Pose Estimation in the Language of Kinematics

Bodo Rosenhahn, Yiwen Zhang, Gerald Sommer

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

Abstract. The paper concerns 2D-3D pose estimation in the algebraic language of kinematics. The pose estimation problem is modelled on the base of several geometric constraint equations. In that way the projective geometric aspect of the topic is only implicitly represented and thus, pose estimation is a pure kinematic problem. The dynamic measurements of these constraints are either points or lines. The authors propose the use of motor algebra to introduce constraint equations, which keep a natural distance measurement, the Hesse distance. The motor algebra is a degenerate geometric algebra in which line transformations are linear ones. 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. 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 [9], up to now there is no unique and general solution of the problem. Pose estimation means to relate several coordinate frames of measurement data and model data by finding out the transformations between, which can subsume rotation and translation. Since we assume our measurement data as 2D and model data as 3D, we are concerned with a 2D-3D pose estimation problem. Camera self-localization and navigation are typical examples of such types of problems. The coupling of projective and Euclidean transformations, both with nonlinear representations in Euclidean space, is the main reason for the difficulties to solve the pose problem. In this paper we attend to a pose estimation related to estimations of line motion as a problem of kinematics. The problem can be linearly represented in motor algebra [8] or dual quaternion algebra [7]. Instead of using invariances as an explicit formulation of geometry as often has been done in projective geometry, we are using implicit formulations of geometry as geometric constraints. We will demonstrate that geometric constraints are well conditioned, in contrast to invariances.

The paper is organized as follows. In section two we will introduce the motor algebra as representation frame for either geometric entities, geometric constraints, and 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 show some results for constraint based pose estimation with real images.

2 The motor algebra in the frame of kinematics

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 concretly, a motor algebra is the 8D even algebra $\mathcal{G}^+_{3,0,1}$, 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 $\hat{\mathbb{H}}$ [11]. 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 $\boldsymbol{C}$, $\boldsymbol{A}\times\boldsymbol{B}=\frac{1}{2}(\boldsymbol{AB}-\boldsymbol{BA})=\boldsymbol{C}$. In a general sense, motors are called all the entities existing in motor algebra. They are constituted by bivectors and scalars. Thus, any geometric entity as points, lines, and planes have a motor representation. 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

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

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

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 (and dilation in case of non-unit motors). 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, spanned by the bivector basis.

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 $\mathbf{M}$ reads

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

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 a line $\boldsymbol{L}_s$

by angle θ, followed by a translation $\boldsymbol{t}_s$ parallel to $\boldsymbol{L}_s$. Then the screw motion equation as motor transformation reads

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

For more detailed introductions see [8] and [10]. Now we will introduce the description of the most important geometric entities [8].

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, $\boldsymbol{P} = \boldsymbol{p} + \boldsymbol{I}d$.

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

$$\boldsymbol{X}' = \boldsymbol{M}\boldsymbol{X}\widetilde{\widetilde{\boldsymbol{M}}} \qquad \boldsymbol{L}' = \boldsymbol{M}\,\boldsymbol{L}\,\widetilde{\boldsymbol{M}} \qquad \boldsymbol{P}' = \boldsymbol{M}\,\boldsymbol{P}\,\widetilde{\widetilde{\boldsymbol{M}}}$$

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. The method of obtaining the line subspaces is out of scope of this paper. Contemporary we simply got line segments by marking certain image points by hand. To estimate the pose, 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 (projection ray) in the space
2. Coplanarity: A 3D point or line has to lie on a plane (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. The most important consequence of implicitly representing projective geometry is that the pose problem is in that framework a pure kinematic problem. A similar approach of avoiding perspective projection equations by using constraint observations of lines has been proposed in [2].

To be more detailed, in the scenario of figure 1 we describe the following situation: We assume 3D points $\boldsymbol{A}'_i$ and lines $\boldsymbol{L'}_{Ai}$ of an object model. Further we extract line subspaces $\boldsymbol{l}_{ai}$ in an image of a calibrated camera and match them with the model.

Three constraints can be depicted:

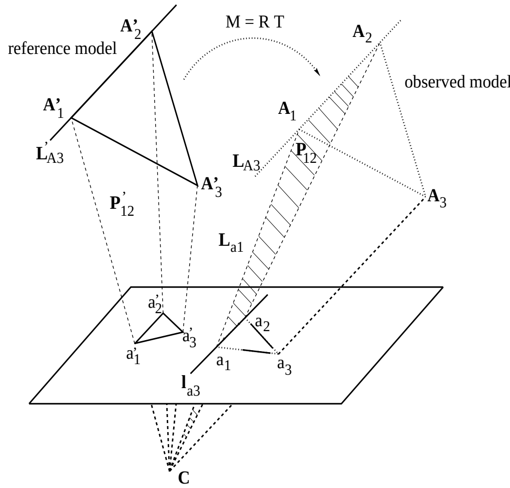

Fig. 1. The scenario. The solid lines at the left hand describe the assumptions: the camera model, the model of the object and the initially extracted lines on the image plane. The dashed lines at the right hand describe the actual pose of the model, which leads to the best fit of the object with the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

1. A transformed point, e.g. $\boldsymbol{A_1}$, of the model point $\boldsymbol{A_1'}$ must lie on the projection ray $\boldsymbol{L_{a1}}$, given by $\boldsymbol{C}$ and the corresponding image point $\boldsymbol{a_1}$.
2. A transformed point, e.g. $\boldsymbol{A_1}$, of the model point $\boldsymbol{A_1'}$ must lie on the projection plane $\boldsymbol{P_{12}}$, given by $\boldsymbol{C}$ and the corresponding image line $\boldsymbol{l}_{a3}$.
3. A transformed line, e.g. $\boldsymbol{L}_{A3}$, of the model line $\boldsymbol{L'}_{A3}$ must lie on the projection plane $\boldsymbol{P_{12}}$, given by $\boldsymbol{C}$ and the the corresponding image line $\boldsymbol{l}_{a3}$.

constraint	entities	dual quaternion algebra	motor algebra
point-line	point $\boldsymbol{X} = 1 + \boldsymbol{Ix}$ line $\boldsymbol{L} = \boldsymbol{n} + \boldsymbol{Im}$	$\boldsymbol{LX} - \boldsymbol{X}\overline{\boldsymbol{L}} = 0$	$\boldsymbol{XL} - \overline{\boldsymbol{L}}\boldsymbol{X} = 0$
point-plane	point $\boldsymbol{X} = 1 + \boldsymbol{Ix}$ plane $\boldsymbol{P} = \boldsymbol{p} + \boldsymbol{I}d$	$\boldsymbol{P}\overline{\boldsymbol{X}} - \boldsymbol{X}\overline{\boldsymbol{P}} = 0$	$\boldsymbol{PX} - \overline{\boldsymbol{XP}} = 0$
line-plane	line $\boldsymbol{L} = \boldsymbol{n} + \boldsymbol{Im}$ plane $\boldsymbol{P} = \boldsymbol{p} + \boldsymbol{I}d$	$\boldsymbol{LP} - \boldsymbol{P}\overline{\boldsymbol{L}} = 0$	$\boldsymbol{LP} + \boldsymbol{P}\overline{\boldsymbol{L}} = 0$

Table 1. The geometric constraints expressed in motor algebra and dual quatenion algebra, respectively.

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. In section 4 we will proceed to analyse them in detail. 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, indicated by primes, these constraints read

$$(\boldsymbol{M}\boldsymbol{X}'\widetilde{\overline{\boldsymbol{M}}})\boldsymbol{L} - \overline{\boldsymbol{L}}(\boldsymbol{M}\boldsymbol{X}'\widetilde{\overline{\boldsymbol{M}}}) = 0$$

$$\boldsymbol{P}(\boldsymbol{M}\boldsymbol{X}'\widetilde{\boldsymbol{M}}) - \overline{(\boldsymbol{M}\boldsymbol{X}'\widetilde{\boldsymbol{M}})}\overline{\boldsymbol{P}} = 0$$
$$(\boldsymbol{M}\boldsymbol{L}'\widetilde{\boldsymbol{M}})\boldsymbol{P} + \boldsymbol{P}\overline{(\boldsymbol{M}\boldsymbol{L}'\widetilde{\boldsymbol{M}})} = 0.$$

These compact equations subsume the pose estimation problem at hand: find the best motor $\boldsymbol{M}$ which satisfies the constraint. We will get a convex optimization problem. Any error measure $|\epsilon| > 0$ of the optimization process as actual deviation from the constraint equation can be interpreted as a distance measure of misalignment with respect to the ideal situation of table 1. That means e.g. that the constraint for a point on a line is almost fulfilled for a point near the line. This will be made clear in the following section 4.

4 Analysis of the constraints

In this section we will analyse the geometry of the constraints introduced in the last section. We want to show that the relations between different entities are controlled by their orthogonal distance, the Hesse distance.

4.1 Point-line constraint

Evaluating the constraint of a point $\boldsymbol{X} = 1 + \boldsymbol{I}\boldsymbol{x}$ collinear to a line $\boldsymbol{L} = \boldsymbol{n} + \boldsymbol{I}\boldsymbol{m}$ leads to

$$\begin{aligned} 0 &= \boldsymbol{X}\boldsymbol{L} - \overline{\boldsymbol{L}}\boldsymbol{X} = (1 + \boldsymbol{I}\boldsymbol{x})(\boldsymbol{n} + \boldsymbol{I}\boldsymbol{m}) - (\boldsymbol{n} - \boldsymbol{I}\boldsymbol{m})(1 + \boldsymbol{I}\boldsymbol{x}) \\ &= \boldsymbol{n} + \boldsymbol{I}\boldsymbol{m} + \boldsymbol{I}\boldsymbol{x}\boldsymbol{n} - \boldsymbol{n} + \boldsymbol{I}\boldsymbol{m} - \boldsymbol{I}\boldsymbol{n}\boldsymbol{x} = \boldsymbol{I}(2\boldsymbol{m} + \boldsymbol{x}\boldsymbol{n} - \boldsymbol{n}\boldsymbol{x}) \\ &= 2\boldsymbol{I}(\boldsymbol{m} - \boldsymbol{n} \times \boldsymbol{x}) \\ \Leftrightarrow 0 &= \boldsymbol{I}(\boldsymbol{m} - \boldsymbol{n} \times \boldsymbol{x}). \end{aligned}$$

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

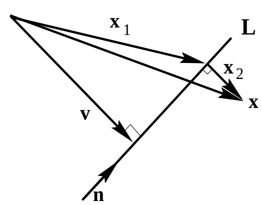

Fig. 2. The line $\boldsymbol{L}$ consists of the direction $\boldsymbol{n}$ and the moment $\boldsymbol{m} = \boldsymbol{n} \times \boldsymbol{v}$. Further, there exists a decomposition $\boldsymbol{x} = \boldsymbol{x_1} + \boldsymbol{x_2}$ with $\boldsymbol{X_1} = (1+\boldsymbol{I}\boldsymbol{x_1}) \in \boldsymbol{L}$ and $\boldsymbol{X_2} = (1+\boldsymbol{I}\boldsymbol{x_2}) \perp \boldsymbol{L}$, so that $\boldsymbol{m} = \boldsymbol{n} \times \boldsymbol{v} = \boldsymbol{n} \times \boldsymbol{x}_1$.

Then we can calculate

$$\begin{aligned} \|\boldsymbol{m} - \boldsymbol{n} \times \boldsymbol{x}\| &= \|\boldsymbol{m} - \boldsymbol{n} \times (\boldsymbol{x_1} + \boldsymbol{x_2})\| = \|\boldsymbol{m} - \boldsymbol{n} \times \boldsymbol{x_1} - \boldsymbol{n} \times \boldsymbol{x_2}\| \\ &= \| - \boldsymbol{n} \times \boldsymbol{x_2}\| = \|\boldsymbol{x_2}\|. \end{aligned}$$

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

4.2 Point-plane constraint

Evaluating the constraint of a point $\boldsymbol{X} = 1 + \boldsymbol{I}\boldsymbol{x}$ coplanar to a plane $\boldsymbol{P} = \boldsymbol{p} + \boldsymbol{I}d$ leads to

$$\begin{aligned} 0 &= \boldsymbol{P}\boldsymbol{X} - \overline{\boldsymbol{X}\boldsymbol{P}} = (\boldsymbol{p} + \boldsymbol{I}d)(1 + \boldsymbol{I}\boldsymbol{x}) - (1 - \boldsymbol{I}\boldsymbol{x})(\boldsymbol{p} - \boldsymbol{I}d) \\ &= \boldsymbol{p} + \boldsymbol{I}\boldsymbol{p}\boldsymbol{x} + \boldsymbol{I}d - \boldsymbol{p} + \boldsymbol{I}d + \boldsymbol{I}\boldsymbol{x}\boldsymbol{p} = \boldsymbol{I}(2d + \boldsymbol{p}\boldsymbol{x} + \boldsymbol{x}\boldsymbol{p}) \\ \Leftrightarrow 0 &= \boldsymbol{I}(d + \boldsymbol{p} \cdot \boldsymbol{x}). \end{aligned}$$

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

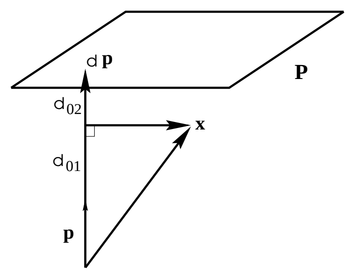

Fig. 3. The value d can be interpreted as a sum $d = d_{01} + d_{02}$ so that $d_{01}\boldsymbol{p}$ corresponds to the orthogonal projection of $\boldsymbol{x}$ onto $\boldsymbol{p}$.

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

$$\begin{aligned} 0 &= \boldsymbol{L}\boldsymbol{P} + \boldsymbol{P}\overline{\boldsymbol{L}} = (\boldsymbol{n} + \boldsymbol{I}\boldsymbol{m})(\boldsymbol{p} + \boldsymbol{I}d) + (\boldsymbol{p} + \boldsymbol{I}d)(\boldsymbol{n} - \boldsymbol{I}\boldsymbol{m}) \\ &= \boldsymbol{n}\boldsymbol{p} + \boldsymbol{I}\boldsymbol{m}\boldsymbol{p} + \boldsymbol{I}\boldsymbol{n}d + \boldsymbol{p}\boldsymbol{n} + \boldsymbol{I}\boldsymbol{n}d - \boldsymbol{I}\boldsymbol{p}\boldsymbol{m} \\ &= \boldsymbol{n}\boldsymbol{p} + \boldsymbol{p}\boldsymbol{n} + \boldsymbol{I}(2d\boldsymbol{n} - \boldsymbol{p}\boldsymbol{m} + \boldsymbol{m}\boldsymbol{p}) \\ \Leftrightarrow 0 &= \boldsymbol{n} \cdot \boldsymbol{p} + \boldsymbol{I}(d\boldsymbol{n} - \boldsymbol{p} \times \boldsymbol{m}) \end{aligned}$$

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 [7]. This is very useful, since the two smaller equation systems are easier to solve than one larger equation system. To analyse 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 $\boldsymbol{s_1} \parallel \boldsymbol{s}$ with

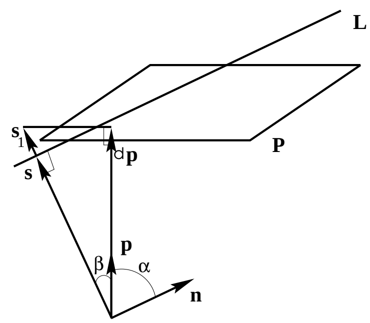

Fig. 4. The plane $\boldsymbol{P}$ consists of its normal $\boldsymbol{p}$ and the Hesse distance d. Furthermore we choose $\boldsymbol{S} = (1 + \boldsymbol{I}\boldsymbol{s}) \in \boldsymbol{L}$ with $\boldsymbol{s} \perp \boldsymbol{n}$. The angle of $\boldsymbol{n}$ and $\boldsymbol{p}$ is α and the angle of $\boldsymbol{s}$ and $\boldsymbol{p}$ is β. We choose the vector $\boldsymbol{s_1}$ with $\boldsymbol{s} \parallel \boldsymbol{s_1}$ so that $d\boldsymbol{p}$ is the orthogonal projection of $(\boldsymbol{s} + \boldsymbol{s_1})$ onto $\boldsymbol{p}$.

$0 = d - (\|\boldsymbol{s}\| + \|\boldsymbol{s_1}\|)\cos(\beta)$. 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 $\boldsymbol{n} \perp \boldsymbol{p}$, then $\boldsymbol{p} \parallel \boldsymbol{s}$ and thus, we will find

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

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 Experiments

In this section we present some experiments with real images. We expect that both the special constraint and the algorithmic approach of using it may influence the results. In our experimental scenario we took a B21 mobile robot equipped with a stereo camera head and positioned it two meters in front of a

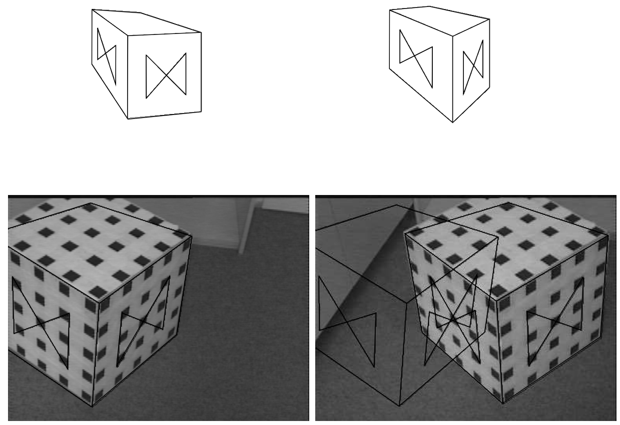

Fig. 5. The scenario of the experiment: In the top row two perspectives of the 3D object model are shown. In the second row (left) the calibration is performed and the 3D object model is projected on the image. Then the camera moved and corresponding line segments are extracted.

calibration cube. We focused one camera on the calibration cube and took an image. Then we moved the robot, 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 first row two perspective views of the 3D object model are shown. In the left image of the second row the calibration is performed and the 3D object model is projected onto the image. Then the camera is moved and corresponding line segments are extracted. To visualize the movement, we also projected the 3D object model on its original position. The aim is to find the pose of the model and so the motion of the camera. In this experiment 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. In table 2 we show the results of different algorithms for pose estimation. In the second column of table 2 EKF denotes the use of an extended Kalman filter. The design of the extended Kalman filters is described in [6]. MAT denotes matrix algebra, SVD denotes the singular value decomposition of a matrix to ensure a rotation matrix as a result. 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 $\boldsymbol{\mathcal{R}}$ and the translation vector $\boldsymbol{t}$, respectively. Since the translation vectors are in mm, the results differ at around 2-3 cm. 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 close to each other, though the implementation leads to different algorithms. Furthermore the EKF's perform more stable than the matrix solution approaches.

no.	$\mathcal{R}$ — t	Constraint	Experiment 1		Error
1	**Rt**EKF — **Rt**EKF	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	**Rt**EKF — **Rt**EKF	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	**Rt**EKF — 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

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

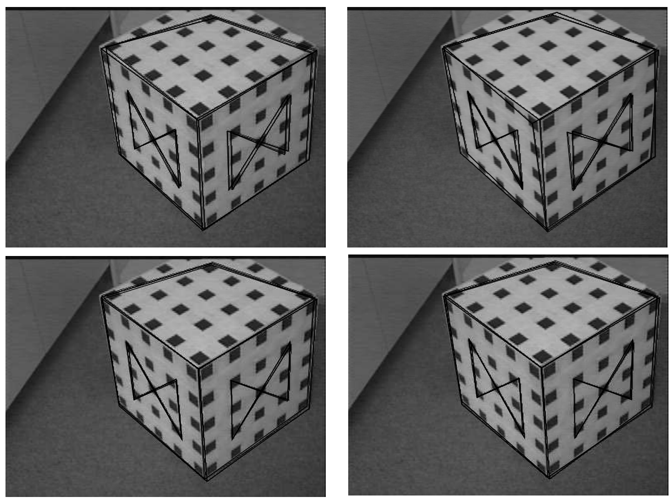

Fig. 6. Visualization of some errors. We calculate the motion of the object and project the transformed object in the image planes. The extracted line segments are also shown. In the first and second row, the results of nos. 5, 3 and nos. 7, 8 of table 2 are visualised respectively.

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 in the first row, left the results of nos. 5, 3 and nos. 7, 8 of table 2 respectively. The results of no. 7 and 8 are very good, compared with the results of the other algorithms.

These results are in agrement with the well known behavior of error propagation in case of matrix based rotation estimation. The EKF performs more stable. This is a consequence of the estimator themselves and of the fact that in our approach rotation is represented as rotors. The concatenation of rotors is more robust than that of rotation matrices.

6 Conclusions

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. This is an eight-dimensional linear space with the property of representing rigid movements in a linear manner. The use of the motor algebra allows to subsume the pose estimation problem by compact equations, since the entities, the transformation of the entities and the constraints for collinearity or coplanarity of entities can be described very economically. Furthermore the introduced constraints contain a natural distance measurement, the Hesse distance. This is the reason why the geometric constraints are well conditioned (in contrast to invariances) and, thus behave more robust in case of noisy data.

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*, 1960.
5. Grimson W. E. L. Object Recognition by Computer. *The MIT Press, Cambridge, MA*, 1990.
6. Sommer G., Rosenhahn B. and Zhang Y. Pose Estimation Using Geometric Constraints *Technical Report 2003, Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität zu Kiel*
7. Daniilidis K. Hand-eye calibration using dual quaternions. *Int. Journ. Robotics Res*, 18: 286–298, 1999.
8. Bayro-Corrochano E. The geometry and algebra of kinematics. *In Sommer G., editor, Geometric Computing with Clifford Algebra. Springer Verlag*, to be published, 2000.
9. 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.
10. 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.
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.

This article was processed using the LaTeX macro package with LLNCS style