Pose Estimation Using Geometric Constraints

Gerald Sommer, Bodo Rosenhahn, and Yiwen Zhang

Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Preußerstrasse 1-9, 24105 Kiel, Germany
{gs,bro,yz}@ks.informatik.uni-kiel.de

Abstract. The paper concerns 2D-3D pose estimation in the algebraic language of kinematics. The pose estimation problem is modeled on the base of several geometric constraint equations. In that way the projective geometric aspect of the topic is implicitly represented and thus, pose estimation is a pure kinematic problem. The authors propose the use of motor algebra to model screw displacements of lines or the use of rotor algebra to model the motion of points. Instead of using matrix based LMS optimization, the development of special extended Kalman filters is proposed. In this paper extended Kalman filters for estimating rotation and translation of several constraints in terms of rotors and motors will be presented. The experiments aim to compare the use of different constraints and different methods of optimal estimating the pose parameters.

1 Introduction

The paper describes the estimation of pose parameters of known rigid objects in the framework of kinematics. The aim is to experimentally verify advantages of extended Kalman filter approaches versus linear least squares optimizations. Pose estimation in the framework of kinematics will be treated as nonlinear optimization with respect to geometric constraint equations expressing the relations between 2D image features and 3D model data.

Pose estimation is a basic visual task. In spite of its importance it has been identified for a long time (see e.g. Grimson [5]), and although there is published an overwhelming number of papers with respect to that topic [8], up to now there is no unique and general solution of the problem. In a general sense, pose estimation can be classified into three categories: 2D-2D, 3D-3D, and 2D-3D. In the first and second category, both the measurement data and the model data are 2D or 3D, respectively. In the third category fall those experiments where measurement data are 2D and model data are 3D. This is the situation we will assume.

An often made assumption is that of rigidity of objects. The wellknown kinematic model of rigid body transformation is a natural one. It consists of rotation and translation. On the other hand, the visual data result from perspective projection, which normally can be modeled using a pinhole camera model.

R. Klette et al. (Eds.): Multi-Image Analysis, LNCS 2032, pp. 153–170, 2001.

In this paper we attend to a pose estimation related to estimations of motion as a problem of kinematics. The problem can be linearly represented in motor algebra [7] or dual quaternion algebra [6]. We are using implicit formulations of geometry as geometric constraints. We will demonstrate that geometric constraints are well conditioned and, thus behave more robust in case of noisy data.

Pose estimation is an optimization problem, formulated in either linear or nonlinear manner, or as either constraint or unconstraint technique. In case of noisy data, which is the standard case in practice, nonlinear optimization techniques are preferred [8]. We will use extended Kalman filters because of their incremental, real-time potential for estimation. In that respect it will be of interest that the estimation error of the fulfillment of the considered geometric constraints keeps a natural distance measure of the considered entities to the actual object frame. Thus, EKF based estimation of geometric constraints permits a progressive scheme of pose estimation.

The paper is organized as follows. In section two we will introduce the motor algebra as representation frame for either geometric entities, geometric constraints, or Euclidean transformations. In section three we introduce the geometric constraints and their changes in an observation scenario. Section four is dedicated to the geometric analysis of these constraints. In section five we will present the EKF approaches for estimating the constraints. In section six we compare the performance of different algorithms for constraint based pose estimation.

2 The Algebraic Frame of Kinematics

In our comparative study we will consider the problem of pose estimation as a kinematic one. In this section we want to sketch the modeling of rigid body motions in the framework of motor algebra, a special degenerate geometric algebra with remarkable advantages.

2.1 The Motor Algebra as Degenerate Geometric Algebra

We introduce the motor algebra as the adequate frame to represent screw transformations in line geometry [7]. This algebra belongs to the family of geometric algebras, a variant of Clifford algebras in which the geometric interpretation of operations is dominantly considered [11].

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

To make it concretely, a motor algebra is the 8D even subalgebra of $\mathcal{G}_{3,0,1}$, derived from $\mathbb{R}^4$, i.e. $n = 4, p = 3, q = 0, r = 1$, with basis vectors γ_k, $k = 1, ..., 4$, and the property $\gamma_1^2 = \gamma_2^2 = \gamma_3^2 = +1$ and $\gamma_4^2 = 0$. Because $\gamma_4^2 = 0$, $\mathcal{G}_{3,0,1}$ is called a degenerate algebra. The motor algebra $\mathcal{G}^+_{3,0,1}$ is of dimension eight and spanned by qualitative different subspaces with the following basis multivectors:

one scalar : 1
six bivectors : $\gamma_2\gamma_3, \gamma_3\gamma_1, \gamma_1\gamma_2, \gamma_4\gamma_1, \gamma_4\gamma_2, \gamma_4\gamma_3$
one pseudoscalar : $\boldsymbol{I} \equiv \gamma_1\gamma_2\gamma_3\gamma_4$.

Because $\gamma_4^2 = 0$, also the unit pseudoscalar squares to zero, i.e. $\boldsymbol{I}^2 = 0$. Remembering that the hypercomplex algebra of quaternions $\mathbb{H}$ represents a 4D linear space with one scalar and three vector components, it can simply be verified that $\mathcal{G}^+_{3,0,1}$ is isomorphic to the algebra of dual quaternions $\widehat{\mathbb{H}}$, [11]. Each dual quaternion $\widehat{q} \in \widehat{\mathbb{H}}$ is related with quaternions $\boldsymbol{q_r}, \boldsymbol{q_d} \in \mathbb{H}$ by $\widehat{q} = \boldsymbol{q_r} + \boldsymbol{I}\boldsymbol{q_d}$. It is obvious from that isomorphism that also quaternions have a representation in geometric algebra, just as complex and real numbers have. Quaternions correspond to the 4D even subalgebra of $\mathcal{G}_{3,0,0}$, derived from $\mathbb{R}^3$. They have the basis $\{1, \gamma_2\gamma_3, \gamma_3\gamma_1, \gamma_1\gamma_2\}$. The advantage of using geometric algebra instead of diverse hypercomplex algebras is the generality of its construction and, derived from that, the existence of algebraic entities with unique interpretation whatever the dimension of the original vector space.

More important is to remark that the bivector basis of motor algebra constitutes the basis for line geometry using Plücker coordinates. Therefore, motor algebra is extraordinary useful to represent line based approaches of kinematics, also in computer vision.

The motor algebra is spanned by bivectors and scalars. Therefore, we will restrict our scope to that case. Let be $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in \langle \mathcal{G}^+_{3,0,1} \rangle_2$ bivectors and $\alpha, \beta \in \langle \mathcal{G}^+_{3,0,1} \rangle_0$ scalars. Then the geometric product of bivectors $\boldsymbol{A}, \boldsymbol{B} \in \langle \mathcal{G}^+_{3,0,1} \rangle_2$, $\boldsymbol{AB}$, splits into $\boldsymbol{AB} = \boldsymbol{A} \cdot \boldsymbol{B} + \boldsymbol{A} \times \boldsymbol{B} + \boldsymbol{A} \wedge \boldsymbol{B}$, where $\mathbf{A} \cdot \mathbf{B}$ is the inner product, which results in a scalar $\boldsymbol{A} \cdot \boldsymbol{B} = \alpha$, $\boldsymbol{A} \wedge \boldsymbol{B}$ is the outer product, which in this case results in a pseudoscalar $\boldsymbol{A} \wedge \boldsymbol{B} = \boldsymbol{I}\beta$, and $\boldsymbol{A} \times \boldsymbol{B}$ is the commutator product, which results in a bivector $\boldsymbol{C}$, $\boldsymbol{A} \times \boldsymbol{B} = \frac{1}{2}(\boldsymbol{AB} - \boldsymbol{BA}) = \boldsymbol{C}$. Changing the sign of the scalar and bivector in the real and the dual parts of the motor leads to the following variants of a motor

$$\boldsymbol{M} = (a_0 + \boldsymbol{a}) + \boldsymbol{I}(b_0 + \boldsymbol{b}) \qquad \widetilde{\boldsymbol{M}} = (a_0 - \boldsymbol{a}) + \boldsymbol{I}(b_0 - \boldsymbol{b})$$
$$\overline{\boldsymbol{M}} = (a_0 + \boldsymbol{a}) - \boldsymbol{I}(b_0 + \boldsymbol{b}) \qquad \overline{\widetilde{\boldsymbol{M}}} = (a_0 - \boldsymbol{a}) - \boldsymbol{I}(b_0 - \boldsymbol{b}).$$

These versions will be used to model the motion of points, lines and planes.

2.2 Rotors, Translators, and Motors

In a general sense, motors are called all the entities existing in motor algebra. Thus, any geometric entity as points, lines, and planes have a motor representation. We will use the term motor in a more restricted sense to call with it a screw transformation, that is an Euclidean transformation embedded in motor

algebra. Its constituents are rotation and translation. In line geometry we represent rotation by a rotation line axis and a rotation angle. The corresponding entity is called a unit rotor, $\boldsymbol{R}$, and reads as follows

$$\boldsymbol{R} = r_0 + r_1\gamma_2\gamma_3 + r_2\gamma_3\gamma_1 + r_3\gamma_1\gamma_2 = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\boldsymbol{n} = \exp\left(\frac{\theta}{2}\boldsymbol{n}\right).$$

Here θ is the rotation angle and $\boldsymbol{n}$ is the unit orientation vector of the rotation axis in bivector representation, spanned by the bivector basis. A unit rotor is in geometric algebra a general entity with a spinor structure, representing rotation in terms of a specified plane. It exists in any dimension and it works for all types of geometric objects, just in contrast to rotation matrices, complex numbers or quaternions. Its very nature is that it is composed by bivectors $\boldsymbol{B}$ and that there is an exponential form $\boldsymbol{R} = \pm\exp\left(\frac{1}{2}\boldsymbol{B}\right)$.

If on the other hand, $\boldsymbol{t} = t_1\gamma_2\gamma_3 + t_2\gamma_3\gamma_1 + t_3\gamma_1\gamma_2$ is a translation vector in bivector representation, it will be represented in motor algebra as the dual part of a motor, called translator $\boldsymbol{T}$ with

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

Thus, a translator is also a special kind of rotor.

Because rotation and translation concatenate multiplicatively in motor algebra, a motor $\boldsymbol{M}$ reads

$$\boldsymbol{M} = \boldsymbol{T}\boldsymbol{R} = \boldsymbol{R} + \boldsymbol{I}\frac{\boldsymbol{t}}{2}\boldsymbol{R} = \boldsymbol{R} + \boldsymbol{I}\boldsymbol{R}'.$$

A motor represents a line transformation as a screw transformation. The line $\boldsymbol{L}$ will be transformed to the line $\boldsymbol{L}'$ by means of a rotation $\boldsymbol{R}_s$ around line $\boldsymbol{L}_s$ by angle θ, followed by a translation $\boldsymbol{t}_s$ parallel to $\boldsymbol{L}_s$. The screw motion equation as motor transformation reads [7], [9]

$$\boldsymbol{L}' = \boldsymbol{T}_s\boldsymbol{R}_s\boldsymbol{L}\widetilde{\boldsymbol{R}}_s\widetilde{\boldsymbol{T}_s} = \boldsymbol{M}\boldsymbol{L}\widetilde{\boldsymbol{M}}.$$

2.3 Motion of Points, Lines, and Planes in Motor Algebra

First, we will introduce the description of the important geometric entities [7].

A point $\boldsymbol{x} \in \mathbb{R}^3$, represented in the bivector basis of $\mathcal{G}^+_{3,0,1}$, i.e. $\boldsymbol{X} \in \mathcal{G}^+_{3,0,1}$, reads $\boldsymbol{X} = 1 + x_1\gamma_4\gamma_1 + x_2\gamma_4\gamma_2 + x_3\gamma_4\gamma_3 = 1 + \boldsymbol{I}\boldsymbol{x}$.

A line $\boldsymbol{L} \in \mathcal{G}^+_{3,0,1}$ is represented by $\boldsymbol{L} = \boldsymbol{n} + \boldsymbol{I}\boldsymbol{m}$ with the line direction $\boldsymbol{n} = n_1\gamma_2\gamma_3 + n_2\gamma_3\gamma_1 + n_3\gamma_1\gamma_2$ and the moment $\boldsymbol{m} = m_1\gamma_2\gamma_3 + m_2\gamma_3\gamma_1 + m_3\gamma_1\gamma_2$.

A plane $\boldsymbol{P} \in \mathcal{G}^+_{3,0,1}$ will be defined by its normal $\boldsymbol{p}$ as bivector and by its Hesse distance to the origin, expressed as the scalar $d = (\boldsymbol{x} \cdot \boldsymbol{p})$, in the following way, $\boldsymbol{P} = \boldsymbol{p} + \boldsymbol{I}d$.

Note that the fact of using line geometry does not prevent to define points and planes, just as in point geometry the other entities also are well defined. In

case of screw motions $\boldsymbol{M} = \boldsymbol{T}_s\boldsymbol{R}_s$ not only line transformations can be modeled, but also point and plane transformations. These are expressed as follows.

$$\boldsymbol{X}' = 1 + \boldsymbol{I}\boldsymbol{x}' = \boldsymbol{M}\boldsymbol{X}\widetilde{\boldsymbol{M}} = \boldsymbol{M}(1 + \boldsymbol{I}\boldsymbol{x})\widetilde{\boldsymbol{M}} = 1 + I(\boldsymbol{R}_s\boldsymbol{x}\widetilde{\boldsymbol{R}_s} + t_s)$$
$$\boldsymbol{L}' = \boldsymbol{n}' + \boldsymbol{I}\boldsymbol{m}' = \boldsymbol{M}\,\boldsymbol{L}\,\widetilde{\boldsymbol{M}} = \boldsymbol{R}_s\boldsymbol{n}\widetilde{\boldsymbol{R}_s} + \boldsymbol{I}(\boldsymbol{R}_s\boldsymbol{n}\widetilde{\boldsymbol{R}_s}' + \boldsymbol{R}_s'\boldsymbol{n}\widetilde{\boldsymbol{R}_s} + \boldsymbol{R}_s\boldsymbol{m}\widetilde{\boldsymbol{R}_s})$$
$$\boldsymbol{P}' = \boldsymbol{p}' + \boldsymbol{I}\boldsymbol{d}' = \boldsymbol{M}\,\boldsymbol{P}\,\widetilde{\boldsymbol{M}} = \boldsymbol{M}(\boldsymbol{p} + \boldsymbol{I}\boldsymbol{d})\widetilde{\boldsymbol{M}} = \boldsymbol{R}_s\boldsymbol{p}\widetilde{\boldsymbol{R}_s} + \boldsymbol{I}((\boldsymbol{R}_s\boldsymbol{p}\widetilde{\boldsymbol{R}_s}) \cdot t_s + d).$$

We will use in this study only point and line transformations because points and lines are the entities of our object models.

3 Geometric Constraints and Pose Estimation

First, we make the following assumptions. The model of an object is given by points and lines in the 3D space. Furthermore we extract line subspaces or points in an image of a calibrated camera and match them with the model of the object. The aim is to find the pose of the object from observations of points and lines in the images at different poses. Figure 1 shows the scenario with respect to observed line subspaces.

We want to estimate the rotation and the translation parameters which lead to the best fit of the model with the extracted line subspaces or points. To estimate the transformations, it is necessary to relate the observed lines in the image to the unknown pose of the object using geometric constraints.

The key idea is that the observed 2D entities together with their corresponding 3D entities are constraint to lie on other, higher order entities which result from the perspective projection. In our considered scenario there are three constraints which are attributed to two classes of constraints:

1. Collinearity: A 3D point has to lie on a line (i.e. a projection ray) in the space
2. Coplanarity: A 3D point or a 3D line has to lie on a plane (i.e. a projection plane).

With the terms projection ray or projection plane, respectively, we mean the image-forming ray which relates a 3D point with the projection center or the infinite set of image-forming rays which relates all 3D points belonging to a 3D line with the projection center, respectively. Thus, by introducing these two entities, we implicitly represent a perspective projection without necessarily formulating it explicitly. Instead, the pose problem is in that framework a purely kinematic problem. A similar approach of avoiding perspective projection equations by using constraint observations of lines has been proposed in [2,3].

In the scenario of figure 1 we describe the following situation: We assume 3D points $\boldsymbol{Y}_i$, and lines $\boldsymbol{S}_i$ of an object or reference model. Further we extract points and lines in an image of a calibrated camera and match them with the model.

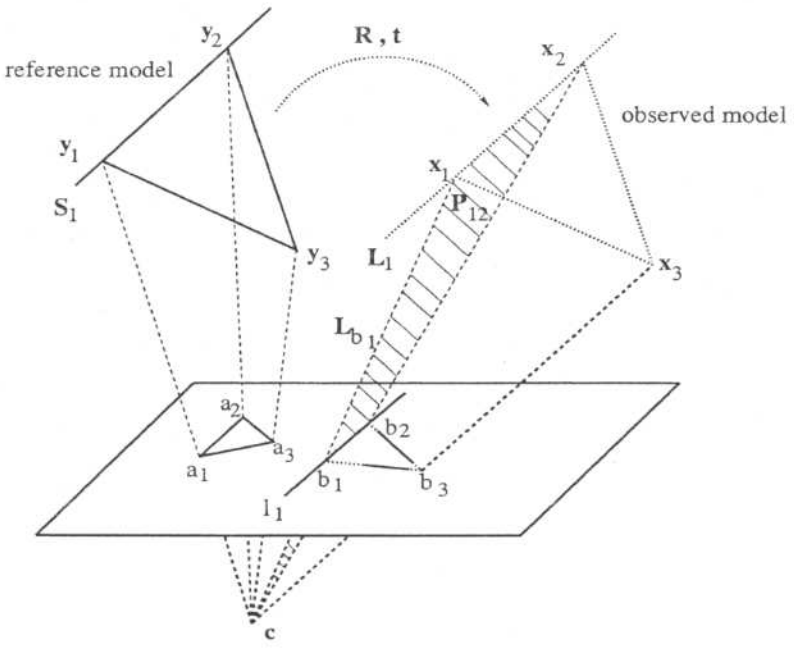

Fig. 1. The scenario. The solid lines at the left hand describe the assumptions: the camera model, the model of the object and the initially extracted lines on the image plane. The dashed lines at the right hand describe the actual pose of the model.

Table 1. The geometric constraints in motor algebra and dual quaternion algebra.

constraint	entities	dual quaternion algebra	motor algebra
point-line	point $\boldsymbol{X} = 1 + \boldsymbol{I}\boldsymbol{x}$ line $\boldsymbol{L} = \boldsymbol{n} + \boldsymbol{I}\boldsymbol{m}$	$\boldsymbol{L}\boldsymbol{X} - \boldsymbol{X}\overline{\boldsymbol{L}} = 0$	$\boldsymbol{X}\boldsymbol{L} - \overline{\boldsymbol{L}}\boldsymbol{X} = 0$
point-plane	point $\boldsymbol{X} = 1 + \boldsymbol{I}\boldsymbol{x}$ plane $\boldsymbol{P} = \boldsymbol{p} + \boldsymbol{I}d$	$\boldsymbol{P}\overline{\boldsymbol{X}} - \boldsymbol{X}\overline{\boldsymbol{P}} = 0$	$\boldsymbol{P}\boldsymbol{X} - \overline{\boldsymbol{X}\boldsymbol{P}} = 0$
line-plane	line $\boldsymbol{L} = \boldsymbol{n} + \boldsymbol{I}\boldsymbol{m}$ plane $\boldsymbol{P} = \boldsymbol{p} + \boldsymbol{I}d$	$\boldsymbol{L}\boldsymbol{P} - \boldsymbol{P}\overline{\boldsymbol{L}} = 0$	$\boldsymbol{L}\boldsymbol{P} + \boldsymbol{P}\overline{\boldsymbol{L}} = 0$

Table 1 gives an overview on the formulations of these constraints in motor algebra, taken from Blaschke [4], who used expressions in dual quaternion algebra. Here we adopt the terms from section 2. The meaning of the constraint equations is immediately clear. They represent the ideal situation, e.g. achieved as the result of the pose estimation procedure with respect to the observation frame. With respect to the previous reference frame these constraints read

$$
\begin{aligned}
(\boldsymbol{M}\boldsymbol{Y}\widetilde{\overline{\boldsymbol{M}}})\boldsymbol{L} - \overline{\boldsymbol{L}}(\boldsymbol{M}\boldsymbol{Y}\widetilde{\overline{\boldsymbol{M}}}) &= 0 \\
\boldsymbol{P}(\boldsymbol{M}\boldsymbol{Y}\widetilde{\overline{\boldsymbol{M}}}) - \overline{(\boldsymbol{M}\boldsymbol{Y}\widetilde{\overline{\boldsymbol{M}}})\boldsymbol{P}} &= 0 \\
(\boldsymbol{M}\boldsymbol{S}\widetilde{\boldsymbol{M}})\boldsymbol{P} + \overline{\boldsymbol{P}(\boldsymbol{M}\boldsymbol{S}\widetilde{\boldsymbol{M}})} &= 0.
\end{aligned}
$$

These compact equations subsume the pose estimation problem at hand: find the best motor $\boldsymbol{M}$ which satisfies the constraint. With respect to the observer frame those entities are variables of the measurement model of the extended Kalman filter on which the motors act.

4 Analysis of Constraints

In this section we will analyze the geometry of the constraints introduced in the last section in motor algebra. We want to show that the relations between different entities are controlled by their orthogonal distance, the Hesse distance. This intuitive result is not only of importance for formulating a mean square minimization method for finding the best motor satisfying the constraints. But in case of noisy data the error of that task can be immediately interpreted as that Hesse distance.

4.1 Point-Line Constraint

Evaluating the constraint of a point $X = 1 + Ix$ collinear to a line $L = n + Im$ leads to

$$0 = XL - \overline{L}X \doteq I(m - n \times x).$$

Since $I \neq 0$, although $I^2 = 0$, the aim is to analyze the bivector $m - n \times x$. Suppose $X \notin L$. Then, nonetheless, there exists a decomposition $x = x_1 + x_2$ with $X_1 = (1 + Ix_1) \in L$ and $X_2 = (1 + Ix_2) \perp L$. Figure 2 shows the scenario. Then we can calculate

$$\|m - n \times x\| = \|m - n \times (x_1 + x_2)\| = \| - n \times x_2\| = \|x_2\|.$$

Thus, satisfying the point-line constraint means to equate the bivectors m and $n \times x$, respectively making the Hesse distance $\|x_2\|$ of the point X to the line L to zero.

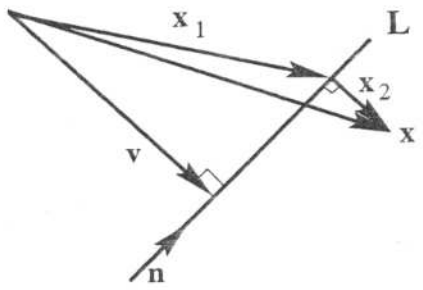

Fig. 2. The line L consists of the direction n and the moment $m = n \times v$. Further, there exists a decomposition $x = x_1 + x_2$ with $X_1 = (1 + Ix_1) \in L$ and $X_2 = (1 + Ix_2) \perp L$, so that $m = n \times v = n \times x_1$.

4.2 Point-Plane Constraint

Evaluating the constraint of a point $X = 1 + Ix$ coplanar to a plane $P = p + Id$ leads to

$$0 = PX - \overline{XP} = I(2d + px + xp) \doteq I(d + p \cdot x).$$

Since $\boldsymbol{I} \neq 0$, although $\boldsymbol{I}^2 = 0$, the aim is to analyze the scalar $d + \boldsymbol{p} \cdot \boldsymbol{x}$. Suppose $\boldsymbol{X} \notin \boldsymbol{P}$. The value d can be interpreted as a sum so that $d = d_{01} + d_{02}$ and $d_{01}\boldsymbol{p}$ is the orthogonal projection of $\boldsymbol{x}$ onto $\boldsymbol{p}$. Figure 3 shows the scenario. Then we

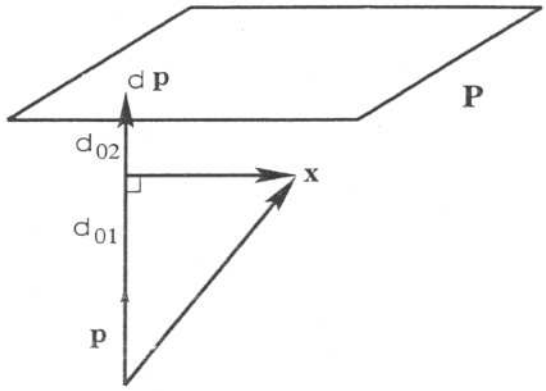

Fig. 3. The value d can be interpreted as a sum $d = d_{01} + d_{02}$ so that $d_{01}\boldsymbol{p}$ corresponds to the orthogonal projection of $\boldsymbol{x}$ onto $\boldsymbol{p}$. That is $d_{01} = -\boldsymbol{p} \cdot \boldsymbol{x}$.

can calculate

$$d + \boldsymbol{p} \cdot \boldsymbol{x} = d_{01} + d_{02} + \boldsymbol{p} \cdot \boldsymbol{x} = d_{01} + \boldsymbol{p} \cdot \boldsymbol{x} + d_{02} = d_{02}.$$

The value of the expression $d + \boldsymbol{p} \cdot \boldsymbol{x}$ corresponds to the Hesse distance of the point $\boldsymbol{X}$ to the plane $\boldsymbol{P}$.

4.3 Line-Plane Constraint

Evaluating the constraint of a line $\boldsymbol{L} = \boldsymbol{n} + \boldsymbol{I}\boldsymbol{m}$ coplanar to a plane $\boldsymbol{P} = \boldsymbol{p} + \boldsymbol{I}d$ leads to

$$0 = \boldsymbol{L}\boldsymbol{P} + \boldsymbol{P}\overline{\boldsymbol{L}} = \boldsymbol{n}\boldsymbol{p} + \boldsymbol{p}\boldsymbol{n} + \boldsymbol{I}(2d\boldsymbol{n} - \boldsymbol{p}\boldsymbol{m} + \boldsymbol{m}\boldsymbol{p}) \doteq \boldsymbol{n} \cdot \boldsymbol{p} + \boldsymbol{I}(d\boldsymbol{n} - \boldsymbol{p} \times \boldsymbol{m})$$

Thus, the constraint can be partitioned in one constraint on the real part of the motor and one constraint on the dual part of the motor. The aim is to analyze the scalar $\boldsymbol{n} \cdot \boldsymbol{p}$ and the bivector $d\boldsymbol{n} - (\boldsymbol{p} \times \boldsymbol{m})$ independently. Suppose $\boldsymbol{L} \notin \boldsymbol{P}$. If $\boldsymbol{n} \not\perp \boldsymbol{p}$ the real part leads to

$$\boldsymbol{n} \cdot \boldsymbol{p} = -\|\boldsymbol{n}\|\|\boldsymbol{p}\| \cos(\alpha) = -\cos(\alpha),$$

where α is the angle between $\boldsymbol{L}$ and $\boldsymbol{P}$, see figure 4. If $\boldsymbol{n} \perp \boldsymbol{p}$, we have $\boldsymbol{n} \cdot \boldsymbol{p} = 0$.

Since the direction of the line is independent of the translation of the rigid body motion, the constraint on the real part can be used to generate equations with the parameters of the rotation as the only unknowns. The constraint on the dual part can then be used to determine the unknown translation. In other words, since the motor to be estimated, $\boldsymbol{M} = \boldsymbol{R} + \boldsymbol{I}\boldsymbol{R}\boldsymbol{T} = \boldsymbol{R} + \boldsymbol{I}\boldsymbol{R}'$, is determined in its real part only by rotation, the real part of the constraint allows to estimate

the rotor $\boldsymbol{R}$, while the dual part of the constraint allows to estimate the rotor $\boldsymbol{R}'$. So it is possible to sequentially separate equations on the unknown rotation from equations on the unknown translation without the limitations, known from the embedding of the problem in Euclidean space [6]. This is very useful, since the two smaller equation systems are easier to solve than one larger equation system.

To analyze the dual part of the constraint, we interpret the moment $\boldsymbol{m}$ of the line representation $\boldsymbol{L} = \boldsymbol{n} + \boldsymbol{I}\boldsymbol{m}$ as $\boldsymbol{m} = \boldsymbol{n} \times \boldsymbol{s}$ and choose a vector $\boldsymbol{s}$ with $\boldsymbol{S} = (1 + \boldsymbol{I}\boldsymbol{s}) \in \boldsymbol{L}$ and $\boldsymbol{s} \perp \boldsymbol{n}$. By expressing the inner product as the anti-commutator product, it can be shown ([1]) that $-(\boldsymbol{p} \times \boldsymbol{m}) = (\boldsymbol{s} \cdot \boldsymbol{p})\boldsymbol{n} - (\boldsymbol{n} \cdot \boldsymbol{p})\boldsymbol{s}$. Now we can evaluate

$$d\boldsymbol{n} - (\boldsymbol{p} \times \boldsymbol{m}) = d\boldsymbol{n} - (\boldsymbol{n} \cdot \boldsymbol{p})\boldsymbol{s} + (\boldsymbol{s} \cdot \boldsymbol{p})\boldsymbol{n}.$$

Figure 4 shows the scenario. Further, we can find a vector $\boldsymbol{s_1}$ with $\boldsymbol{s} \parallel \boldsymbol{s_1}$, so

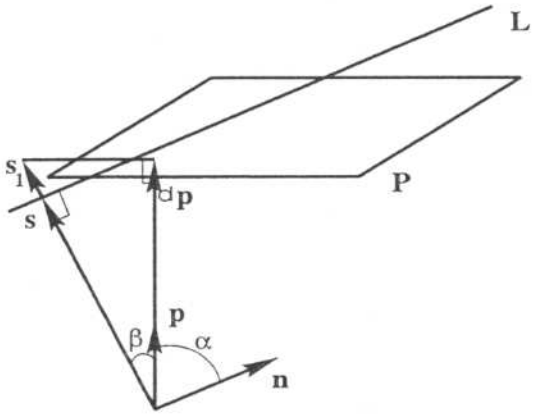

Fig. 4. The plane $\boldsymbol{P}$ consists of its normal $\boldsymbol{p}$ and the Hesse distance d. Furthermore we choose $\boldsymbol{S} = (1 + \boldsymbol{I}\boldsymbol{s}) \in \boldsymbol{L}$ with $\boldsymbol{s} \perp \boldsymbol{n}$.

that

$$0 = d - (\|\boldsymbol{s}\| + \|\boldsymbol{s_1}\|)\cos(\beta).$$

The vector $\boldsymbol{s_1}$ might also be antiparallel to $\boldsymbol{s}$. This leads to a change of the sign, but does not affect the constraint itself. Now we can evaluate

$$d\boldsymbol{n} - (\boldsymbol{n} \cdot \boldsymbol{p})\boldsymbol{s} + (\boldsymbol{s} \cdot \boldsymbol{p})\boldsymbol{n} = d\boldsymbol{n} - \|\boldsymbol{s}\|\cos(\beta)\boldsymbol{n} + \cos(\alpha)\boldsymbol{s} = \|\boldsymbol{s_1}\|\cos(\beta)\boldsymbol{n} + \cos(\alpha)\boldsymbol{s}.$$

The error of the dual part consists of the vector $\boldsymbol{s}$ scaled by the angle α and the direction $\boldsymbol{n}$ scaled by the norm of $\boldsymbol{s_1}$ and the angle β.
If $\boldsymbol{n} \perp \boldsymbol{p}$, we will find

$$\|d\boldsymbol{n} - (\boldsymbol{p} \times \boldsymbol{m})\| = \|d\boldsymbol{n} + (\boldsymbol{s} \cdot \boldsymbol{p})\boldsymbol{n} - (\boldsymbol{n} \cdot \boldsymbol{p})\boldsymbol{s}\| = |(d + \boldsymbol{s} \cdot \boldsymbol{p})|$$

This means, in agreement to the point-plane constraint, that $(d + \boldsymbol{s} \cdot \boldsymbol{p})$ describes the Hesse distance of the line to the plane.

This analysis shows that the considered constraints are not only qualitative constraints, but also quantitative ones. This is very important, since we want to measure the extend of fulfillment of these constraints in the case of noisy data.

5 The Extended Kalman Filter for Pose Estimation

In this section we want to present the design of EKFs for estimating the pose based on three constraints. Because an EKF is defined in the frame of linear vector algebra, it will be necessary to map the estimation task from any chosen algebraic embedding to linear vector algebra (see e.g. [9]).

5.1 EKF Pose Estimation Based on Point-Line Constraint

In case of point based measurements of the object at different poses, an algebraic embedding of the problem in the 4D linear space of the algebra of rotors $\mathcal{G}^+_{3,0,0}$, which is isomorphic to that one of quaternions $\mathbb{H}$, will be sufficient [7,9]. Thus, rotation will be represented by a unit rotor $\boldsymbol{R}$ and translation will be a bivector $\boldsymbol{t}$. A point $\boldsymbol{y_1}$ transformed to $\boldsymbol{x_1}$ reads $\boldsymbol{x_1} = \boldsymbol{R}\boldsymbol{y_1}\tilde{\boldsymbol{R}} + \boldsymbol{t}$. We denote the four components of the rotor as

$$\boldsymbol{R} = r_0 + r_1\sigma_2\sigma_3 + r_2\sigma_3\sigma_1 + r_3\sigma_1\sigma_2.$$

To convert a rotor $\boldsymbol{R}$ into a rotation matrix $\boldsymbol{\mathcal{R}}$, simple conversion rules are at hand:

$$\boldsymbol{\mathcal{R}} = \begin{pmatrix} r_0^2 + r_1^2 - r_2^2 - r_3^2 & 2(r_1r_2 + r_0r_3) & 2(r_1r_3 - r_0r_2) \\ 2(r_1r_2 - r_0r_3) & r_0^2 - r_1^2 + r_2^2 - r_3^2 & 2(r_2r_3 + r_0r_1) \\ 2(r_1r_3 + r_0r_2) & 2(r_2r_3 - r_0r_1) & r_0^2 - r_1^2 - r_2^2 + r_3^2 \end{pmatrix}.$$

In vector algebra, the above point transformation model can be described as

$$\mathbf{x_1} = \boldsymbol{\mathcal{R}}\mathbf{y_1} + \mathbf{t}.$$

The projection ray $\mathbf{L}_{b_1}$ in the point-line equation is represented by Plücker coordinates $(\mathbf{n_1}, \mathbf{m_1})$, where $\mathbf{n_1}$ is its unit direction and $\mathbf{m_1}$ its moment. The point-line constraint equation in vector algebra of $\mathbb{R}^3$ reads

$$\mathbf{f_1} = \mathbf{m_1} - \mathbf{n_1} \times \mathbf{x_1} = \mathbf{m_1} - \mathbf{n_1} \times (\boldsymbol{\mathcal{R}}\mathbf{y_1} + \mathbf{t}) = \mathbf{0}.$$

Let the state vector $\mathbf{s}$ for the EKF be a 7D vector, composed in terms of the rotor coefficients for rotation and translation,

$$\mathbf{s} = (\mathbf{R}^T, \mathbf{t}^T)^T = (r_0, r_1, r_2, r_3, t_1, t_2, t_3)^T.$$

The rotation coefficients must satisfy the unit condition

$$\mathbf{f_2} = \mathbf{R}^T\mathbf{R} - 1 = r_0^2 + r_1^2 + r_2^2 + r_3^2 - 1 = 0.$$

The noise free measurement vector $\mathbf{a_i}$ is given by the actual line parameters $\mathbf{n_i}$ and $\mathbf{m_i}$, and the actual 3D point measurements $\mathbf{y_i}$,

$$\mathbf{a_i} = (\mathbf{n_i}^T, \mathbf{m_i}^T, \mathbf{y_i}^T)^T = (n_{i1}, n_{i2}, n_{i3}, m_{i1}, m_{i2}, m_{i3}, y_{i1}, y_{i2}, y_{i3})^T.$$

For a sequence of measurements $\mathbf{a_i}$ and states $\mathbf{s_i}$, the constraint equations

$$\mathbf{f_i}(\mathbf{a_i}, \mathbf{s_i}) = \begin{pmatrix} \mathbf{f_{1i}} \\ \mathbf{f_{2i}} \end{pmatrix} = \begin{pmatrix} \mathbf{m_i} - \mathbf{n_i} \times (\mathcal{R}_\mathbf{i}\mathbf{y_i} + \mathbf{t_i}) \\ \mathbf{R_i}^T\mathbf{R_i} - 1 \end{pmatrix} = \mathbf{0}$$

relate measurements and states in a nonlinear manner. The system model in this static case should be $\mathbf{s_{i+1}} = \mathbf{s_i} + \boldsymbol{\zeta_i}$, where $\boldsymbol{\zeta_i}$ is a vector random sequence with known statistics, $E[\boldsymbol{\zeta_i}] = 0$, $E[\boldsymbol{\zeta_i}^T\boldsymbol{\zeta_k}] = \mathcal{Q}_\mathbf{i}\delta_\mathbf{ik}$, where $\delta_\mathbf{ik}$ is the Kronecker delta and the matrix $\mathcal{Q}_\mathbf{i}$ is assumed to be nonnegative definite. We assume that the measurement system is disturbed by additive white noise, i.e., the real observed measurement $\mathbf{a_i'}$ is expressed as $\mathbf{a_i'} = \mathbf{a_i} + \boldsymbol{\eta_i}$.

The vector $\boldsymbol{\eta_i}$ is an additive, random sequence with known statistics, $E[\boldsymbol{\eta_i}] = 0$, $E[\boldsymbol{\eta_i}^T\boldsymbol{\eta_k}] = \mathcal{W}_\mathbf{i}\delta_\mathbf{ik}$, where the matrix $\mathcal{W}_\mathbf{i}$ is assumed to be nonnegative definite.

Since the observation equation is nonlinear (that means, the relationship between the measurement $\mathbf{a_i'}$ and state $\mathbf{s_i}$ is nonlinear), we expand $\mathbf{f_i}(\mathbf{a_i}, \mathbf{s_i})$ into a Taylor series about the $(\mathbf{a_i'}, \hat{\mathbf{s}}_{\mathbf{i/i-1}})$, where $\mathbf{a_i'}$ is the real measurement and $\hat{\mathbf{s}}_{\mathbf{i/i-1}}$ is the predicted state at situation i. By ignoring the second order terms, we get the linearized measurement equation

$$\mathbf{z_i} = \mathcal{H}_\mathbf{i}\mathbf{s_i} + \boldsymbol{\xi_i},$$

where

$$\begin{aligned} \mathbf{z_i} &= \mathbf{f_i}(\mathbf{a_i'}, \hat{\mathbf{s}}_{\mathbf{i/i-1}}) - \frac{\partial \mathbf{f_i}(\mathbf{a_i'}, \hat{\mathbf{s}}_{\mathbf{i/i-1}})}{\partial \mathbf{s_i}} \hat{\mathbf{s}}_{\mathbf{i/i-1}} \\ &= \begin{pmatrix} \mathbf{m_i'} - \mathbf{n_i'} \times (\hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{y_i'} + \hat{\mathbf{t}}_{\mathbf{i/i-1}}) \\ \hat{\mathbf{R}}^T_{\mathbf{i/i-1}}\hat{\mathbf{R}}_{\mathbf{i/i-1}} - 1 \end{pmatrix} + \mathcal{H}_\mathbf{i}\hat{\mathbf{s}}_{\mathbf{i/i-1}}. \end{aligned}$$

The measurement matrix $\mathcal{H}_\mathbf{i}$ of the linearized measurement $\mathbf{z_i}$ reads

$$\mathcal{H}_\mathbf{i} = -\frac{\partial \mathbf{f_i}(\mathbf{a_i'}, \hat{\mathbf{s}}_{\mathbf{i/i-1}})}{\partial \mathbf{s_i}} = \begin{pmatrix} \mathcal{C}_{\mathbf{n_i'}}\mathcal{D}_{\hat{\mathcal{R}}\mathbf{y'}} & \mathcal{C}_{\mathbf{n_i'}} \\ \mathcal{D}_\mathbf{R} & \mathbf{0}_{1\times 3} \end{pmatrix},$$

where

$$\mathcal{D}_\mathbf{R} = \frac{\partial(\hat{\mathbf{R}}^T_{\mathbf{i/i-1}}\hat{\mathbf{R}}_{\mathbf{i/i-1}} - 1)}{\partial \mathbf{R_i}} = \begin{pmatrix} -2\hat{r}_{(i/i-1)0} & -2\hat{r}_{(i/i-1)1} & -2\hat{r}_{(i/i-1)2} & -2\hat{r}_{(i/i-1)3} \end{pmatrix},$$

$$\mathcal{D}_{\hat{\mathcal{R}}\mathbf{y'}} = \frac{\partial(\hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{y_i'})}{\partial \mathbf{R_i}} = \begin{pmatrix} d_1 & d_2 & d_3 & d_4 \\ d_4 & -d_3 & d_2 & -d_1 \\ -d_3 & -d_4 & d_1 & d_2 \end{pmatrix},$$

$$
\begin{aligned}
d_1 &= 2(\hat{r}_{(i/i-1)0}y'_{i1} + \hat{r}_{(i/i-1)3}y'_{i2} - \hat{r}_{(i/i-1)2}y'_{i3}),\\
d_2 &= 2(\hat{r}_{(i/i-1)1}y'_{i1} + \hat{r}_{(i/i-1)2}y'_{i2} + \hat{r}_{(i/i-1)3}y'_{i3}),\\
d_3 &= 2(-\hat{r}_{(i/i-1)2}y'_{i1} + \hat{r}_{(i/i-1)1}y'_{i2} - \hat{r}_{(i/i-1)0}y'_{i3}),\\
d_4 &= 2(-\hat{r}_{(i/i-1)3}y'_{i1} + \hat{r}_{(i/i-1)0}y'_{i2} + \hat{r}_{(i/i-1)1}y'_{i3}).
\end{aligned}
$$

The 3×3 matrix $\mathcal{C}_{\mathbf{n}'_\mathbf{i}}$ is the skew-symmetric matrix of $\mathbf{n}'_\mathbf{i}$. For any vector $\mathbf{y}$, we have $\mathcal{C}_{\mathbf{n}'_\mathbf{i}}\mathbf{y} = \mathbf{n}'_\mathbf{i} \times \mathbf{y}$ with

$$
\mathcal{C}_{\mathbf{n}'_\mathbf{i}} = \begin{pmatrix} 0 & -n'_{i3} & n'_{i2} \\ n'_{i3} & 0 & -n'_{i1} \\ -n'_{i2} & n'_{i1} & 0 \end{pmatrix}.
$$

The measurement noise is given by

$$
\begin{aligned}
\boldsymbol{\xi}_\mathbf{i} &= -\frac{\partial \mathbf{f_i}(\mathbf{a}'_\mathbf{i}, \hat{\mathbf{s}}_{\mathbf{i/i-1}})}{\partial \mathbf{a_i}}(\mathbf{a_i} - \mathbf{a}'_\mathbf{i}) = \frac{\partial \mathbf{f_i}(\mathbf{a}'_\mathbf{i}, \hat{\mathbf{s}}_{\mathbf{i/i-1}})}{\partial \mathbf{a_i}}\boldsymbol{\eta}_\mathbf{i}\\
&= \begin{pmatrix} \mathcal{C}_{\hat{\mathbf{x}}_{\mathbf{i/i-1}}} & \mathbf{I}_{3\times 3} & -\mathcal{C}_{\mathbf{n}'_\mathbf{i}}\hat{\mathcal{R}}_{\mathbf{i/i-1}} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{pmatrix}_{4\times 9} \boldsymbol{\eta}_\mathbf{i},
\end{aligned}
$$

where $\mathbf{I}_{3\times 3}$ is a unit matrix and $\mathcal{C}_{\hat{\mathbf{x}}_{\mathbf{i/i-1}}}$ is the skew-symmetric matrix of $\hat{\mathbf{x}}_{\mathbf{i/i-1}}$ with

$$
\hat{\mathbf{x}}_{\mathbf{i/i-1}} = \hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{y}'_\mathbf{i} + \hat{\mathbf{t}}_{\mathbf{i/i-1}}.
$$

The expectation and the covariance of the new measurement noise $\boldsymbol{\xi}_\mathbf{i}$ are easily derived from that of $\mathbf{a}'_\mathbf{i}$ as

$$
E[\boldsymbol{\xi}_\mathbf{i}] = \mathbf{0} \text{ and } E[\boldsymbol{\xi}_\mathbf{i}^T\boldsymbol{\xi}_\mathbf{i}] = \boldsymbol{\mathcal{V}}_\mathbf{i} = (\frac{\partial \mathbf{f_i}(\mathbf{a}'_\mathbf{i}, \hat{\mathbf{s}}_{\mathbf{i/i-1}})}{\partial \mathbf{a_i}})\boldsymbol{\mathcal{W}}_\mathbf{i}(\frac{\partial \mathbf{f_i}(\mathbf{a}'_\mathbf{i}, \hat{\mathbf{s}}_{\mathbf{i/i-1}})}{\partial \mathbf{a_i}})^T.
$$

The EKF motion estimation algorithms based on point-plane and line-plane constraints can be derived in a similar way.

5.2 EKF Pose Estimation Based on Point-Plane Constraint

The projection plane $\mathbf{P_{12}}$ in the point-plane constraint equation is represented by $(d_1, \mathbf{p_1})$, where d_1 is its Hesse distance and $\mathbf{p_1}$ its unit direction. The point-plane constraint equation in vector algebra of $\mathbb{R}^3$ reads

$$
d_1 - \mathbf{p_1}^T(\mathcal{R}\mathbf{x_1} + \mathbf{t}) = 0.
$$

With the measurement vector $\mathbf{a_i} = (d_i, \mathbf{p_i}^T, \mathbf{y_i}^T)^T$ and the same state vector $\mathbf{s}$ as above, the measurement $\mathbf{z_i}$ of linearized measurement equation reads

$$
\mathbf{z_i} = \begin{pmatrix} d'_\mathbf{i} - \mathbf{p}'^T_\mathbf{i}(\hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{y}'_\mathbf{i} + \hat{\mathbf{t}}_{\mathbf{i/i-1}}) \\ \hat{\mathbf{R}}^T_{\mathbf{i/i-1}}\hat{\mathbf{R}}_{\mathbf{i/i-1}} - 1 \end{pmatrix} + \mathcal{H}_\mathbf{i}\hat{\mathbf{s}}_{\mathbf{i/i-1}}.
$$

The measurement matrix $\mathcal{H}_i$ of the linearized measurement $\mathbf{z_i}$ now reads

$$\mathcal{H}_i = \begin{pmatrix} \mathbf{p_i}^T \mathcal{D}_{\hat{\mathcal{R}}\mathbf{y}'} & \mathbf{p_i}^T \\ \mathcal{D}_{\mathbf{R}} & \mathbf{0}_{1\times 3} \end{pmatrix}.$$

The measurement noise is given by

$$\xi_i = \begin{pmatrix} 1 & -(\hat{\mathcal{R}}_{i/i-1}\mathbf{y}'_i + \hat{\mathbf{t}}_{i/i-1})^T & -(\mathbf{p}_i'^T \hat{\mathcal{R}}_{i/i-1}) \\ 0 & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{pmatrix}_{2\times 7} \eta_i.$$

5.3 EKF Pose Estimation Based on Line-Plane Constraint

Using the line-plane constraint, the reference model entity in $\mathcal{G}^+_{3,0,1}$ [9,7] is the Plücker line $\boldsymbol{S_1} = \boldsymbol{n_1} + \boldsymbol{I m_1}$. This line transformed by a motor $\boldsymbol{M} = \boldsymbol{R} + \boldsymbol{IR'}$ reads

$$\boldsymbol{L_1} = \boldsymbol{M S_1 \widetilde{M}} = \boldsymbol{R n_1 \widetilde{R}} + \boldsymbol{I}(\boldsymbol{R n_1 \widetilde{R'}} + \boldsymbol{R' n_1 \widetilde{R}} + \boldsymbol{R m_1 \widetilde{R}}) = \boldsymbol{u_1} + \boldsymbol{I v_1}.$$

We denote the 8 components of the motor as

$$\boldsymbol{M} = r_0 + r_1\gamma_2\gamma_3 + r_2\gamma_3\gamma_1 + r_3\gamma_1\gamma_2 + \boldsymbol{I}(r'_0 + r'_1\gamma_2\gamma_3 + r'_2\gamma_3\gamma_1 + r'_3\gamma_1\gamma_2).$$

The line motion equation can be equivalently expressed by vector form,

$$\mathbf{u_1} = \mathcal{R}\mathbf{n_1} \text{ and } \mathbf{v_1} = \mathcal{A}\mathbf{n_1} + \mathcal{R}\mathbf{m_1},$$

with

$$\mathcal{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix},$$

$$\begin{aligned}
a_{11} &= 2(r'_0r_0 + r'_1r_1 - r'_2r_2 - r'_3r_3), & a_{12} &= 2(r'_3r_0 + r'_2r_1 + r'_1r_2 + r'_0r_3), \\
a_{13} &= 2(-r'_2r_0 + r'_3r_1 - r'_0r_2 + r'_1r_3), & a_{21} &= 2(-r'_3r_0 + r'_2r_1 + r'_1r_2 - r'_0r_3), \\
a_{22} &= 2(r'_0r_0 - r'_1r_1 + r'_2r_2 - r'_3r_3), & a_{23} &= 2(r'_1r_0 + r'_0r_1 + r'_3r_2 + r'_2r_3), \\
a_{31} &= 2(r'_2r_0 + r'_3r_1 + r'_0r_2 + r'_1r_3), & a_{32} &= 2(-r'_1r_0 - r'_0r_1 + r'_3r_2 + r'_2r_3), \\
a_{33} &= 2(r'_0r_0 - r'_1r_1 - r'_2r_2 + r'_3r_3).
\end{aligned}$$

The line-plane constraint equation in vector algebra of $\mathbb{R}^3$ reads

$$\begin{pmatrix} \mathbf{f_1} \\ \mathbf{f_2} \end{pmatrix} = \begin{pmatrix} \mathbf{p_1}^T\mathbf{u_1} \\ d_1\mathbf{u_1} + \mathbf{v_1} \times \mathbf{p_1} \end{pmatrix} = \begin{pmatrix} \mathbf{p_1}^T(\mathcal{R}\mathbf{n_1}) \\ d_1\mathcal{R}\mathbf{n_1} + (\mathcal{A}\mathbf{n_1} + \mathcal{R}\mathbf{m_1}) \times \mathbf{p_1} \end{pmatrix} = \mathbf{0}.$$

We use the 8 components of the motor as the state vector for the EKF,

$$\mathbf{s} = (r_0, r_1, r_2, r_3, r'_0, r'_1, r'_2, r'_3)^T$$

and these 8 components must satisfy both the unit and orthogonal conditions:

$$\begin{aligned}
f_3 &= r_0^2 + r_1^2 + r_2^2 + r_3^2 - 1 = 0, \\
f_4 &= r_0r'_0 + r_1r'_1 + r_2r'_2 + r_3r'_3 = 0.
\end{aligned}$$

The 10D noise free measurement vector $\mathbf{a_i}$ is given by the true plane parameters d_i and $\mathbf{p_i}$, and the true 6D line parameters $(\mathbf{n_i}, \mathbf{m_i})$,

$$\mathbf{a_i} = (d_i, \mathbf{p_i}^T, \mathbf{n_i}^T, \mathbf{m_i}^T)^T = (d_i, p_{i1}, p_{i2}, p_{i3}, n_{i1}, n_{i2}, n_{i3}, m_{i1}, m_{i2}, m_{i3})^T.$$

The new measurement in linearized equation reads

$$\mathbf{z_i} = \begin{pmatrix} \mathbf{p_i'}^T(\hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{n_i'}) \\ d_i'\hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{n_i'} + (\hat{\mathcal{A}}_{\mathbf{i/i-1}}\mathbf{n_i'} + \hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{m_i'}) \times \mathbf{p_i'} \\ \hat{\mathbf{R}}^T_{\mathbf{i/i-1}}\hat{\mathbf{R}}_{\mathbf{i/i-1}} - 1 \\ \hat{\mathbf{R}}^T_{\mathbf{i/i-1}}\hat{\mathbf{R}}'_{\mathbf{i/i-1}} \end{pmatrix} + \mathcal{H}_\mathbf{i}\hat{\mathbf{s}}_{\mathbf{i/i-1}}.$$

The measurement matrix $\mathcal{H}_\mathbf{i}$ of the linearized measurement $\mathbf{z_i}$ reads

$$\mathcal{H}_\mathbf{i} = \begin{pmatrix} -\mathbf{p_i'}^T\mathcal{D}_{\hat{\mathcal{R}}\mathbf{n'}} & \mathbf{0}_{1\times 4} \\ -d_\mathbf{i}'\mathcal{D}_{\hat{\mathcal{R}}\mathbf{n'}} + \mathcal{C}_{\mathbf{p_i'}}(\mathcal{D}_{\hat{\mathcal{A}}\mathbf{n'}} + \mathcal{D}_{\hat{\mathcal{R}}\mathbf{m'}}) & \mathcal{C}_{\mathbf{p_i'}}\mathcal{D}_{\hat{\mathcal{R}}\mathbf{n'}} \\ \mathcal{D}_\mathbf{R} & \mathbf{0}_{1\times 4} \\ \mathcal{D}_{R'} & \frac{1}{2}\mathcal{D}_\mathbf{R} \end{pmatrix},$$

where $\mathcal{D}_{\hat{\mathcal{R}}\mathbf{n'}} = \frac{\partial(\hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{n_i'})}{\partial \mathbf{R}_i}$, $\mathcal{D}_{\hat{\mathcal{R}}\mathbf{m'}} = \frac{\partial(\hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{m_i'})}{\partial \mathbf{R_i}}$, $\mathcal{D}_{\hat{\mathcal{A}}\mathbf{n'}} = \frac{\partial(\hat{\mathcal{A}}_{\mathbf{i/i-1}}\mathbf{n_i'})}{\partial \mathbf{R_i}}$, $\mathcal{D}_{R'} = \frac{\partial(\hat{\mathbf{R}}^T_{\mathbf{i/i-1}}\hat{\mathbf{R}}'_{\mathbf{i/i-1}})}{\partial \mathbf{R_i}}$ and $\frac{1}{2}\mathcal{D}_\mathbf{R} = \frac{\partial(\hat{\mathbf{R}}^T_{\mathbf{i/i-1}}\hat{\mathbf{R}}'_{\mathbf{i/i-1}})}{\partial \mathbf{R_i'}}$.

The 3×3 matrix $\mathcal{C}_{\mathbf{p_i'}}$ is the skew-symmetric matrix of $\mathbf{p_i'}$. The measurement noise is given by

$$\xi_\mathbf{i} = \begin{pmatrix} 0 & \mathbf{n_i'}^T\hat{\mathcal{R}}_{\mathbf{i/i-1}} & \mathbf{p_i'}^T\hat{\mathcal{R}}_{\mathbf{i/i-1}} & \mathbf{0}_{1\times 3} \\ \hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{n_i'} & \mathcal{C}_{\hat{\mathbf{v}}_\mathbf{i}} & d_\mathbf{i}'\hat{\mathcal{R}}_{\mathbf{i/i-1}} - \mathcal{C}_{\mathbf{p_i'}}\hat{\mathcal{A}}_{\mathbf{i/i-1}} & -\mathcal{C}_{\mathbf{p_i'}}\hat{\mathcal{R}}_{\mathbf{i/i-1}} \\ \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 3} \end{pmatrix} \eta_\mathbf{i}$$

where $\mathcal{C}_{\hat{\mathbf{v}}_\mathbf{i}}$ is skew-symmetric matrix of $\hat{\mathbf{v}}_\mathbf{i}$, and $\hat{\mathbf{v}}_\mathbf{i}$ is defined as

$$\hat{\mathbf{v}}_\mathbf{i} = \hat{\mathcal{A}}_{\mathbf{i/i-1}}\mathbf{n_i'} + \hat{\mathcal{R}}_{\mathbf{i/i-1}}\mathbf{m_i'}.$$

Having linearized the measurement models, the EKF implementation is straightforward and standard. Further implementation details will not be repeated here [10,9,12]. In next section, we will denote the EKF as **RtEKF**, if the state explicitly uses the rotor components of rotation $\mathbf{R}$ and of translation $\mathbf{t}$, or MEKF, if the components of motor $\mathbf{M}$ are used.

6 Experiments

In this section we present some experiments with real images. The aim of the experiments is to study the performance of the algorithms for pose estimation based on geometric constraints. We expect that both the special constraints and the algorithmic approach of using them may influence the results.

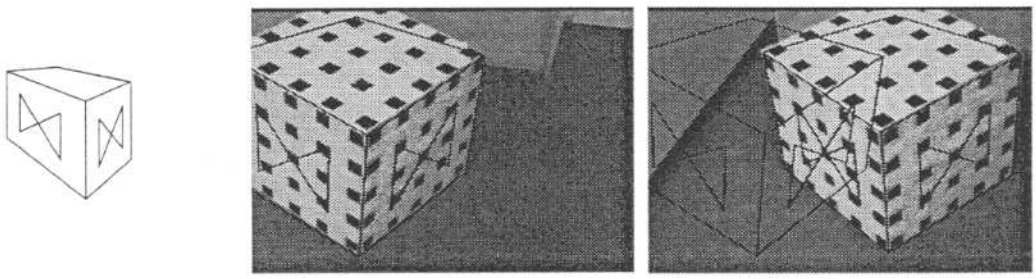

Fig. 5. The scenario of the experiment: The calibration of an object model is performed and the 3D object model is projected on the image. Then the camera moved and corresponding line segments are extracted.

Table 2. The experiment 1 results in different qualities of derived motion parameters, depending on the used constraints and algorithms to evaluate their validity.

no.	$\mathcal{R}$ — t	Constraint	Experiment 1		Error
1	RtEKF — RtEKF	XL-XL	$\mathcal{R} = \begin{pmatrix} 0.987 & 0.089 & -0.138 \\ -0.117 & 0.969 & -0.218 \\ 0.115 & 0.231 & 0.966 \end{pmatrix}$	$\mathbf{t} = \begin{pmatrix} -58.21 \\ -217.26 \\ 160.60 \end{pmatrix}$	5.2
2	SVD — MAT	XL-XL	$\mathcal{R} = \begin{pmatrix} 0.976 & 0.107 & -0.191 \\ -0.156 & 0.952 & -0.264 \\ 0.154 & 0.287 & 0.945 \end{pmatrix}$	$\mathbf{t} = \begin{pmatrix} -60.12 \\ -212.16 \\ 106.60 \end{pmatrix}$	6.7
3	RtEKF — RtEKF	XP-XP	$\mathcal{R} = \begin{pmatrix} 0.987 & 0.092 & -0.133 \\ -0.118 & 0.973 & -0.200 \\ 0.111 & 0.213 & 0.970 \end{pmatrix}$	$\mathbf{t} = \begin{pmatrix} -52.67 \\ -217.00 \\ 139.00 \end{pmatrix}$	5.5
4	RtEKF — MAT	XP-XP	$\mathcal{R} = \begin{pmatrix} 0.986 & 0.115 & -0.118 \\ -0.141 & 0.958 & -0.247 \\ 0.085 & 0.260 & 0.962 \end{pmatrix}$	$\mathbf{t} = \begin{pmatrix} -71.44 \\ -219.34 \\ 124.71 \end{pmatrix}$	3.7
5	SVD — MAT	XP-XP	$\mathcal{R} = \begin{pmatrix} 0.979 & 0.101 & -0.177 \\ -0.144 & 0.957 & -0.251 \\ 0.143 & 0.271 & 0.952 \end{pmatrix}$	$\mathbf{t} = \begin{pmatrix} -65.55 \\ -221.18 \\ 105.87 \end{pmatrix}$	5.3
6	SVD — MAT	LP-XP	$\mathcal{R} = \begin{pmatrix} 0.976 & 0.109 & -0.187 \\ -0.158 & 0.950 & -0.266 \\ 0.149 & 0.289 & 0.945 \end{pmatrix}$	$\mathbf{t} = \begin{pmatrix} -66.57 \\ -216.18 \\ 100.53 \end{pmatrix}$	7.1
7	MEKF — MEKF	LP-LP	$\mathcal{R} = \begin{pmatrix} 0.985 & 0.106 & -0.134 \\ -0.133 & 0.969 & -0.208 \\ 0.107 & 0.229 & 0.969 \end{pmatrix}$	$\mathbf{t} = \begin{pmatrix} -50.10 \\ -212.60 \\ 142.20 \end{pmatrix}$	2.9
8	MEKF — MAT	LP-LP	$\mathcal{R} = \begin{pmatrix} 0.985 & 0.106 & -0.134 \\ -0.133 & 0.968 & -0.213 \\ 0.108 & 0.228 & 0.968 \end{pmatrix}$	$\mathbf{t} = \begin{pmatrix} -67.78 \\ -227.73 \\ 123.90 \end{pmatrix}$	2.7
9	SVD — MAT	LP-LP	$\mathcal{R} = \begin{pmatrix} 0.976 & 0.109 & -0.187 \\ -0.158 & 0.950 & -0.266 \\ 0.149 & 0.289 & 0.945 \end{pmatrix}$	$\mathbf{t} = \begin{pmatrix} -80.58 \\ -225.59 \\ 93.93 \end{pmatrix}$	6.9

In our experimental scenario we positioned a camera two meters in front of a calibration cube. We focused the camera on the calibration cube and took an image. Then we moved the camera, focused the camera again on the cube and took another image. The edge size of the calibration cube is 46 cm and the image size is 384 × 288 pixel. Furthermore, we defined on the calibration cube a 3D object model. Figure 5 shows the scenario. In the left images the calibration is performed and the 3D object model is projected on the image. Then the camera is moved and corresponding line segments are extracted. In these experiments we actually selected certain points by hand and from these the depicted line segments are derived and, by knowing the camera calibration by the cube of the first image, the actual projection ray and projection plane parameters are computed.

For the first experiment we show in table 2 the results of different algorithms for pose estimation. In the second column of table 2 EKF denotes the use of the EKFs derived in section 5, MAT denotes matrix algebra, SVD denotes the singular value decomposition of a matrix. In the third column the used constraints,

point-line (XL), point-plane (XP) and line-plane (LP) are indicated. The fourth column shows the results of the estimated rotation matrix $\mathcal{R}$ and the translation vector $\mathbf{t}$, respectively. The translation vectors are shown in mm. The fifth column shows the error of the equation system. Since the error of the equation system describes the Hesse distance of the entities, the value of the error is an approximation of the squared average distance of the entities. It is easy to see, that the results obtained with the different approaches are all very close to each other, though the implementation leads to totally different calculations and algorithms. Furthermore the EKF's perform more stable than the matrix solution approaches.

The visualization of some errors is done in figure 6. We calculated the motion of the object and projected the transformed object in the image plane. The extracted line segments are overlayed in addition. Figure 6 shows the results of nos. 5, 3, 7 and 8 of table 2, respectively.

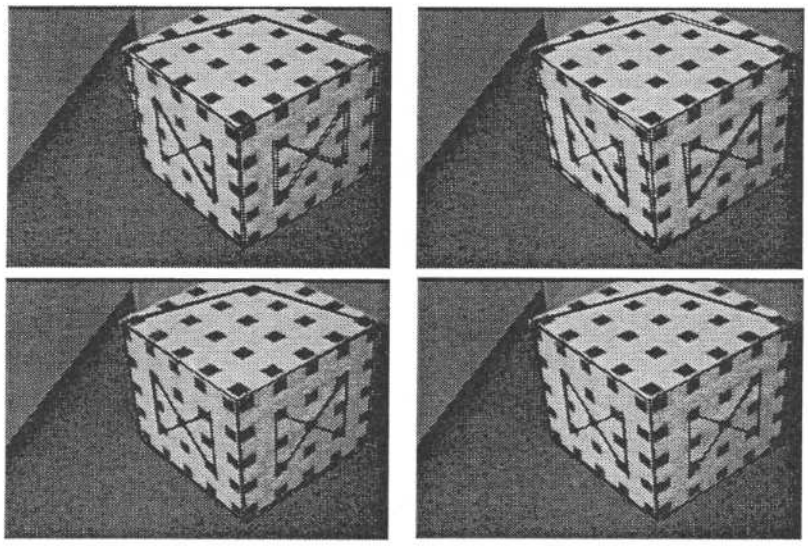

Fig. 6. Visualization of some errors. The results of nos. 5, 3 7 and 8 of table 2 are visualized respectively.

In a second experiment we compare the noise sensitivity of the Kalman filters and of the matrix solution approaches for pose estimation. The experiment is organized as follows. We took the point correspondences of the first experiment and estimated both $\mathcal{R}$ and $\mathbf{t}$. Then we added a Gaussian noise error on the extracted image points. The error varied from 0 to 16 pixels in 0.25 steps and we estimated $\mathcal{R}'$ and $\mathbf{t}'$ for each step. Then we calculated the error between $\mathcal{R}'$ and $\mathcal{R}$ and between $\mathbf{t}'$ and $\mathbf{t}$. The results are shown in figure 7. Since $\mathcal{R}$ and $\mathcal{R}'$ are rotation matrices, the absolute value of the error differs in the range $0 \leq \epsilon_{\mathcal{R}} \leq 1$. The error of the translation vector is evaluated in mm. So the error of the translation vector differs by using the matrix solution approach at around $0 \leq \epsilon_{\mathbf{t}} \leq 10$ cm, while using the Kalman filter the corresponding range is $0 \leq \epsilon_{\mathbf{t}} \leq 6$ cm. The matrix based solutions look all very similar. Compared with the

EKF results they are very sensitive to noise and the variances between the noise steps are very high. The results are in agreement with the well known behavior of error propagation in case of matrix based rotation estimation. The EKF based solutions perform all very stable and the behavior of the different constraints are also very similar. This is a consequence of the estimators themselves and of the fact that the concatenation of rotors is more robust than that of rotation matrices. It is obvious, that the results of these experiments are affected by the method to obtain the entities in the image. In this experiment we selected certain points directly by hand and derived from these the line subspaces. So the quality of the line subspaces is directly connected to the quality of the point extraction. For comparison purposes between the algorithms this is necessary and reasonable. But for real applications, since the extraction of lines is more stable than that of points, the XP or LP algorithms should be preferred.

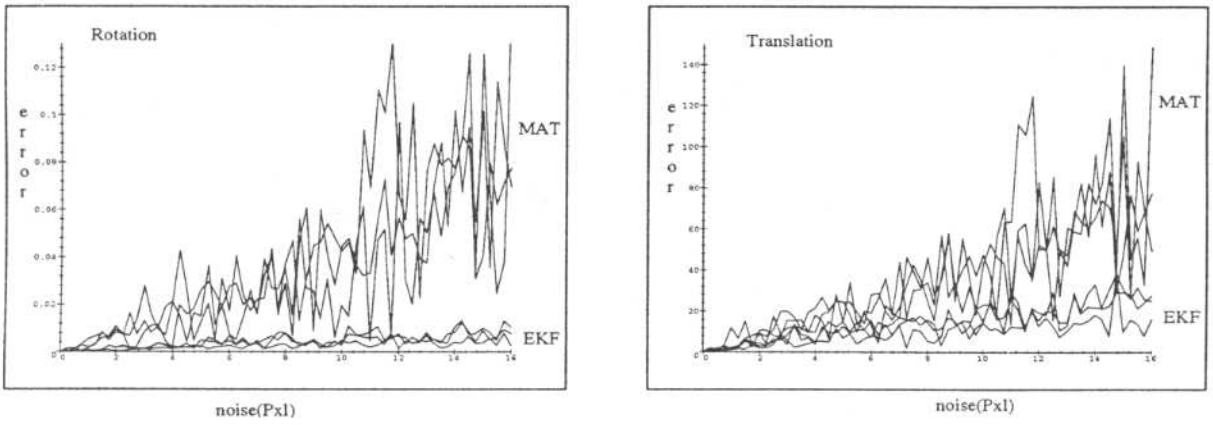

Fig. 7. Performance comparison of different methods in case of noisy data. With increasing noise the EKF performs with more accurate and more stable estimates than the matrix based methods.

7 Conclusions

In this paper we describe a framework for 2D-3D pose estimation. The aim of the paper is to compare several pose modeling approaches and estimation methods with respect to their performance. The main contribution of the paper is to formulate 2D-3D pose determination in the language of kinematics as a problem of estimating rotation and translation from geometric constraint equations. There are three such constraints which relate the model frame to an observation frame. The model data are either points or lines. The observation frame is constituted by lines or planes. Any deviations from the constraint correspond the Hesse distance of the involved geometric entities. From this starting point as a useful algebraic frame for handling line motion, the motor algebra has been introduced. The estimation procedure is realized as extended Kalman filters (EKF).

The paper presents EKFs for estimating rotation and translation for each constraint model in different algebraic frames. The experiments show advantages of that representation and of the EKF approaches in comparison to normal matrix based LMS algorithms, all applied within the context of constraint based pose estimation.

References

[1] C. Perwass and J. Lasenby. A novel axiomatic derivation of geometric algebra. *Technical Report CUED/F - INFENG/TR.347, Cambridge University Engineering Department*, 1999.

[2] Shevlin F. Analysis of orientation problems using Plücker lines. *International Conference on Pattern Recognition, Brisbane*, 1: 685–689, 1998.

[3] Horaud R., Phong T.Q. and P.D. Tao. Object pose from 2-d to 3-d point and line correspondences. *International Journal of Computer Vision*, 15: 225–243, 1995.

[4] Blaschke W. Mathematische Monographien 4, Kinematik und Quaternionen. *Deutscher Verlag der Wissenschaften*, Berlin, 1960.

[5] Grimson W. E. L. Object Recognition by Computer. *The MIT Press, Cambridge, MA*, 1990.

[6] Daniilidis K. Hand-eye calibration using dual quaternions. *Int. Journ. Robotics Res*, 18: 286–298, 1999.

[7] Bayro-Corrochano E. The geometry and algebra of kinematics. *In Sommer G., editor, Geometric Computing with Clifford Algebra. Springer Verlag*, to be published, 2000.

[8] Carceroni R. L. and C. M. Brown. Numerical methods for model-based pose recovery. *Techn. Rept. 659, Comp. Sci. Dept., The Univ. of Rochester, Rochester, N. Y.*, August 1998.

[9] Zhang Y., Sommer G., and E. Bayro-Corrochano. The motor extended Kalman filter for dynamic rigid motion estimation from line observations. *In G. Sommer, editor, Geometric Computing with Clifford Algebra. Springer Verlag*, to be published, 2000.

[10] Sommer G., Rosenhahn B. and Zhang Y. Pose estimation using geometric constraints. *Techn. Rept. 2003, Institut für Informatik und Praktische Mathematik Christian-Albrechts-Universität zu Kiel*, 2000.

[11] Hestenes D., Li H. and A. Rockwood. New algebraic tools for classical geometry. *In Sommer G., editor, Geometric Computing with Clifford Algebra. Springer Verlag*, to be published, 2000.

[12] Zhang, Z. and O. Faugeras. 3D Dynamic Scene Analysis. *Springer Verlag*, 1992.