \in \mathbb{R}^{2^n}$ and $\Phi(\boldsymbol{A} \circ \boldsymbol{B}) = \mathsf{U}(\Phi(\boldsymbol{A}))\,\Phi(\boldsymbol{B}) = \mathsf{V}(\Phi(\boldsymbol{B}))\,\Phi(\boldsymbol{A})$. In particular, using Φ we can write $\Phi(\boldsymbol{a}) = \mathsf{a}$ and $\Phi(\boldsymbol{b}) = \mathsf{b}$. The isomorphism allows us to apply standard numerical algorithms to Geometric Algebra equations. We can also reduce the complexity of the equations considerably by only mapping those components of multivectors that are actually needed. In the following we therefore assume that Φ maps to the minimum number of components necessary.

4. Estimation in Geometric Algebra

In order to benefit from the variety of geometric entities, we use the well known Geometric Algebra of the (projective) *conformal space* of Euclidean 3D-space, see [7, 1].

The embedding function[2] $\mathcal{K}$ is defined as (cf. [7, 12])

$$\mathcal{K}:\ \boldsymbol{x} \;=\; (x_1, x_2, x_3)^{\mathsf{T}} \;\longmapsto\; \boldsymbol{X} := \boldsymbol{x} + \tfrac{1}{2}\,\boldsymbol{x}^2\,\mathbf{e}_\infty + \mathbf{e}_o\,, \tag{4.1}$$

$\mathcal{K}$ maps $\boldsymbol{x} \in \mathbb{R}^3$ to $\boldsymbol{X} \in \mathbb{R}^{4,1}$, the basis of which can be written as $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_\infty, \mathbf{e}_o\}$. Note that the linear combinations $\mathbf{e}_\infty = \mathbf{e}_- + \mathbf{e}_+$ and $\mathbf{e}_o = \frac{1}{2}(\mathbf{e}_- - \mathbf{e}_+)$ substitute the basis vectors $\mathbf{e}_-$ and $\mathbf{e}_+$, respectively. Due to $\mathbf{e}_-^2 = -1$ and $\mathbf{e}_+^2 = +1$, we have $\mathbf{e}_\infty^2 = \mathbf{e}_o^2 \overset{!}{=} 0$.

In the scope of parameter estimation in conformal space, we have to obey the rules of error propagation when embedding the Euclidian observations into conformal space. Assume that vector $\boldsymbol{x}$ is a random vector variable with a Gaussian distribution and $\bar{\boldsymbol{x}}$ is its mean value. Furthermore, we denote the 3×3 covariance matrix of $\boldsymbol{x}$ by $\Sigma_{\boldsymbol{x}}$. Let $\mathcal{E}$ denote the expectation value operator, i.e. $\mathcal{E}[\boldsymbol{x}] = \bar{\boldsymbol{x}}$. Then we make the approximation that

$$\mathcal{E}\big[\mathcal{K}(\boldsymbol{x})\big] \;=\; \bar{\boldsymbol{x}} + \tfrac{1}{2}\,\bar{\boldsymbol{x}}^2\,\mathbf{e}_\infty + \mathbf{e}_o + \tfrac{1}{2}\mathrm{trace}\big(\Sigma_{\boldsymbol{x}}\big)\mathbf{e}_\infty \quad\approx\quad \mathcal{K}\big(\mathcal{E}[\boldsymbol{x}]\big). \tag{4.2}$$

We still need to know what the corresponding covariance matrix for $\boldsymbol{X} = \mathcal{K}(\bar{\boldsymbol{x}})$ is. This is easily done employing Gaussian error propagation, which is, for the case of matrix notation, shown in [8]. We first evaluate the Jacobian of $\mathcal{K}$,

$$\mathsf{J}_{\mathcal{K}}(\bar{\boldsymbol{x}}) := \frac{\partial \mathcal{K}}{\partial \boldsymbol{X}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \bar{x}_1 & \bar{x}_2 & \bar{x}_3 \\ 0 & 0 & 0 \end{pmatrix}. \tag{4.3}$$

The covariance matrix $\Sigma_{\boldsymbol{X}}$ of $\boldsymbol{X}$ for the conformal space is then given by

$$\Sigma_{\boldsymbol{X}} = \mathsf{J}_{\mathcal{K}}(\bar{\boldsymbol{x}})\;\Sigma_{\boldsymbol{x}}\;\mathsf{J}_{\mathcal{K}}^{\mathsf{T}}(\bar{\boldsymbol{x}}). \tag{4.4}$$

Next, we show exemplarily that Geometric Algebra offers a unified framework to derive the constraint equations for geometrical problems, so that the Gauss-Helmert method can be applied. Since the standard algebra operations between multivectors can be mapped to *bilinear* functions, the estimation of all algebra elements is basically the same. This means in particular that operators as well as geometric entities are represented by vectors and their estimation is therefore very similar. In order to apply the Gauss-Helmert method as described in section 2, we need to define a G-constraint function that relates the parameter vector and the data vectors, as well as a H-constraint function for the parameter vector p alone. Subsequently, we assume to have an initial estimate $\hat{\mathsf{p}}$ of the parameter vector.

[2] Vectors of $\mathbb{G}_n$, representing *conformal points*, are symbolized by capital letters, like general multivectors.

4.1. Fitting a Circle in 3D

Now we show how the Gauss-Helmert method can be used in Geometric Algebra to fit a circle in 3D-space to a set of N data points $\{\boldsymbol{b}_{1..N}\}$. Each data point is given with its mean $\boldsymbol{b}_i$ and covariance matrix $\Sigma_{\boldsymbol{b}_i}$, assuming a Gaussian distribution.

We represent a circle in conformal space by the inner product null space $\mathbb{X}$ of a 2-blade $\boldsymbol{C}$. That space consists of all conformal points $\boldsymbol{X}$, the inner product of which with the circle $\boldsymbol{C}$ is equal 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 $\boldsymbol{S}_r$ 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 $\boldsymbol{S}_r = \boldsymbol{m} + \frac{1}{2}(\boldsymbol{m}^2 - r^2)\,\mathbf{e}_\infty + \mathbf{e}_o$. Given some vector $\boldsymbol{x}$, it can be verified that $\mathcal{K}(\boldsymbol{x}) \cdot \boldsymbol{S}_r = 0 \iff \|\boldsymbol{x} - \boldsymbol{m}\|_2 = r$.

Intuitively speaking, given two intersecting spheres $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, the set of circle points consists of all points $\boldsymbol{X}$ that lie on sphere $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. In Conformal Geometric Algebra intersection can mostly be expressed by simply taking the outer product; the intersection circle is thus defined by $\boldsymbol{C} = \boldsymbol{S}_1 \wedge \boldsymbol{S}_2$. According to the above discussion we build the inner product with a point $\boldsymbol{X}$

$$\boldsymbol{X} \cdot (\boldsymbol{S}_1 \wedge \boldsymbol{S}_2) = \underbrace{(\boldsymbol{X} \cdot \boldsymbol{S}_1)}_{\in\mathbb{R}}\boldsymbol{S}_2 - \underbrace{(\boldsymbol{X} \cdot \boldsymbol{S}_2)}_{\in\mathbb{R}}\boldsymbol{S}_1\,. \tag{4.5}$$

The terms cannot cancel each other, if $\boldsymbol{S}_1$ 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 circles inner product null space itself. With the help of isomorphism Φ, introduced at the end of section 3, it remains to transfer the inner product expression $\boldsymbol{X} \cdot \boldsymbol{C}$ to an equivalent matrix expression. Note that the inner product of a vector with a bivector results in a vector. Since a vector has five components, there are five constraints for each point that has to lie on a circle.

We use a ten component parameter vector p to represent circle $\boldsymbol{C}$, since all ten basis blades of grade two in $\mathbb{G}_5$ contribute to the definition of a circle: $\Phi(\boldsymbol{C}) = \mathsf{p} \in \mathbb{R}^{10}$. The observed points $\{\boldsymbol{b}_{1..N}\}$ are embedded by $\mathcal{K}$ and then mapped by Φ as follows: $\Phi(\mathcal{K}(\boldsymbol{b}_i) = \boldsymbol{B}_i) = \mathsf{b}_i \in \mathbb{R}^5$. Similarly and according to equation (4.4), the covariance matrices are embedded to 5×5 matrices.

The application of Φ to our constraint results in

$$\Phi(\boldsymbol{B}_i \cdot \boldsymbol{C}) = \mathsf{U}(\mathsf{b}_i)\,\mathsf{p} = \mathsf{V}(\mathsf{p})\,\mathsf{b}_i \tag{4.6}$$

$$= \mathsf{g}(\mathsf{b}_i, \mathsf{p})\,, \tag{4.7}$$

which can be differentiated easily. The Jacobians U and V, which are required by equation (2.2) for the Gauss-Helmert method, follow therefore implicitly from the bilinearity of Geometric Algebra products.

A circle in Euclidian space can be described by a minimum number of six parameters (three for the center, two for the circle plane and one for the radius).

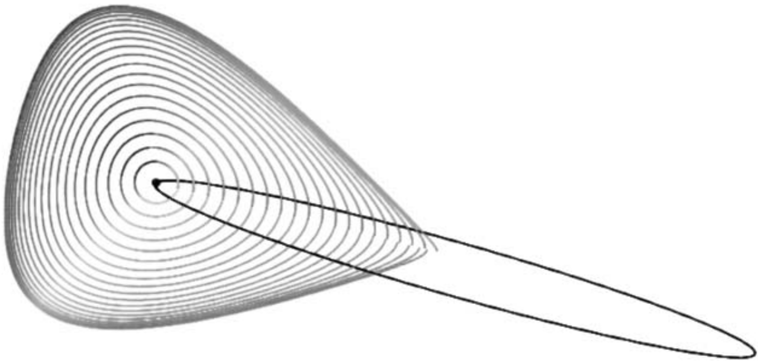

Figure 1. Metric induced by the inner product between a circle and points. Each of the concentric curves visualizes a locus of constant distance regarding the inner product.

Hence, we deal with a functional dependency of grade $4 = 10 - 6$ within the parameter vector p. As mentioned in section 2, 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 infer matrix H.

Having computed the matrices U, V and H we are able to apply the Gauss-Helmert method to solve for the corrections $\Delta\mathsf{p}$ and $\Delta\mathsf{b}$. Experimental results for the circle fit problem are given in section 5.1.

It is worth noting that the described approach uses algebraic fitting, where the inner product of the G-constraint imposes a non-Euclidian metric. The latter is visualized in figure 1 with the software *CLUCalc*, [10]. Note that the curves are equidistant with respect to the inner product. Points inside the circle disc, for example, seem to be subject to the smallest gradient. The metric resembles only locally, close to the circle, a Euclidian one.

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 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. Figures 2 and 3 exemplarily show the uncertainty of an estimated circle and an estimated line (a circle with radius infinity). The surrounding tubes, indicated by slices, show the standard deviation of the estimates.

4.2. Fitting two Point Clouds in 3D

In this subsection, we describe how the Gauss-Helmert method can be used to estimate a rigid body motion. Those operations are given by *motors* in Geometric Algebra. A motor extends a rotation ($\rightarrow$ rotor) by a translational component along the axis of rotation. Hence, we can think of motors as screw motions (cf. [14]). In the scope of *pose estimation*, a *pose* is uniquely characterized by a rigid body

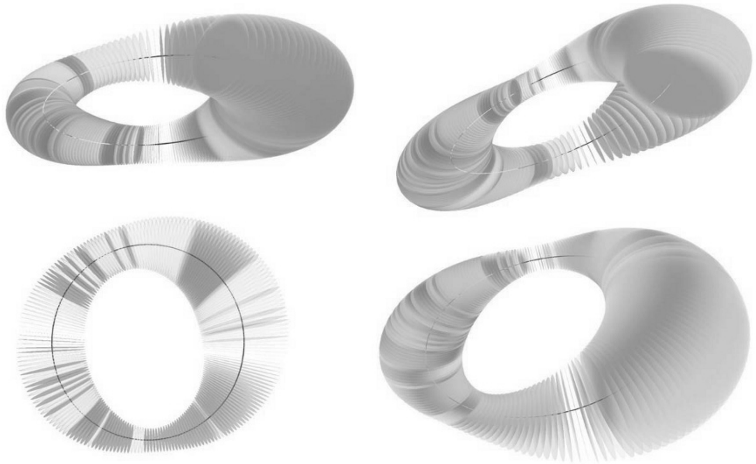

FIGURE 2. Fitting a circle: the uncertainty, i.e. standard deviation, of an estimated circle in four different views. The image in the lower left depicts a top view of the circle.

FIGURE 3. The uncertainty of a line, indicated by a twisted elliptic tube.

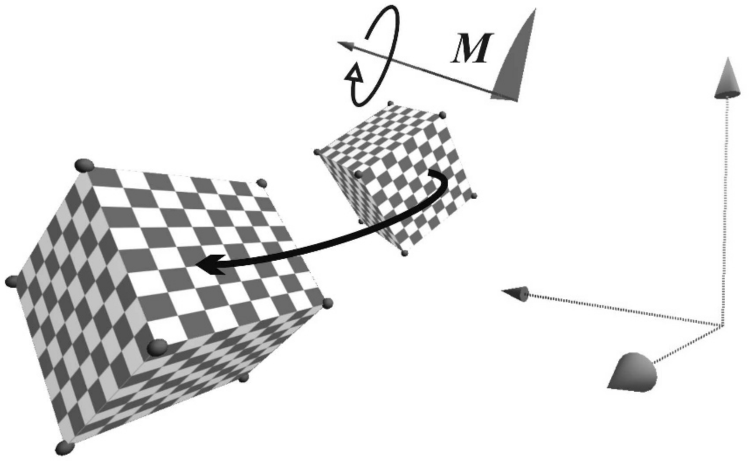

FIGURE 4. Fitting two point clouds: the rotation of the interrelating motor is indicated by the partial disc, whereas the axis of rotation with translation is given by the arrow attached to the disc.

motion and can thus be represented by a motor. The estimation of motors is the first step towards the perspective pose estimation problem.

Subsequently , we assume to have two sets of N Euclidian points $\{\boldsymbol{a}_{1..N}\}$ and $\{\boldsymbol{b}_{1..N}\}$. The latter represent the observations, the covariance matrices of which are given by $\{\Sigma_{\boldsymbol{b}_{1..N}}\}$. The set $\{\boldsymbol{a}_{1..N}\}$ is assumed to have no uncertainty[3]. Let $\boldsymbol{A}_i = \mathcal{K}(\boldsymbol{a}_i)$ and $\boldsymbol{B}_i = \mathcal{K}(\boldsymbol{b}_i)$ denote the embeddings of $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$, respectively. Besides, we set $\Phi(\boldsymbol{A}_i) = \mathsf{a}_i$, $\Phi(\boldsymbol{B}_i) = \mathsf{b}_i$ and $\Phi(\boldsymbol{M}) = \mathsf{p}$. We search for the motor $\boldsymbol{M}$, which best transforms the corresponding conformal points $\{\boldsymbol{A}_{1..N}\}$ to $\{\boldsymbol{B}_{1..N}\}$. The scenario is shown in figure 4. Using Geometric Algebra, we can easily write $\boldsymbol{M}\boldsymbol{A}_i\widetilde{\boldsymbol{M}} = \boldsymbol{B}_i$, cf. [13]. 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

$$\begin{array}{ccccccccc} (\boldsymbol{M} & & \boldsymbol{A}_i) & - & (\boldsymbol{B}_i & & \boldsymbol{M}) & = & 0 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \mathsf{p}^k & \mathsf{G}^t{}_{kl} & \mathsf{a}_i{}^l & - & \mathsf{b}_i{}^l & \mathsf{G}^t{}_{lk} & \mathsf{p}^k & = & 0^t \end{array} . \tag{4.8}$$

The tensor G encodes the geometric product. In order to evaluate the matrices U and V, we differentiate equation (4.8) 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 $\mathsf{V}(\mathsf{p}) = -\,\mathsf{p}^k\mathsf{G}^t{}_{lk}$.

We parametrize $\boldsymbol{M}$ in terms of vector $\mathsf{p} \in \mathbb{R}^8$ (a motor is compounded of a scalar part, six bivectors and one quadvector). Since a rigid body motion is defined

[3]For example, it can be regarded as a wire frame model of some object, the observations stem from.

by at least six parameters, we need to have constraints on the parameters, again. We choose $\mathsf{h}(\mathsf{p}) = \Phi(\boldsymbol{M}\widetilde{\boldsymbol{M}} - 1) = \mathsf{p}^k\, \mathsf{p}^l\, \mathsf{R}^m{}_l\, \mathsf{G}^t{}_{km} - \delta^{t,1}$ to be the H-constraint. The tensor R encodes the reverse operation and $\delta^{t,1}$ is zero, except for $t = 1$. Differentiation yields $\mathsf{H}(\mathsf{p}) = \mathsf{p}^l\, (\mathsf{R}^m{}_l\, \mathsf{G}^t{}_{km} + \mathsf{R}^m{}_k\, \mathsf{G}^t{}_{lm})$.

By simply substituting the matrices U, V and H into the respective equations given in the theoretical part dealing with the Gauss-Helmert method, we can compute the estimate for $\boldsymbol{M}$. Experimental results regarding the motor estimation are presented in section 5.2.

4.3. Perspective Pose Estimation

Perspective Pose estimation consists of determining the orientation and position of an internally calibrated camera (see e.g. [9]), given a 3D-model in a scene and a set of N correspondence points from an image of that scene. The model serves as a reference to an external (world) coordinate system. If we determine the models position and orientation with respect to the camera coordinates, we are able to infer the pose of the camera. The interrelation is a rigid body motion. Thus, we describe the pose in terms of a motor $\boldsymbol{M}$ [14, 15]. A typical setup is shown in figure 5; three model points are fitted to their respective projection rays.

Let $\{\boldsymbol{a}_{1..N}\}$ be a set of N Euclidian points. They are assumed to have no uncertainty, for they represent a known 3D-model of some object $\mathcal{O}$. Furthermore, let $\{\boldsymbol{B}_{1..N}\}$ be an equal set of corresponding projection rays. They are evaluated by $\boldsymbol{B}_i = (\mathbf{e}_\infty \wedge \mathbf{e}_o \wedge \mathcal{K}(\boldsymbol{b}_i))I$, where $\{\boldsymbol{b}_{1..N}\}$ denotes the set of observed image points of $\mathcal{O}$. Here, element $I \in \mathbb{G}_5$ is the *pseudoscalar* and basis vector $\mathbf{e}_o$ plays the role of the origin, i.e. $\mathcal{K}((0,0,0)^\mathsf{T}) = \mathbf{e}_o$. Note that the projection rays are taken as new observations. In order to provide covariance matrices for them, we have to embed the covariance matrices belonging to the image points $\{\boldsymbol{b}_{1..N}\}$. This is done using equation (4.4). Next, we have to compute the covariance matrices for the lines $\{\boldsymbol{B}_{1..N}\}$, which can be achieved easily employing error propagation of first order.

Using $\mathcal{K}(\boldsymbol{a}_i) = \boldsymbol{A}_i$, we formulate the G-constraint

$$(\boldsymbol{M}\boldsymbol{A}_i\widetilde{\boldsymbol{M}}) \cdot \boldsymbol{B}_i = 0\,, \tag{4.9}$$

where we use the fact that the inner product of a point and a line is zero, *iff* the point is located on the line. Again, we set $\Phi(\boldsymbol{M}) = \mathsf{p}$. The preceding equation can then be translated to the tensor expression

$$\mathsf{p}^k\, \mathsf{p}^l\, \mathsf{a}_i{}^r\, \mathsf{b}_i{}^s\, \Pi^t{}_{klrs} = 0^t\,, \tag{4.10}$$

where the product tensor Π is given by $\Pi^t{}_{klrs} = \mathsf{G}^a{}_{kr}\, \mathsf{G}^c{}_{ab}\, \mathsf{R}^b{}_l\, \mathsf{N}^t{}_{cs}$. Here, G, R and N denote the tensors encoding the geometric product, the reverse operation and the inner product, respectively. We do not give explicit expressions for $\mathsf{U}(\mathsf{b}_i)$, $\mathsf{V}(\mathsf{p})$ and $\mathsf{H}(\mathsf{p})$, but state that we obtain a set of different matrices $\{\mathsf{V}_{1..N}\}$ and that matrix U depends on p, as well. We can again use the H-constraint given in the previous subsection.

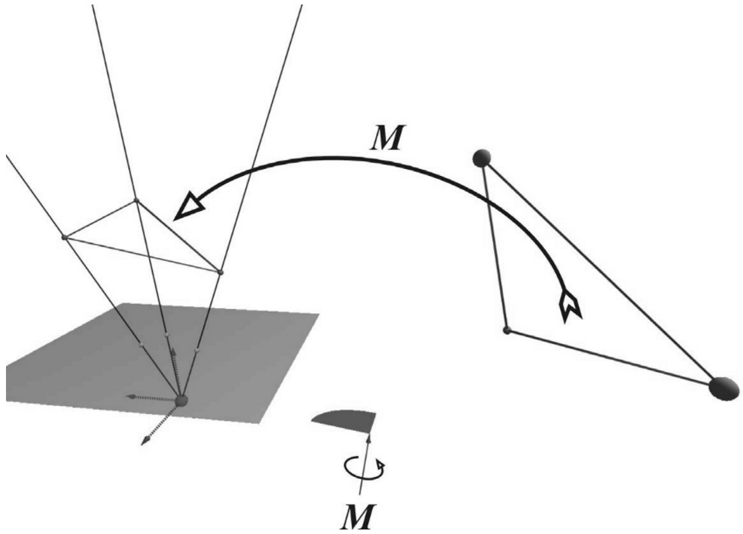

FIGURE 5. Pose estimation: fitting model points to corresponding projection rays in 3D. An image plane and a motor $\boldsymbol{M}$, fitting the triangle to the rays, are drawn as well.

In section 5.3 we report results for the pose estimation concerning synthetic setups.

5. Experimental Results

To show the quality of the proposed estimation method of geometric entities and operators, we present three synthetic experiments. In the first experiment we fit 3D-circles to uncertain data points and in the second experiment we estimate general rotations between two 3D-point clouds. The third experiment deals with the perspective pose estimation problem. Note that the presented values were obtained from an iterative application of the Gauss-Helmert method. Usually four iterations were needed to reach a stable result, but never more than ten. If the magnitude of the correction vector $\Delta\mathsf{p}$ for the parameters falls below 10^{-11}, we call the estimate a stable result.

In two experiments, we compare the Gauss-Helmert method with a *singular value decomposition* (SVD) based approach [4]. It is a purely algebraic approach, which implicitly arises from the G-constraint, if the latter is merely linear dependent on the parameter vector p; the differentiation of $\mathsf{g}(\mathsf{b}_i, \mathsf{p})$ with respect to p results then in matrix $\mathsf{U}(\mathsf{b}_i)$, see e.g. equation (4.7). We obtain

$$(\forall i, {\scriptstyle 1 \leq i \leq N}) : \quad \mathsf{g}(\mathsf{b}_i, \mathsf{p}) \;=\; \mathsf{U}(\mathsf{b}_i)\mathsf{p} \;\stackrel{!}{=}\; \mathsf{0}\,.$$

[4]Besides, the SVD approach provides the respective initial estimates for the Gauss-Helmert method

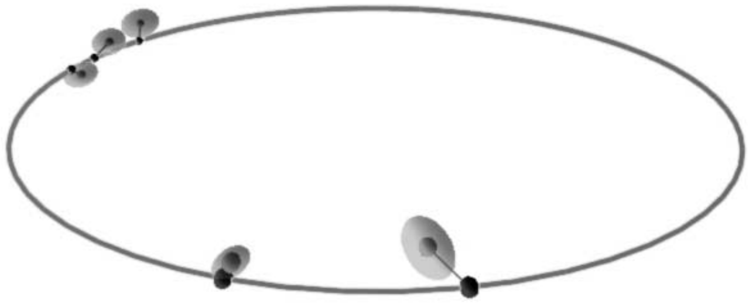

FIGURE 6. True circle with five randomly selected points on it. The generated noisy points are drawn with their respective uncertainty ellipsoids. The angle range was set to 180°.

Hence, p has to be a common element of the right null spaces of each matrix $\mathsf{U}_i := \mathsf{U}(\mathsf{b}_i)$. This is equivalent to p being part of the right null space of the matrix $\mathsf{S} = \sum_{i=1}^{N} \mathsf{U}_i^{\mathsf{T}} \mathsf{U}_i$. Such null space vectors can be computed easily using an SVD. Note that in the presence of noise, the column vector in K_r corresponding to the smallest singular value in D has to be selected, if $\mathsf{S} = \mathsf{K}_l \mathsf{D} \mathsf{K}_r^{\mathsf{T}}$. Next, we report our results.

5.1. Fitting a Circle in 3D

To generate the uncertain data to which a circle is to be fitted, we first create a "true" circle $\boldsymbol{C}$ of radius one, oriented arbitrarily in 3D-space. We then randomly select N points $\{\boldsymbol{a}_{1..N}\}$ on the true circle within a given angle range (please note that in this section, we only work with Euclidian points). For each of these points a covariance matrix is generated randomly, within a certain range. This yields $\{\Sigma_{\boldsymbol{a}_{1..N}}\}$. For each of the $\boldsymbol{a}_i$, $\Sigma_{\boldsymbol{a}_i}$ is used to generate a Gaussian distributed random error vector $\boldsymbol{r}_i$. The data points $\{\boldsymbol{b}_{1..N}\}$ with corresponding covariance matrices $\{\Sigma_{\boldsymbol{b}_{1..N}}\}$ are then given by $\boldsymbol{b}_i = \boldsymbol{a}_i + \boldsymbol{r}_i$ and $\Sigma_{\boldsymbol{b}_i} = \Sigma_{\boldsymbol{a}_i}$, see figure 6. The standard deviation of the set $\{\|\boldsymbol{r}\|_{1..N}\}$ will be denoted by σ_r. For each angle range, 30 sets of true points $\{\boldsymbol{a}_{1..N}\}$ and for each of these sets, 40 sets of data points $\{\boldsymbol{b}_{1..N}\}$ were generated.

A circle is then fitted to each of the data point sets. We will denote a circle estimate by $\widehat{\boldsymbol{C}}$ and the shortest Euclidian vector between a true point $\boldsymbol{a}_i$ and $\widehat{\boldsymbol{C}}$ by $\boldsymbol{d}_i$. For each $\widehat{C}$ we then evaluate two quality measures: the Euclidean RMS distance $\delta_E := \sqrt{\sum_{i=1}^{N} \boldsymbol{d}_i^{\mathsf{T}} \boldsymbol{d}_i / N}$ and the Mahalanobis RMS distance $\delta_\Sigma := \sqrt{\sum_{i=1}^{N} \boldsymbol{d}_i^{\mathsf{T}} \Sigma_{\boldsymbol{b}_i} \boldsymbol{d}_i / N}$. The latter measure uses the covariance matrices as local metrics for the distance measure. δ_Σ is a unit-less value that is > 1, $= 1$ or < 1 if d_n lies outside, on or inside the standard deviation error ellipsoid represented by $\Sigma_{\boldsymbol{b}_i}$. For each true point set, the mean and standard deviation of the δ_E and δ_Σ over all data point sets is denoted by Δ_E, σ_E and Δ_Σ, σ_Σ, respectively. Finally,

σ_r	Angle Range	$\bar{\Delta}_\Sigma$ ($\bar{\sigma}_\Sigma$) SVD	$\bar{\Delta}_\Sigma$ ($\bar{\sigma}_\Sigma$) GH	$\bar{\Delta}_E$ ($\bar{\sigma}_E$) SVD	$\bar{\Delta}_E$ ($\bar{\sigma}_E$) GH
0.07	10°	2.13 (0.90)	1.26 (0.52)	0.047 (0.015)	0.030 (0.009)
	60°	1.20 (0.44)	0.92 (0.31)	0.033 (0.010)	0.028 (0.009)
	180°	1.38 (0.56)	0.97 (0.36)	0.030 (0.009)	0.025 (0.008)
0.15	10°	2.17 (0.90)	1.15 (0.51)	0.100 (0.032)	0.057 (0.019)
	60°	1.91 (0.99)	1.35 (0.68)	0.083 (0.033)	0.069 (0.028)
	180°	1.21 (0.44)	0.90 (0.30)	0.070 (0.022)	0.058 (0.018)

Table 1. Results of circle estimation for SVD method (SVD) and Gauss-Helmert method (GH).

σ_r	$\bar{\Delta}_\Sigma$ ($\bar{\sigma}_\Sigma$) Std	$\bar{\Delta}_\Sigma$ ($\bar{\sigma}_\Sigma$) SVD	$\bar{\Delta}_\Sigma$ ($\bar{\sigma}_\Sigma$) GH	$\bar{\Delta}_E$ ($\bar{\sigma}_E$) Std	$\bar{\Delta}_E$ ($\bar{\sigma}_E$) SVD	$\bar{\Delta}_E$ ($\bar{\sigma}_E$) GH
0.09	1.44 (0.59)	1.47 (0.63)	0.68 (0.22)	0.037 (0.011)	0.037 (0.012)	0.024 (0.009)
0.18	1.47 (0.62)	1.53 (0.67)	0.72 (0.25)	0.078 (0.024)	0.079 (0.026)	0.052 (0.019)

Table 2. Result of general rotation estimation for standard method (Std), SVD method (SVD) and Gauss-Helmert method (GH).

we take the mean of the Δ_E, σ_E and Δ_Σ, σ_Σ over all true point sets, which are then denoted by $\bar{\Delta}_E$, $\bar{\sigma}_E$ and $\bar{\Delta}_\Sigma$, $\bar{\sigma}_\Sigma$. These quality measures are evaluated for the circle estimates by the SVD and the Gauss-Helmert (GH) method. In table 1 the results for different values of σ_r and different angle ranges is given. In all cases 10 data points are used.

It can be seen that for different levels of noise (σ_r) the Gauss-Helmert method always performs better in the mean quality and the mean standard deviation than the SVD method. It is also interesting to note that the Euclidean measure $\bar{\Delta}_E$ is approximately doubled when σ_r is doubled, while the "stochastic" measure $\bar{\Delta}_\Sigma$, only increases slightly. This is to be expected, since an increase in σ_r implies larger values in the $\{\Sigma_{\boldsymbol{b}_{1..N}}\}$. Note that $\bar{\Delta}_\Sigma < 1$ implies that the estimated circle lies mostly inside the standard deviation ellipsoids of the true points.

5.2. Fitting two Point Clouds in 3D

For the evaluation of a general rotor, the "true" points $\{\boldsymbol{a}_{1..N}\}$ are a cloud of Gaussian distributed points about the origin with standard deviation 0.8. These points are then transformed by a "true" general rotation $\boldsymbol{R}$. Given the set $\{\boldsymbol{a}'_{1..N}\}$ of rotated true points, noise is added to generate the data points $\{\boldsymbol{b}_{1..N}\}$ in just the same way as for the circle. For each of 40 sets of true points, 40 data point sets are generated and a general rotor $\widehat{\boldsymbol{R}}$ is estimated. Using $\widehat{\boldsymbol{R}}$ the true points are rotated to give $\{\hat{\boldsymbol{a}}'_{1..N}\}$. The distance vectors $\{\boldsymbol{d}_{1..N}\}$ are then defined as $\boldsymbol{d}_i := \boldsymbol{a}'_i - \hat{\boldsymbol{a}}'_i$. From the $\{\boldsymbol{d}_{1..N}\}$ the same quality measures as for the circle are evaluated. In table 2 we compare the results of the Gauss-Helmert (GH) method with the initial SVD estimate and a standard approach (Std) described in [2]. Since the quality measures did not give significantly different results for rotation angles between 3 and 160 degrees, the mean of the respective values over all rotation angles are shown in the table. The rotation axis always points along the z-axis and is moved one unit away from the origin along the x-axis. In all experiments 10 points are used.

It can be seen that for different levels of noise (σ_r) the Gauss-Helmert method always performs significantly better in the mean quality and the mean standard deviation than the other two. Just as for the circle the Euclidean measure $\bar{\Delta}_E$ is approximately doubled when σ_r is doubled, while the "stochastic" measure $\bar{\Delta}_\Sigma$, only increases slightly. Note that $\bar{\Delta}_\Sigma < 1$ implies that the points $\{\hat{\boldsymbol{a}}'_{1..N}\}$ lie mostly inside the standard deviation ellipsoids of the $\{\boldsymbol{a}'_{1..N}\}$.

5.3. Perspective Pose Estimation

The assumed imaging geometry, in case of the pose estimation, basically resembles a normalized one [9]: the optical axis is aligned to the x-coordinate, the focal point is supposed to be at the origin $(0, 0, 0)$ and the image plane is centered at $(1, 0, 0)$, i.e. it has unit distance to the focal point. In the beginning we create a motor $\boldsymbol{M}_t$, which we use to form our scenarios. Next, we generate a cloud of N Gaussian distributed points $\{\boldsymbol{b}_{1..N}\}$ with a standard deviation of $\sqrt{2}$ and a bias of $(7, 0, 0)$. We use $\widetilde{\boldsymbol{M}_t}$ to transform $\{\boldsymbol{b}_{1..N}\}$, yielding the cloud $\{\boldsymbol{a}_{1..N}\}$, which can be thought of as our reference model. Like in the preceding experiments, we randomly generate a set of N covariance matrices $\{\Sigma_{\boldsymbol{b}_{1..N}}\}$ to account for image noise. For each $\boldsymbol{b}_i$ the corresponding $\Sigma_{\boldsymbol{b}_i}$ is used to generate a Gaussian distributed random error vector $\boldsymbol{r}_i$. Eventually, we build the noisy values $\boldsymbol{b}'_i = \boldsymbol{b}_i + \boldsymbol{r}_i$. They are then taken for computing the projection rays $\{\boldsymbol{B}_{1..N}\}$ and the image points, respectively. It is important, that the $\{\Sigma_{\boldsymbol{b}_{1..N}}\}$ are generated, such that none of them introduces an uncertainty parallel to the optical axis. Hence, there is no uncertainty perpendicular to the image plane, as desired for image points. We estimate the motor $\widehat{\boldsymbol{M}}$, which fits the $\{\boldsymbol{a}_{1..N}\}$ to the corresponding $\{\boldsymbol{B}_{1..N}\}$. Notice that $\boldsymbol{M}_t$ already represents a good choice. We refer to it as ground truth[5], although it is not the optimum.

In the pose estimation experiments, we vary the rotational angle ω of $\boldsymbol{M}_t$ and its enclosed angle ϕ with the optical axis, separately. Here, we use three levels of noise μ_r, denoting the arithmetic mean of the set $\{\|\boldsymbol{r}\|_{1..N}\}$. For each scenario, $N = 15$ points and 100 trials are used. We consider three motors: the motor $\boldsymbol{M}_t$ (TRUE) and the motor estimated by the Gauss-Helmert method (GH). In case of the pose estimation no `SVD` estimate is available, since the G-constraint is quadratically dependent on the parameter vector p. Therefore, an (so far unpublished) geometrical method (GEM), representing the third motor, is used to provide an initial estimate for the Gauss-Helmert method. We assess the quality of a motor $\widehat{\boldsymbol{M}}$ by applying it to the problem setup, i.e. by rotating the model $\{\boldsymbol{a}_{1..N}\}$ to the point set $\{\hat{\boldsymbol{b}}_{1..N}\}$. Next, we measure the distances between the

[5]This only holds in the noiseless case.

transformed model points $\{\hat{\boldsymbol{b}}_{1..N}\}$ and their respective projection rays $\{\boldsymbol{B}_{1..N}\}$. The N distances are then combined by means of the RMS distance. For a set of 100 trials, we compute the Euclidian RMS distance of the 100 respective motors and denote it by μ. Its variance is denoted by σ.

It became obvious, that there is almost no impact on the results if we switch the angle ϕ from 20° to 50° and vice versa. We thus merged the results. Besides, there was no noticeable trend when we varied angle ω, as depicted in table 3.

	Angle ω	10°	40°	70°	100°
Method	TRUE	0.223	0.230	0.229	0.226
	GEM	0.229	0.237	0.235	0.230
	GH	0.215	0.219	0.215	0.213

Table 3. Results of the pose estimation: means μ for $\mu_r = 0.2$ and varying rotation angles ω. The angles were initially used to create the synthetic setups.

	TRUE		GEM		GH	
μ_r	μ	σ	μ	σ	μ	σ
0.200	0.227	0.037	0.233	0.045	0.215	0.040
0.283	0.320	0.051	0.330	0.066	0.304	0.055
0.416	0.470	0.074	0.476	0.095	0.441	0.081

Table 4. Results of the pose estimation for the proposed Gauss-Helmert method (GH), the geometric method (GEM) and the ground truth regarding the noiseless case (TRUE).

Also, the results of the third experiment substantiated the performance of the Gauss-Helmert method. If we compare the results of table 3, it can be seen that the overall fit quality improved reliably regarding the geometric approach (GEM). Moreover, the results are even better than the results of the true motor (TRUE), which is supposed to be superior, in case the number of observations would tend to infinity. Besides, table 4 shows that the standard deviation improves with respect to the input given by the GEM method.

6. Conclusion

In this paper it was shown by synthetic experiments that taking the uncertainty in the data into account, when estimating geometric entities and operators, does improve the results. Geometric Algebra offers a unifying framework where the mostly linear constraints on geometric entities and geometric operators are provided in an implicit manner. Especially, constraints can be expressed succinctly and dimension independently in such a way that linear estimation procedures may be applied.

References

[1] P. Anglès, *Construction de revêtements du groupe conforme d'un espace vectoriel muni d'une ≪métrique≫ de type* (p, q). Ann. Inst. H. Poincaré, **33**(1) (1980), 33–51.

[2] K. S. Arun, T. S. Huang, and S. D. Blostein, *Least-squares fitting of two 3-d point sets.* IEEE Transactions on Pattern Analysis and Machine Intelligence, **9** (5) (1987), 698–700.

[3] W. Förstner, A. Brunn, and S. Heuel, *Statistically testing uncertain geometric relations.* In G. Sommer, N. Krüger, and C. Perwass, editors, *Mustererkennung 2000*, Informatik Aktuell, Springer, Berlin, 2000, pages 17–26.

[4] F.R. Helmert, *Die Ausgleichsrechnung nach der Methode der kleinsten Quadrate.* Teubner, Leipzig, 1872.

[5] S. Heuel, *Uncertain Projective Geometry*, volume 3008 of LNCS. Springer, 2004.

[6] K.-R. Koch, *Parameter Estimation and Hypothesis Testing in Linear Models.* Springer, 1997.

[7] Hongbo Li, David Hestenes, and Alyn Rockwood, *Generalized homogeneous coordinates for computational geometry.* In G. Sommer, editor, Geometric Computing with Clifford Algebras, Springer-Verlag, 2001, pages 27–59.

[8] E.M. Mikhail and F. Ackermann, *Observations and Least Squares.* University Press of America, Lanham, MD20706, USA, 1976.

[9] Faugeras Olivier, *Three-Dimensional Computer Vision.* MIT Press, 1993.

[10] C. Perwass, C. Gebken, and D. Grest, *CLUCalc.* http://www.clucalc.info/, 2004.

[11] C. Perwass, C. Gebken, and G. Sommer, *Estimation of geometric entities and operators from uncertain data.* In 27. Symposium für Mustererkennung, DAGM 2005, Wien, 29.8.-2.9.005, number 3663 in LNCS. Springer-Verlag, Berlin, Heidelberg, 2005.

[12] C. Perwass and D. Hildenbrand, *Aspects of geometric algebra in Euclidean, projective and conformal space.* Technical Report Number 0310, CAU Kiel, Institut für Informatik, September 2003.

[13] C. Perwass and G. Sommer, *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*, Birkhäuser, 2002, pages 341–349.

[14] B. Rosenhahn, *Pose Estimation Revisited.* PhD thesis, Inst. f. Informatik u. Prakt. Math. der Christian-Albrechts-Universität zu Kiel, 2003.

[15] B. Rosenhahn and G. Sommer, *Pose estimation in conformal geometric algebra, part I: The stratification of mathematical spaces.* Journal of Mathematical Imaging and Vision, 22:27–48, 2005.

Christian Gebken
Christian-Albrechts-University, Feldstr. 18, D-24105 Kiel, Germany
e-mail: `c.gebken@gmx.net`

Christian Perwass
Herdweg 94, D-70193 Stuttgart, Germany

Gerald Sommer
Klausdorfer Str. 27a, D-24161 Altenholz, Germany

Received: October, 2005.
Accepted: June, 2006.