



The Motor Extended Kalman Filter: A Geometric Approach for Rigid Motion Estimation

EDUARDO BAYRO-CORROCHANO*

Centro de Investigación en Matemáticas, A.C., Apartado Postal 402, 36000-Guanajuato, Gto, Mexico

edb@fractal.cimat.mx

YIWEN ZHANG

Computer Science Institute, Christian Albrechts University, Preußersstraße 1-9, 24105, Kiel, Germany

Abstract. In this paper the motor algebra for linearizing the 3D Euclidean motion of lines is used as the oretical basis for the development of a novel extended Kalman filter called the motor extended Kalman filter (MEKF). Due to its nature the MEKF can be used as online approach as opposed to batch SVD methods. The MEKF does not encounter singularities when computing the Kalman gain and it can estimate simultaneously the translation and rotation transformations. Many algorithms in the literature compute the translation and rotation transformations separately. The experimental part demonstrates that the motor extended Kalman filter is an useful approach for estimation of dynamic motion problems. We compare the MEKF with an analytical method using simulated data. We present also an application using real images of a visual guided robot manipulator; the aim of this experiment is to demonstrate how we can use the online MEKF algorithm. After the system has been calibrated, the MEKF estimates accurately the relative position of the end-effector and a 3D reference line. We believe that future vision systems being reliably calibrated will certainly make great use of the MEKF algorithm.

Keywords: computer vision, Clifford algebra, geometric algebra, kinematics, dynamics, rotors; motors, screws, Kalman filter techniques, extended Kalman filter, visual robotics

1. Introduction

The registration of the motion of a moving object or the computation of the motion between measurement frames in front of an observer is fundamental for various tasks in visual robotics, such as camera calibration, hand-eye calibration, tracking, object collision avoidance and surveillance. The most basic of the 3D geometric primitives of the visual space for motion computation are points (corners). These local features are sensitive to noise and quantization errors that jeopardize the motion estimation. Alternatively, the use of lines (edges) or global features such as planes or surfaces makes the motion estimation process more robust, however, the computational cost is increased especially in the case of planes or surfaces. Since an artificial ob-

server relies on image sequences, only the projected motion of points, lines, conics, curves or surfaces is distinguished. Using this information we can then compute the actual motion of the object.

In this work we are interested in the estimation of the actual rigid motion of an object or more generally the motion between the coordinate axis of the observation frames using 3D observation, as depicted in Fig. 1. We can describe the position and orientation of the coordinate frame B_i relative to the frame A by using a state vector X_i . This state vector satisfies a dynamic model known as the plant model

$$X_i = \Phi_{i/i-1}(X_{i-1}, W_i), \quad (1)$$

where W_i denotes random noise with zero mean, normally distributed statistics and with known statistics. The measurement of a line of a scene L_i is corrupted

*To whom correspondence should be addressed.

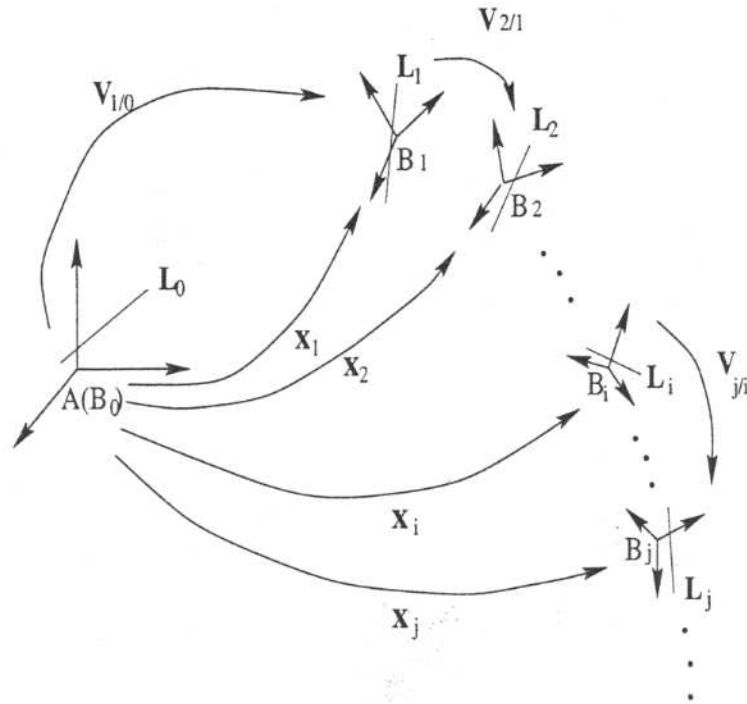


Figure 1. Coordinate frames for observation of rigid motions.

by noise with known statistics, zero mean, normally distributed and uncorrelated with respect to W_i . The relationship between the measurements and the state is given by the following measurement function model

$$f(L_0, L_i, X_i, V_0, V_i) = 0. \quad (2)$$

In this noisy scenario the task we have is to find the best estimate of the state variable vector \hat{X}_i .

In the literature we distinguish two main groups: the least squares solutions or batch and the recursive methods [20]. The key characteristic of these methods is whether they compute the translation and rotation transformations simultaneously or decoupled. Arun et al. use point sets by means of least-squares fitting [1] to estimate the rotation and translation separately. In contrast Bayro-Corrochano and Daniilidis use the motion of lines to estimate the motion displacement components simultaneously [6, 10] for solving the hand-eye problem. Zhang and Faugeras, using Plücker line sets, propose an analytical solution in terms of least-squares to estimate the motion displacement separately [22]. All these approaches are used for static motion estimation.

The application of the Kalman filter as a recursive minimum variance estimator has been popular since the sixties [16, 17, 21]. In order to estimate dynamic

motion parameters, the authors used the Kalman filter together with different types of state variable representations. For instance, Bar-Itzhack et al. used point sets for the quaternion extended Kalman filter to estimate the dynamic rotation [3], and Zhang and Faugeras, used line segments with their midpoints to estimate all dynamic motion parameters with a standard extended Kalman filter [22]. Recently, Azarbayejani and Pentland [2] applied the extended Kalman filter for the estimation of motion and structure using relative orientation constraints in terms of quaternions.

In this paper, we present the development of a novel extended Kalman filter in the geometric algebra framework. This recursive filter has the virtue of estimating simultaneously the translation and rotation components. This characteristic has not been achieved with the Kalman filter before because the authors could not overcome the singularities when dealing with Plücker lines. The key idea for the filter design is to work in the 4D geometric algebra, called motor algebra, and to represent the motion of the measurement frames as a motion of lines. The paper shows that the motor extended Kalman filter is an attractive estimation approach, particularly in case of dynamic motion problems.

The paper is organized as follows: Section two outlines the geometric algebra. Section three introduces briefly the 4D motor algebra. Section four is devoted to

the linear modeling of the 3D motion of lines using the motor algebra. Section five gives a brief introduction to Kalman filter techniques required for the understanding of the rotor Kalman filter. Section six describes the motor extended Kalman filter as a natural generalization. Section seven introduces briefly the analytical method for the recovering of the 3D motion parameters using line observations. In section eight, we compare the MEKF algorithm and an analytical method using simulated data of the screw motion of an object. Then we apply the MEKF algorithm to estimate the relative motion of a robot manipulator which is guided using a stereo vision system. Finally, section nine is dedicated to discussion and conclusions.

In this paper, the signature of a geometric algebra \mathcal{G}_n will be clearly specified by $\mathcal{G}_{p,q,r}$ where p , q and r stand for the numbers of basis vectors which square to $+1$, -1 and 0 respectively. An even subalgebra will be specified by $\mathcal{G}_{p,q,r}^+$. We denote scalars with lower case, matrices with upper case, vectors in 3D with bold lower case and vectors in 4D with bold upper case.

2. Geometric Algebra

Clifford algebra is a coordinate-free approach to geometry based on the algebras of Grassmann [11] and Clifford [9]. The approach to Clifford algebra we adopt here was pioneered in the 1960's by David Hestenes [12] and later, with Garret Sobczyk, developed into a unified language for mathematics and physics [13, 14]. Some preliminary applications of geometric algebra to the field of computer vision and neural computing have already been given [5, 7].

2.1. Basic Definitions

Let \mathcal{G}_n denote the geometric algebra of n -dimensions. The *geometric* or *Clifford product* of two vectors \mathbf{a} and \mathbf{b} is written \mathbf{ab} and can be expressed as a sum of its

symmetric and antisymmetric parts

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b}, \quad (3)$$

where the inner product $\mathbf{a} \cdot \mathbf{b}$ and the outer product $\mathbf{a} \wedge \mathbf{b}$ are defined by

$$\mathbf{a} \cdot \mathbf{b} = \frac{1}{2}(\mathbf{ab} + \mathbf{ba}) \quad (4)$$

$$\mathbf{a} \wedge \mathbf{b} = \frac{1}{2}(\mathbf{ab} - \mathbf{ba}). \quad (5)$$

The inner product of two vectors is the standard *scalar* or *dot* product and produces a scalar. The outer or wedge product of two vectors is a new quantity which we call a *bivector*. We think of a bivector as an oriented area in the plane containing \mathbf{a} and \mathbf{b} , formed by sweeping \mathbf{a} along \mathbf{b} , see Fig. 2(a).

The bivector $\mathbf{b} \wedge \mathbf{a}$ has the opposite orientation and is antisymmetric as given in Eq. (5). The outer product is immediately generalizable to higher dimensions. The outer product of k vectors is a k -vector or k -blade, and is said to have *grade* k , see Fig. 2(b). A *multivector* (a linear combination of objects of different grades) is said to be *homogeneous* if it contains terms of only a single grade.

In a space of 3 dimensions we can construct a trivector $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$, but no 4-vectors exist since there is no possibility of sweeping the volume element $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$ over a 4th dimension. The highest grade element in a space is called the *pseudoscalar*. The unit pseudoscalar is denoted by I and is crucial when discussing duality.

2.2. The Geometric Algebra of n -D Space

In an n -dimensional space we can introduce an orthonormal basis of vectors $\{\sigma_i\} i = 1, \dots, n$, such that

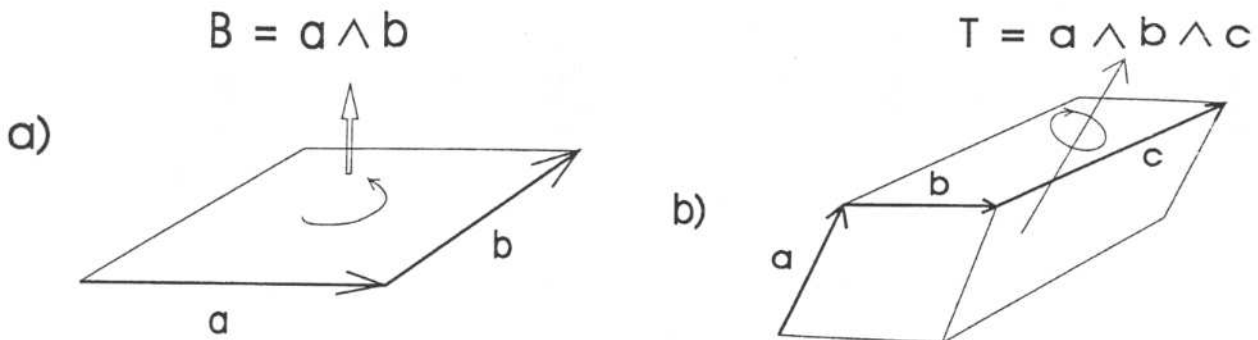


Figure 2. (a) The directed area, or bivector, $\mathbf{a} \wedge \mathbf{b}$. (b) The oriented volume, or trivector, $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$.

$\sigma_i \cdot \sigma_j = \delta_{ij}$. This leads to a basis for the entire algebra:

$$1, \quad \{\sigma_i\}, \quad \{\sigma_i \wedge \sigma_j\}, \quad \{\sigma_i \wedge \sigma_j \wedge \sigma_k\}, \dots, \quad \sigma_1 \wedge \sigma_2 \wedge \dots \wedge \sigma_n. \quad (6)$$

Note that the basis vectors are not represented by bold symbols. Any multivector can be expressed in terms of this basis. In this paper a geometric algebra \mathcal{G}_n is of the form $\mathcal{G}_{p,q,r}$, where p , q , and r stand for the number of basis vectors which square to 1, -1 and 0, respectively, where $n = p + q + r$. Its even subalgebra will be denoted by $\mathcal{G}_{p,q,r}^+$.

In the n -D space there are multivectors of grade 0 (scalars), grade 1 (vectors), grade 2 (bivectors), grade 3 (trivectors), etc. . . up to grade n . Any two such multivectors can be multiplied using the geometric product. Consider two multivectors A_r and B_s of grades r and s respectively. The geometric product of A_r and B_s can be written as

$$A_r B_s = \langle AB \rangle_{r+s} + \langle AB \rangle_{r+s-2} + \dots + \langle AB \rangle_{|r-s|} \quad (7)$$

where $\langle M \rangle_t$ is used to denote the t -grade part of multivector M , e.g. consider the geometric product of two vectors $ab = \langle ab \rangle_0 + \langle ab \rangle_2 = a \cdot b + a \wedge b$. As simple illustration the geometric product of $A = 5\sigma_3 + 3\sigma_1\sigma_2$ and $b = 9\sigma_2 + 7\sigma_3$ is

$$\begin{aligned} Ab &= 35(\sigma_3)^2 + 27\sigma_1(\sigma_2)^2 + 45\sigma_3\sigma_2 + 21\sigma_1\sigma_2\sigma_3 \\ &= 35 + 27\sigma_1 - 45\sigma_2\sigma_3 + 21I, \end{aligned} \quad (8)$$

note that $\sigma_i \sigma_i = (\sigma_i)^2 = \sigma_i \cdot \sigma_i = 1$ and $\sigma_i \sigma_j = \sigma_i \wedge \sigma_j$, where the geometric product of equal unit basis vectors equals 1 and of different ones equals to their wedge, which for simple notation can be omitted.

2.3. The Geometric Algebra of 3-D Space

The basis for the geometric algebra $\mathcal{G}_{3,0,0}$, of the 3-D space has $2^3 = 8$ elements and is given by:

$$\underbrace{1}_{\text{scalar}}, \underbrace{\{\sigma_1, \sigma_2, \sigma_3\}}_{\text{vectors}}, \underbrace{\{\sigma_1\sigma_2, \sigma_2\sigma_3, \sigma_3\sigma_1\}}_{\text{bivectors}}, \underbrace{\{\sigma_1\sigma_2\sigma_3\}}_{\text{trivector}} \equiv I. \quad (9)$$

It can easily be verified that the trivector or pseudoscalar $\sigma_1\sigma_2\sigma_3$ squares to -1 and commutes with all multivectors in the 3-D algebra. We therefore give it

the symbol I ; noting that this is not the uninterpreted commutative scalar imaginary j used in quantum mechanics and engineering.

2.4. Rotors

Multiplication of the three basis vectors σ_1 , σ_2 , and σ_3 by I results in the three basis bivectors $\sigma_1\sigma_2 = I\sigma_3$, $\sigma_2\sigma_3 = I\sigma_1$ and $\sigma_3\sigma_1 = I\sigma_2$. These simple bivectors rotate vectors in their own plane by 90° , e.g. $(\sigma_1\sigma_2)\sigma_2 = \sigma_1$, $(\sigma_2\sigma_3)\sigma_2 = -\sigma_3$ etc. Identifying the i , j , k of the quaternion algebra with $I\sigma_1$, $-I\sigma_2$, $I\sigma_3$, the famous Hamilton relations $i^2 = j^2 = k^2 = ijk = -1$ can be recovered. Since the i , j , k are bivectors, it comes as no surprise that they represent 90° rotations in orthogonal directions and provide a well-suited system for the representation of general 3D rotations, see Fig. 3.

In geometric algebra a rotor (short name for rotator), R , is an even-grade element of the algebra which satisfies $R\tilde{R} = 1$, where \tilde{R} stands for the conjugate of R . If $\mathcal{A} = \{a_0, a_1, a_2, a_3\} \in \mathcal{G}_{3,0,0}$ represents a unit quaternion, then the rotor which performs the same rotation is simply given by

$$\begin{aligned} R &= a_0 + a_1 I\sigma_1 - a_2 I\sigma_2 + a_3 I\sigma_3 \\ &= \underbrace{a_0}_{\text{scalar}} + \underbrace{a_1\sigma_{23} - a_2\sigma_{31} + a_3\sigma_{12}}_{\text{bivectors}}. \end{aligned} \quad (10)$$

The quaternion algebra is therefore seen to be a subset of the geometric algebra of 3-space. The conjugate of a rotor is given by

$$\tilde{R} = a_0 - a_1\sigma_{23} + a_2\sigma_{31} - a_3\sigma_{12}. \quad (11)$$

A rotation can be performed by a pair of reflections, see Fig. 3. It can easily be shown that the result of reflecting a vector a in the plane perpendicular to a unit vector n is $a_\perp - a_\parallel = a' = -nan^{-1}$, where a_\perp and a_\parallel respectively denote projections of a perpendicular and parallel to n . Thus, a reflection of a in the plane perpendicular to n , followed by a reflection in the plane perpendicular to another unit vector m gives the new vector $b = -m(-nan^{-1})m^{-1} = (mn)a(mn)^{-1} = Ra\tilde{R}$. Using the geometric product, we can show that the rotor R of Eq. (10) is a multivector consisting of both a scalar part and a bivector part, i.e. $R = mn = m \cdot n + m \wedge n$. These components correspond to the scalar and vector parts of an equivalent unit quaternion in $\mathcal{G}_{3,0,0}$. Considering the scalar and the bivector parts, we can further

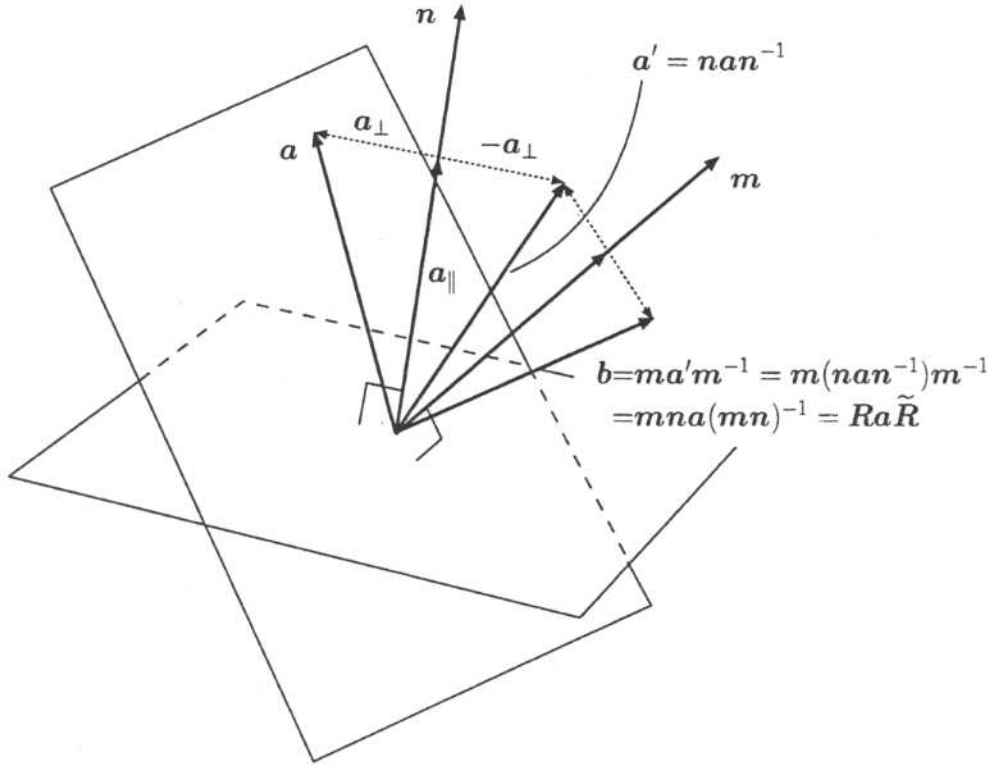


Figure 3. The rotor in the 3D space formed by a pair of reflections.

write the Euler representation of a rotor as follows

$$R = e^{\frac{\theta}{2} \mathbf{n}} = \cos \frac{\theta}{2} + \mathbf{n} \sin \frac{\theta}{2}, \quad (12)$$

where the rotation axis $\mathbf{n} = n_1\sigma_2\sigma_3 + n_2\sigma_3\sigma_1 + n_3\sigma_1\sigma_2$ is spanned by the bivector basis.

The transformation in terms of a rotor $\mathbf{a} \mapsto R\mathbf{a}\tilde{R} = \mathbf{b}$ is a very general way of handling rotations; it works for multivectors of any grade and in spaces of any dimension, in contrast to quaternion calculus. Rotors combine in a straightforward manner, i.e. a rotor R_1 followed by a rotor R_2 is equivalent to a total rotor R where $R = R_1R_2$.

3. The Motor Algebra

Clifford introduced the motors with the name biquaternions [8, 21]. Motor is the abbreviation of “moment and vector”. Motors are the dual numbers for 3D kinematics with the necessary condition of $I^2 = 0$. They can be found in the special 4D even subalgebra of $\mathcal{G}_{3,0,1}$ or motor algebra. This even subalgebra denoted by $\mathcal{G}_{3,0,1}^+$ is spanned by a basis of scalars, bivectors and

pseudoscalars,

$$\underbrace{1}_{\text{scalar}}, \underbrace{\gamma_2\gamma_3, \gamma_3\gamma_1, \gamma_1\gamma_2, \gamma_4\gamma_1, \gamma_4\gamma_2, \gamma_4\gamma_3}_{\substack{\text{6 bivectors} \\ \underbrace{I}_{\text{unit pseudoscalar}}}}. \quad (13)$$

Note that the bivectors in the basis correspond to the same basis for spanning 3D lines. Also note that the dual of a scalar is the pseudoscalar P and the duals of the first three basis bivectors are the next three, for example the dual of $\gamma_2\gamma_3$ is $I\gamma_2\gamma_3$ or $\gamma_4\gamma_1$.

3.1. Motors, Rotors and Translators in $\mathcal{G}_{3,0,1}^+$

Since a rigid motion consists of a rotation and a translation, it should be possible to express a motor in terms of individual operators like rotors and translators. The motor action can be described basically in terms of two steps: rotate one axis until its direction is parallel to another axis then shift it to overlap the another one, see Fig. 4. Note that the indicated vectors in the figure will be represented in next sections as bivectors.

Let us now express this procedure algebraically. First, let us consider a simple rotor in its Euler

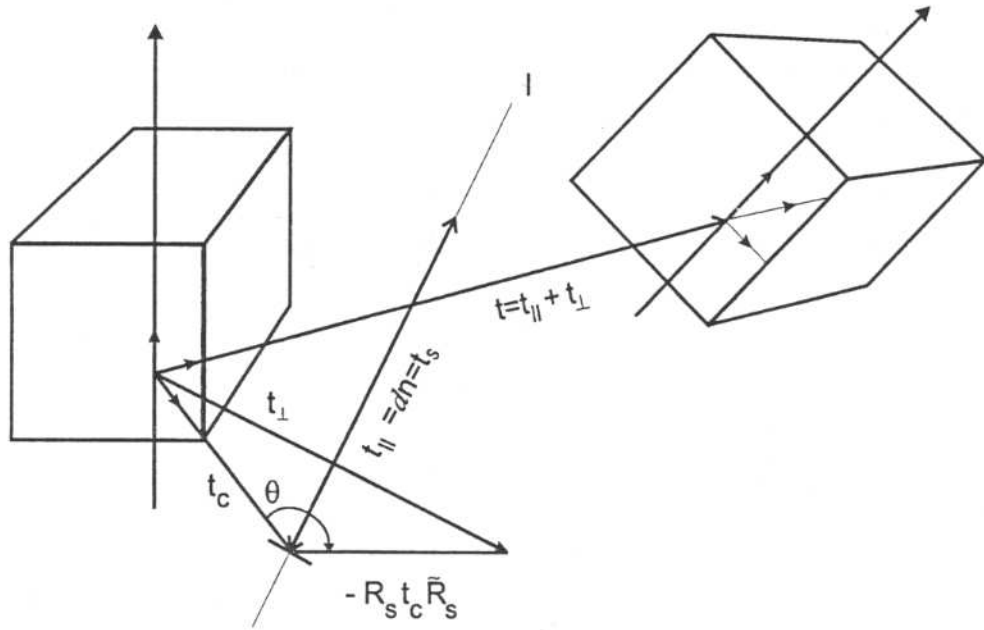


Figure 4. Screw motion of an object about the line axis l with t_s longitudinal displacement by d and rotation R_s with angle θ .

representation for a rotation with angle θ . The rotor of a screw motion, should be represented in terms of a screw axis line as follows

$$\begin{aligned} R_s &= a_0 + a_s n + I a_s n \wedge t_c = a_c + a_s (n + I m) \\ &= \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right) (n + I m) \\ &= \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right) l. \end{aligned} \quad (14)$$

Note that the line is expressed using a unit bivector for direction n and dual bivector for the momentum $m = n \wedge t_c$.

The motor is defined by sliding along the rotation axis line l the distance $t_s = dn$. Since a motor is applied from the left and its conjugated from the right we use the half of t_s when we define the motor

$$\begin{aligned} M &= T_s R_s = \left(1 + I \frac{t_s}{2}\right) (a_0 + a + I a \wedge t_c) \\ &= \left(1 + I \frac{dn}{2}\right) (a_c + a_s n + I a_s n \wedge t_c) \\ &= a_c + a_s n + I a_s n \wedge t_c + I \frac{d}{2} a_c n - I \frac{d}{2} a_s n n \\ &= \left(a_c - I \frac{d}{2} a_s\right) + \left(a_s + I a_c \frac{d}{2}\right) (n + I n \wedge t_c) \\ &= \left(a_c - I a_s \frac{d}{2}\right) + \left(a_s + I a_c \frac{d}{2}\right) l. \end{aligned} \quad (15)$$

Note that this expression of the motor makes explicit the unitary screw axis line l . Now let us express a motor as an Euler representation. Substituting the constants $a_c = \cos(\frac{\theta}{2})$ and $a_s = \sin(\frac{\theta}{2})$ in the motor Eq. (15) and using the property of scalar functions with dual argument we get

$$\begin{aligned} M &= T_s R_s = \left(\cos\left(\frac{\theta}{2}\right) - I \sin\left(\frac{\theta}{2}\right) \frac{d}{2}\right) \\ &\quad + \left(\sin\left(\frac{\theta}{2}\right) + I \cos\left(\frac{\theta}{2}\right) \frac{d}{2}\right) l \\ &= \cos\left(\frac{\theta}{2} + I \frac{d}{2}\right) + \sin\left(\frac{\theta}{2} + I \frac{d}{2}\right) l. \end{aligned} \quad (16)$$

If we want to express the motor using only a rotor and its conjugated given by

$$\tilde{R} = r_0 - r_1 \sigma_2 \sigma_3 - r_2 \sigma_3 \sigma_1 - r_3 \sigma_1 \sigma_2 = r_0 - \mathbf{r}, \quad (17)$$

we proceed as follows

$$\begin{aligned} M &= T_s R_s = \left(1 + I \frac{t_s}{2}\right) R_s \\ &= R_s + I \frac{t_s}{2} R_s = R_s + I R'_s. \end{aligned} \quad (18)$$

We can now express the bivector t_s in terms of the rotors

$$R'_s \tilde{R}_s = \left(\frac{t_s}{2} R_s\right) \tilde{R}_s \quad (19)$$

so that

$$t_s = 2R'_s \tilde{R}. \quad (20)$$

Figure 4 shows that t is a 3D distance vector referred to coordinate system of the rotation axis of an object and t_{\parallel} or t_s is a bivector along the motor axis line. The distance t , considered here as a bivector can be computed in terms of the bivectors t_c and t_s as follows

$$\begin{aligned} t &= t_{\perp} + t_{\parallel} = t_{\perp} + t_s = (t_c - R_s t_c \tilde{R}_s) + (t \cdot n)n \\ &= (t_c - R_s t_c \tilde{R}_s) + dn = t_c - R_s t_c \tilde{R}_s + t_s \\ &= (t_c - R_s t_c \tilde{R}_s) + (2R'_s \tilde{R}_s). \end{aligned} \quad (21)$$

3.2. Properties of Motors

A general motor can be expressed as

$$M_{\alpha} = \alpha M \quad (22)$$

where $\alpha \in R$ and M is a unit motor as in the previous sections. The norm of a unit motor M is defined by

$$\begin{aligned} |M| &= M\tilde{M} = T_s R_s \tilde{R}_s \tilde{T}_s = \left(1 + I \frac{t_s}{2}\right) R_s \tilde{R}_s \left(1 - I \frac{t_s}{2}\right) \\ &= 1 + I \frac{t_s}{2} - I \frac{t_s}{2} = 1, \end{aligned} \quad (23)$$

where \tilde{M} is the conjugate of the motor, defined by

$$\tilde{M} = \tilde{T}R = \tilde{R}\tilde{T} \quad (24)$$

and 1 is the identity of the motor multiplication. Now using the Eq. (18), and considering the unit motor magnitude we find two useful properties

$$\begin{aligned} |M| &= M\tilde{M} = (R_s + IR'_s)(\tilde{R}_s + I\tilde{R}'_s) \\ &= R_s \tilde{R}_s + I(\tilde{R}_s R'_s + \tilde{R}'_s R_s) = 1. \end{aligned} \quad (25)$$

This requires that

$$R_s \tilde{R}_s = 1 \quad (26)$$

$$\tilde{R}_s R'_s + \tilde{R}'_s R_s = 2(r_0 r'_0 - \mathbf{r} \cdot \mathbf{r}') = 0. \quad (27)$$

The combination of two rigid motions can be expressed using two motors. The resultant motor

describes the overall displacement, namely

$$\begin{aligned} M_c &= M_a M_b = (R_{s_a} + IR'_{s_a})(R_{s_b} + IR'_{s_b}) \\ &= R_{s_a} R_{s_b} + I(R_{s_a} R'_{s_b} + R'_{s_a} R_{s_b}) \\ &= R_{s_c} + IR'_{s_c}. \end{aligned} \quad (28)$$

Note that pure rotations combine multiplicatively, whereas the dual parts containing the translation combine additively.

Using the Eq. (18), let us express a motor in terms of a scalar, bivector, dual scalar and dual bivector

$$\begin{aligned} M &= T_s R_s = R_s + IR'_s = (a_0 + a_1 \gamma_2 \gamma_3 + a_2 \gamma_3 \gamma_2 \\ &\quad + a_3 \gamma_2 \gamma_1) + I(b_0 + b_1 \gamma_2 \gamma_3 + b_2 \gamma_3 \gamma_2 + b_3 \gamma_2 \gamma_1) \\ &= (a_0 + \mathbf{a}) + I(b_0 + \mathbf{b}). \end{aligned} \quad (29)$$

We can use another notation to enhance the components of the real and dual parts of the motor as follows

$$M = (a_0, \mathbf{a}) + I(b_0, \mathbf{b}), \quad (30)$$

where each term within the parenthesis consists of a scalar part and a 3D bivector.

4. Linear Modeling of the Motion of 3D lines using the Motor Algebra

The modeling of the 3-D motion of lines using the motor algebra $\mathcal{G}_{3,0,1}^+$ takes place in a 4D space where the rotation and translation are operators applied multiplicatively; as a result the 3D general motion becomes linear. Having a linear method we can then compute, for example, the unknown rotation and translation simultaneously in such cases as the hand-eye problem [6] or estimate dynamic motion using the motor extended Kalman filter. In these problems, if we use the 3D geometric algebra $\mathcal{G}_{3,0,0}$, we are unfortunately compelled to compute the translation decoupled of rotation, increasing therefore the inaccuracy.

4.1. Representation of 3D Lines Using the Motor Algebra

The basis bivectors of the algebra of motors $\mathcal{G}_{3,0,1}^+$, spans the line space in 4D. Assume that two points $X_1 = (X_{11}, X_{12}, X_{13}, 1)$ and $X_2 = (X_{21}, X_{22}, X_{23}, 1)$ in the space of $\mathcal{G}_{3,0,1}$ lying on the hyper-plane $X_4 = 1$ belong to the line L_d . Note that we use now upper

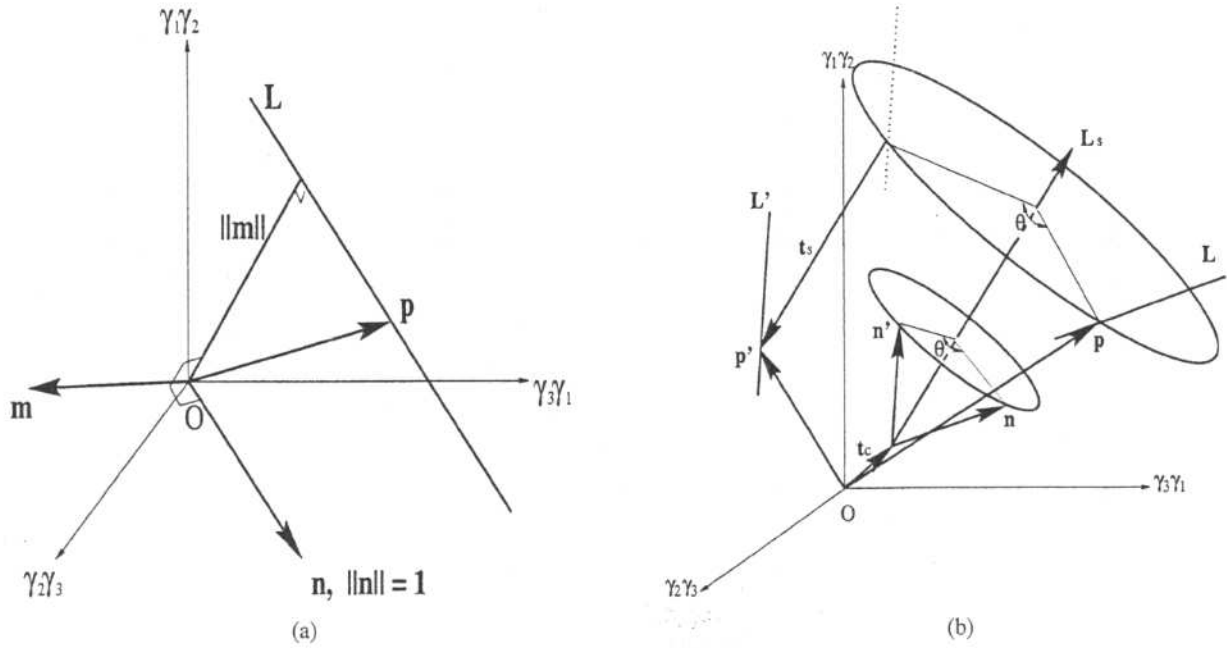


Figure 5. (a) The moment and the direction of the line using dual bivector basis. (b) The screw motion of a line.

letters for points in the 4D space to differentiate them from the points of the 3D space in the $\mathcal{G}_{3,0,0}$. The line belongs to $\mathcal{G}_{3,0,1}^+$ and therefore it can be defined simply as an outer product of these points, i.e.

$$\begin{aligned} L_d = X_1 \wedge X_2 = & (X_{12}X_{23} - X_{13}X_{22})\gamma_2\gamma_3 \\ & + (X_{13}X_{21} - X_{11}X_{23})\gamma_3\gamma_1 \\ & + (X_{11}X_{22} - X_{12}X_{21})\gamma_1\gamma_2 + (X_{21} - X_{11})\gamma_4\gamma_1 \\ & + (X_{22} - X_{12})\gamma_4\gamma_2 + (X_{23} - X_{13})\gamma_4\gamma_3. \end{aligned} \quad (31)$$

Since this equation consists only of bivectors, it can be expressed straightforwardly in terms of the bivector basis, namely

$$\begin{aligned} L_d = & (L^{23}\gamma_2\gamma_3 + L^{31}\gamma_3\gamma_1 + L^{12}\gamma_1\gamma_2) \\ & + (L^{41}\gamma_4\gamma_1 + L^{42}\gamma_4\gamma_2 + L^{43}\gamma_4\gamma_3) \\ = & (L^{23}\gamma_2\gamma_3 + L^{31}\gamma_3\gamma_1 + L^{12}\gamma_1\gamma_2) \\ & + I(L^{41}\gamma_2\gamma_3 + L^{42}\gamma_3\gamma_1 + L^{43}\gamma_1\gamma_2), \end{aligned} \quad (32)$$

where $\gamma_4\gamma_1 = I\gamma_2\gamma_3$, $\gamma_4\gamma_2 = I\gamma_3\gamma_1$ and $\gamma_4\gamma_3 = I\gamma_1\gamma_2$. To agree with the conventional line representation, we interchange the coefficients of the bivectors and we express three of them as the dual part as follows

$$\begin{aligned} L_d = & (L^{41}\gamma_2\gamma_3 + L^{42}\gamma_3\gamma_1 + L^{43}\gamma_1\gamma_2) \\ & + (L^{23}\gamma_2\gamma_3 + L^{31}\gamma_3\gamma_1 + L^{12}\gamma_1\gamma_2), \end{aligned} \quad (33)$$

Note that this is equivalent to the line expression using Plücker coordinates.

The real part can be seen as the direction of this line, and the dual part as the moment of the plane where the line lies. This plane crosses the origin and it is described by the outer product between n and any vector p touching the line, i.e.

$$L_d = n + n \wedge p = n + Im. \quad (34)$$

Note that the term $n \times p$ is orthogonal to the bivector n , so we can use the dual bivector representation for the moment m . The representation of the line according the Eq. (34) is given in Fig. 5(a).

This line representation using dual numbers is easier to understand and to manipulate algebraically and it is fully equivalent to the one in terms of Plücker coordinates. Using the notation with brackets the line equation reads

$$L_d = (0, n) + I(0, m). \quad (35)$$

where the n and m are spanned with a 3D bivector basis. Note that the first element of within the parenthesis corresponds to the scalar and the second to the bivector part.

4.2. Modeling the Motion of Lines Using the Motor Algebra

The motion of a 3D line, or the screw motion of a line can also be seen as a rotation of the line about the axis

line L_s , followed by its translation along this axis line, as depicted in Fig. 5(b). Note that in the figure the line L_s is shifted from the origin in t_c . In Section 3.1 we explained in detail how a motor implements the screw motion of an object about a line axis. Considering now the 3D line as a geometric object, and using the line Eq. (34), we can express the motion of a 3D line as follows

$$\begin{aligned} L' &= n' + Im' = M(L)\tilde{M} \\ &= T_s R_s (n + Im) \tilde{R}_s \tilde{T}_s. \end{aligned} \quad (36)$$

This equation can be expressed purely in terms of rotors as follows

$$\begin{aligned} L' &= \left(1 + I \frac{t_s}{2}\right) R_s (n + Im) \tilde{R}_s \left(1 - I \frac{t_s}{2}\right) \\ &= \left(1 + I \frac{t_s}{2}\right) \left(R_s n \tilde{R}_s + I R_s m \tilde{R}_s - I R_s n \tilde{R}_s \frac{t_s}{2}\right) \\ &= R_s n \tilde{R}_s + I \left(-R_s n \tilde{R}_s \frac{t_s}{2} + \frac{t_s}{2} R_s n \tilde{R}_s + R_s m \tilde{R}_s\right) \\ &= R_s n \tilde{R}_s + I (R_s n \tilde{R}_s' + R_s' n \tilde{R}_s + R_s m \tilde{R}_s) \end{aligned} \quad (37)$$

This equation is very useful for estimating the rotation and translation simultaneously as in the case of hand-eye calibration [6] or for the algorithm for the motor extended Kalman filter.

4.3. Representation of the Line Motion Model in Linear Algebra

The line motion model presented in the last section uses the geometric algebra $\mathcal{G}_{3,0,1}^+$. Because the EKF computer algorithm is implemented using techniques of linear algebra, we should also formulate the line motion model $L' = ML\tilde{M}$ in the frame of linear algebra.

Let us start using rotor relations. The multiplication of two rotors U and V in geometric algebra $\mathcal{G}_{3,0,1}^+$ reads

$$\begin{aligned} W = UV &= (u_0 + u)(v_0 + v) \\ &= u_0 v_0 + u \cdot v + u_0 v + v_0 u + u \wedge v. \end{aligned} \quad (38)$$

Multiplication of these two rotors in linear algebra is

$$W = \mathcal{U}_{Rl} V = \mathcal{V}_{Rr} U, \quad (39)$$

where $U = (u_0 \ u_1 \ u_2 \ u_3)^T$, $V = (v_0 \ v_1 \ v_2 \ v_3)^T$ and

$$\begin{aligned} \mathcal{U}_{Rl} &= \begin{pmatrix} u_0 & -u_1 & -u_2 & -u_3 \\ u_1 & u_0 & u_3 & -u_2 \\ u_2 & -u_3 & u_0 & u_1 \\ u_3 & u_2 & -u_1 & u_0 \end{pmatrix}, \\ \mathcal{V}_{Rr} &= \begin{pmatrix} v_0 & -v_1 & -v_2 & -v_3 \\ v_1 & v_0 & -v_3 & v_2 \\ v_2 & v_3 & v_0 & -v_1 \\ v_3 & -v_2 & v_1 & v_0 \end{pmatrix}. \end{aligned} \quad (40)$$

We call \mathcal{U}_{Rl} “left-multiplication matrix of rotor U ” and \mathcal{V}_{Rr} “right-multiplication matrix of rotor V ”.

The product of $S = U + IU'$ and $T = V + IV'$ in geometric algebra $\mathcal{G}_{3,0,1}^+$ gives

$$\begin{aligned} Q = ST &= (U + IU')(V + IV') \\ &= UV + I(UV' + U'V), \end{aligned} \quad (41)$$

where U, U', V and V' are all expressed in the form of rotors. Multiplication of these two motors in terms of matrices

$$Q = \mathcal{S}_{Ml} T = \mathcal{T}_{Mr} S, \quad (42)$$

where

$$\begin{aligned} S &= (u_0 \ u_1 \ u_2 \ u_3 \ u_0' \ u_1' \ u_2' \ u_3')^T, \\ T &= (v_0 \ v_1 \ v_2 \ v_3 \ v_0' \ v_1' \ v_2' \ v_3')^T, \\ \mathcal{S}_{Ml} &= \begin{pmatrix} \mathcal{U}_{Rl} & \mathbf{0}_{4 \times 4} \\ \mathcal{U}_{Rl}' & \mathcal{U}_{Rl} \end{pmatrix}, \quad \mathcal{T}_{Mr} = \begin{pmatrix} \mathcal{V}_{Rr} & \mathbf{0}_{4 \times 4} \\ \mathcal{V}_{Rr}' & \mathcal{V}_{Rr} \end{pmatrix}, \end{aligned} \quad (43)$$

We call \mathcal{S}_{Ml} “left-multiplication matrix of motor S ” and \mathcal{T}_{Mr} “right-multiplication matrix of motor T ”.

To convert the line motion model of Eq. (36) to matrix algebra we can handle the real and dual components n, m, n' and m' of the lines L and L' as rotors with zero scalar. By right multiplication of both sides of Eq. (36) by M we get

$$L'M - ML = 0. \quad (44)$$

This results in the following linear motion equation

$$(\mathcal{L}_{Ml}' - \mathcal{L}_{Mr})M = \mathcal{A}_M M = 0, \quad (45)$$

this matrix representation of the line motion was suggested in [10]. The constraints of Eqs. (26) and (27),

respectively, now are

$$\mathbf{R}^T \mathbf{R} = \mathbf{I}, \quad (46)$$

$$\mathbf{R}^T \mathbf{R}' = \mathbf{0}, \quad (47)$$

with $\mathbf{R} = (r_0 \ r_1 \ r_2 \ r_3)^T$, $\mathbf{R}' = (r'_0 \ r'_1 \ r'_2 \ r'_3)^T$ and $\mathbf{M} = \mathbf{R} + \mathbf{I}\mathbf{R}'$.

These properties will be used for the implementation of the MEKF algorithm in the subsection 6.2.

5. Recursive Estimation using Kalman Filter Techniques

The Kalman filter is a linear recursive algorithm, that is unbiased and of minimum variance. It is used to estimate optimally the unknown state of a linear dynamic system using noisy data which is taken at discrete real-time intervals. The extended Kalman filter (EKF) approach modifies the standard Kalman filter (used for linear systems) in order to treat noisy nonlinear systems. It starts with an initial guess, then updates continually the predicted state with new measurements. Unfortunately, if the disturbances are so large that the linearization is inadequate to describe the system, the filter will not converge to a reasonable estimate. First, we give a brief outline of the Kalman filter and of the extended Kalman filter, using this background we will explain thereafter the rotor and motor extended Kalman filters. For a more complete introduction the reader should refer to [21].

5.1. The Kalman Filter

Let us describe the Eq. (1) of a dynamical system by a linear difference state equation as follows

$$\mathbf{X}_i = \Phi_{i/i-1} \mathbf{X}_{i-1} + \mathbf{W}_i. \quad (48)$$

The state of the system at t_i is given by the n -dimensional vector \mathbf{X}_i . The term $\Phi_{i/i-1}$ is an $n \times n$ transition matrix and \mathbf{W}_i is the random error with the known first- and second-order characteristics

$$E[\mathbf{W}_i] = \mathbf{0}, \quad i = 0, 1, \dots \quad (49)$$

$$E[\mathbf{W}_i \mathbf{W}_j^T] = \mathbf{Q}_i \delta_{ij} \quad (50)$$

where δ_{ij} is the Kronecker delta function. The matrix \mathbf{Q}_i is assumed to be nonnegative-definite.

Suppose that at each time t_i there is available an m -dimensional vector of measurement \mathbf{Z}_i that is linearly related to the state and which is corrupted by the additive noise \mathbf{V}_i . This is the so called observation equation

$$\mathbf{Z}_i = \mathcal{H}_i \mathbf{X}_i + \mathbf{V}_i, \quad (51)$$

where \mathcal{H}_i is a known $m \times n$ observation matrix, and the vector \mathbf{V}_i a random error with the known statistics

$$E[\mathbf{V}_i] = \mathbf{0}, \quad i = 0, 1, \dots \quad (52)$$

$$E[\mathbf{V}_i \mathbf{V}_j^T] = \mathbf{C}_i \delta_{ij}, \quad (53)$$

where the matrix \mathbf{C}_i is assumed to be nonnegative-definite [21].

Further, assume that the random processes \mathbf{W}_i and \mathbf{V}_i are uncorrelated, i.e. for each i, j

$$E[\mathbf{W}_i \mathbf{V}_j^T] = \mathbf{0}, \quad (54)$$

where $\mathbf{0}$ is the zero matrix.

Given the preceding models (48) and (51), we shall determine an estimate $\hat{\mathbf{X}}_i$ of the state at t_i as a linear combination of an estimate $\hat{\mathbf{X}}_{i-1}$ at t_{i-1} and the data \mathbf{Z}_i measured at time t_i . By defining an unknown $(n \times m)$ gain matrix \mathcal{K}_i , the estimate $\hat{\mathbf{X}}_i$ is given by

$$\hat{\mathbf{X}}_i = \Phi_{i/i-1} \hat{\mathbf{X}}_{i-1} + \mathcal{K}_i [\mathbf{Z}_i - \mathcal{H}_i \Phi_{i/i-1} \hat{\mathbf{X}}_{i-1}] \quad (55)$$

The matrix \mathcal{K}_i is determined so that the estimate has the minimal variance. That is, the $\hat{\mathbf{X}}_i$ is chosen to minimize its mean squared error

$$E_{\text{MIN}} = \{E[(\hat{\mathbf{X}}_i - \mathbf{X}_i)^T (\hat{\mathbf{X}}_i - \mathbf{X}_i)]\}_{\text{MIN}}. \quad (56)$$

Equation (56) is equivalent to the minimization of the trace of state error covariance matrix \mathcal{P}_i

$$\begin{aligned} E_{\text{MIN}} &= \{\text{trace } \mathcal{P}_i\}_{\text{MIN}} \\ &= \{\text{trace } E[(\hat{\mathbf{X}}_i - \mathbf{X}_i)(\hat{\mathbf{X}}_i - \mathbf{X}_i)^T]\}_{\text{MIN}}. \end{aligned} \quad (57)$$

By substituting (51) in (55), and then substituting the new (55) and (48) in (57), it can be shown that the trace of matrix \mathcal{P}_i will be minimized by choosing the following optimal gain matrix \mathcal{K}_i ,

$$\mathcal{K}_i = \mathcal{P}_{i/i-1} \mathcal{H}_i^T (\mathcal{H}_i \mathcal{P}_{i/i-1} \mathcal{H}_i^T + \mathbf{C}_i)^{-1}, \quad (58)$$

where $\mathcal{P}_{i/i-1}$ is the error covariance matrix

$$\mathcal{P}_{i/i-1} = \Phi_{i/i-1} \mathcal{P}_i \Phi_{i/i-1}^T + \mathbf{Q}_i. \quad (59)$$

of the predicted state

$$\hat{X}_{i/i-1} = \Phi_{i/i-1} \hat{X}_i. \quad (60)$$

With this optimal gain matrix K_i , the matrix P_i reduces to

$$\begin{aligned} P_i &= P_{i/i-1} - K_i H_i P_{i/i-1} \\ &= (I - K_i H_i) P_{i/i-1}. \end{aligned} \quad (61)$$

Equations (55), (59), (58) and (61) constitute the Kalman filter equations for the model of the system (48) and that of the measurement (51), respectively.

From (58), we see that as the measurement error covariance matrix C_i approaches zero, the gain matrix K_i weights the residual more heavily,

$$\lim_{C_i \rightarrow 0} K_i = H_i^{-1}. \quad (62)$$

On the other hand, as the estimated state error covariance P_i approaches zero, the gain K_i weights the residual less heavily,

$$\lim_{P_i \rightarrow 0} K_i = 0. \quad (63)$$

Another way of thinking about the weighting the Kalman filter by K_i is that as the measurement error covariance matrix C_i approaches zero, the actual measurement Z_i is "trusted" more and more, while the predicted state $\Phi_{i/i-1} \hat{X}_i$ is trusted less and less. On the other hand, as the estimated state error covariance P_i approaches zero the actual measurement Z_i is trusted less and less, while the predicted state $\Phi_{i/i-1} \hat{X}_i$ (the dynamic model) is trusted more and more.

5.2. The Extended Kalman Filter

As described previously, the Kalman filter addresses the general problem of trying to estimate the state X_i of a discrete-time controlled process that is governed by a linear stochastic difference equation. But what happens if the process and (or) the relation between the measurement and the state is non-linear? Some of the most interesting and successful applications of Kalman filtering are concerned with such situations. A Kalman filter that linearizes about the current predicted state $\hat{X}_{i/i-1}$ and measurement Z_i is referred to as an extended Kalman filter or EKF.

In computer vision the measurement model is usually found to be described by a nonlinear observation equation $f_i(Z_{0,i}, X_i) = 0$, where the parameter $Z_{0,i}$ is the accurate measurement. In practice, such a measurement is affected by random errors. We assume that the measurement system is disturbed by additive white noise

$$Z_i = Z_{0,i} + V_i, \quad (64)$$

where the statistics of noise V_i are given by (52) and (53).

To apply the Kalman filter technique, we must expand the nonlinear observation equation into a first order Taylor series about $(Z_i, \hat{X}_{i/i-1})$,

$$\begin{aligned} f_i(Z_{0,i}, X_i) &= f_i(Z_i, \hat{X}_{i/i-1}) + \frac{\partial f_i(Z_i, \hat{X}_{i/i-1})}{\partial Z_{0,i}} (Z_{0,i} - Z_i) \\ &\quad + \frac{\partial f_i(Z_i, \hat{X}_{i/i-1})}{\partial X_i} (X_i - \hat{X}_{i/i-1}) + \mathcal{O}^2 = 0. \end{aligned} \quad (65)$$

By ignoring the second order term \mathcal{O}^2 , the linearized measurement Eq. (65) becomes

$$Y_i = H_i X_i + N_i, \quad (66)$$

where Y_i is the new measurement vector, N_i is the noise vector of the new measurement, and H_i is the linearized transformation matrix. The components of the Eq. (66) are given by

$$Y_i = -f_i(Z_i, \hat{X}_{i/i-1}) + \frac{\partial f_i(Z_i, \hat{X}_{i/i-1})}{\partial X_i} \hat{X}_{i/i-1},$$

$$H_i = \frac{\partial f_i(Z_i, \hat{X}_{i/i-1})}{\partial X_i},$$

$$N_i = \frac{\partial f_i(Z_i, \hat{X}_{i/i-1})}{\partial Z_{0,i}} (Z_{0,i} - Z_i),$$

$$E[N_i] = 0,$$

$$E[N_i N_i^T] = C_{i/i-1}$$

$$= \frac{\partial f_i(Z_i, \hat{X}_{i/i-1})}{\partial Z_{0,i}} C_i \frac{\partial f_i(Z_i, \hat{X}_{i/i-1})^T}{\partial Z_{0,i}},$$

where C_i is given by the statistics of measurement (53). This linearized equation (66) is a general form for the nonlinear model. We will use this form for our particular nonlinear measurement model later in Section 6.

5.3. The Rotor Extended Kalman Filter

This subsection describes an EKF algorithm to estimate rotors or quaternions. We call it Rotor Extended Kalman Filter (REKF). In the static case the measurements of points free of error satisfy

$$\mathbf{R}_{i+1} = \mathbf{R}_i, \quad (67)$$

$$\mathbf{p}'_{0,i+1} = \mathcal{R}(\mathbf{R}_{i+1})\mathbf{p}_{0,i+1}, \quad (68)$$

where $\{\mathbf{p}_{0,i}\}$ and $\{\mathbf{p}'_{0,i}\}$ are sets of points before and after rotation, respectively, and \mathbf{R}_i is the rotation quaternion for the i th pair of points $\mathbf{p}_{0,i}$ and $\mathbf{p}'_{0,i}$. $\mathcal{R}(\mathbf{R})$ is the matrix representation of \mathbf{R}

$$\begin{aligned} \mathcal{R}(\mathbf{R}) &= \begin{pmatrix} r_1^2 + r_2^2 - r_3^2 - r_4^2 & 2(r_2r_3 + r_1r_4) & 2(r_2r_4 - r_1r_3) \\ 2(r_2r_3 - r_1r_4) & r_1^2 - r_2^2 + r_3^2 - r_4^2 & 2(r_4r_3 + r_1r_2) \\ 2(r_2r_4 + r_1r_3) & 2(r_4r_3 - r_1r_2) & r_1^2 - r_2^2 - r_3^2 + r_4^2 \end{pmatrix}, \end{aligned} \quad (69)$$

where r_j for $j = 1, 2, 3, 4$ are the four components of \mathbf{R} which satisfies

$$\|\mathbf{R}\| = 1. \quad (70)$$

Let us assume that the measurements $\{\mathbf{p}_{i+1}\}$ and $\{\mathbf{p}'_{i+1}\}$ of $\{\mathbf{p}_{0,i+1}\}$ and $\{\mathbf{p}'_{0,i+1}\}$ are corrupted by the noise $\{\mathbf{n}_{i+1}\}$ and $\{\mathbf{n}'_{i+1}\}$, respectively, such that

$$\mathbf{p}_{i+1} = \mathbf{p}_{0,i+1} + \mathbf{n}_{i+1} \quad (71)$$

$$\mathbf{p}'_{i+1} = \mathbf{p}'_{0,i+1} + \mathbf{n}'_{i+1}. \quad (72)$$

Here the noise vectors $\{\mathbf{n}_{i+1}\}$ and $\{\mathbf{n}'_{i+1}\}$ are assumed to have zero mean and the known covariance matrices $\{\mathbf{C}_{i+1}\}$ and $\{\mathbf{C}'_{i+1}\}$, respectively.

We rewrite the Eq. (68) as the function f_{i+1} depending of the variables $(\mathbf{p}_{0,i+1}, \mathbf{p}'_{0,i+1}, \mathbf{R}_{i+1})$

$$\begin{aligned} f_{i+1}(\mathbf{p}_{0,i+1}, \mathbf{p}'_{0,i+1}, \mathbf{R}_{i+1}) \\ = \mathbf{p}'_{0,i+1} - \mathcal{R}(\mathbf{R}_{i+1})\mathbf{p}_{0,i+1} = \mathbf{0} \end{aligned} \quad (73)$$

Expanding this equation about $(\mathbf{p}_{i+1}, \mathbf{p}'_{i+1}, \hat{\mathbf{R}}_{i+1/i})$ in terms of first order Taylor series we get

$$\begin{aligned} f_{i+1}(\mathbf{p}_{0,i+1}, \mathbf{p}'_{0,i+1}, \mathbf{R}_{i+1}) \\ = f_{i+1}(\mathbf{p}_{i+1}, \mathbf{p}'_{i+1}, \hat{\mathbf{R}}_{i+1/i}) \\ + \frac{\partial f_{i+1}(\mathbf{p}_{i+1}, \mathbf{p}'_{i+1}, \hat{\mathbf{R}}_{i+1/i})}{\partial \mathbf{p}'_{0,i+1}} (\mathbf{p}'_{0,i+1} - \mathbf{p}'_{i+1}) \end{aligned}$$

$$\begin{aligned} + \frac{\partial f_{i+1}(\mathbf{p}_{i+1}, \mathbf{p}'_{i+1}, \hat{\mathbf{R}}_{i+1/i})}{\partial \mathbf{p}_{0,i+1}} (\mathbf{p}_{0,i+1} - \mathbf{p}_{i+1}) \\ + \frac{\partial f_{i+1}(\mathbf{p}_{i+1}, \mathbf{p}'_{i+1}, \hat{\mathbf{R}}_{i+1/i})}{\partial \mathbf{R}_{i+1}} (\mathbf{R}_{i+1} - \hat{\mathbf{R}}_{i+1/i}) + \mathcal{O}^2 \\ = \mathbf{0}, \end{aligned} \quad (74)$$

where the second order term \mathcal{O}^2 can be omitted, and

$$\frac{\partial f_{i+1}(\mathbf{p}_{i+1}, \mathbf{p}'_{i+1}, \hat{\mathbf{R}}_{i+1/i})}{\partial \mathbf{p}'_{0,i+1}} = \mathbf{1}, \quad (75)$$

$$\frac{\partial f_{i+1}(\mathbf{p}_{i+1}, \mathbf{p}'_{i+1}, \hat{\mathbf{R}}_{i+1/i})}{\partial \mathbf{p}_{0,i+1}} = -\mathcal{R}(\hat{\mathbf{R}}_{i+1/i}), \quad (76)$$

$$\frac{\partial f_{i+1}(\mathbf{p}_{i+1}, \mathbf{p}'_{i+1}, \hat{\mathbf{R}}_{i+1/i})}{\partial \mathbf{R}_{i+1}} = \frac{\partial \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{p}_{i+1}}{\partial \mathbf{R}_{i+1}}. \quad (77)$$

In order to compute the expression (77), we utilize the following vectors

$$\mathbf{p}_{i+1} \cong (\mathbf{p}_1 \quad \mathbf{p}_2 \quad \mathbf{p}_3)^T, \quad (78)$$

$$\mathbf{R}_{i+1} \cong (\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{r}_4)^T \quad (79)$$

$$\hat{\mathbf{R}}_{i+1/i} \cong (\hat{\mathbf{r}}_1 \quad \hat{\mathbf{r}}_2 \quad \hat{\mathbf{r}}_3 \quad \hat{\mathbf{r}}_4)^T, \quad (80)$$

thus we can write

$$\begin{aligned} \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{p}_{i+1} &= \begin{pmatrix} \hat{r}_1^2 + \hat{r}_2^2 - \hat{r}_3^2 - \hat{r}_4^2 & 2(\hat{r}_2\hat{r}_3 + \hat{r}_1\hat{r}_4) & 2(\hat{r}_2\hat{r}_4 - \hat{r}_1\hat{r}_3) \\ 2(\hat{r}_2\hat{r}_3 - \hat{r}_1\hat{r}_4) & \hat{r}_1^2 - \hat{r}_2^2 + \hat{r}_3^2 - \hat{r}_4^2 & 2(\hat{r}_4\hat{r}_3 + \hat{r}_1\hat{r}_2) \\ 2(\hat{r}_2\hat{r}_4 + \hat{r}_1\hat{r}_3) & 2(\hat{r}_4\hat{r}_3 - \hat{r}_1\hat{r}_2) & \hat{r}_1^2 - \hat{r}_2^2 - \hat{r}_3^2 + \hat{r}_4^2 \end{pmatrix} \\ &\begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix}_{i+1} = (\hat{\mathbf{p}}'_1 \quad \hat{\mathbf{p}}'_2 \quad \hat{\mathbf{p}}'_3)^T = \hat{\mathbf{p}}'_{i+1/i}. \end{aligned} \quad (81)$$

The derivative of the vector $\hat{\mathbf{p}}'_{i+1/i}$ with respect to a vector \mathbf{R}_{i+1} is of the form

$$\begin{aligned} \frac{\partial \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{p}_{i+1}}{\partial \mathbf{R}_{i+1}} &= \begin{pmatrix} \frac{\partial \hat{\mathbf{p}}'_1}{\partial \mathbf{r}_1} & \frac{\partial \hat{\mathbf{p}}'_1}{\partial \mathbf{r}_2} & \frac{\partial \hat{\mathbf{p}}'_1}{\partial \mathbf{r}_3} & \frac{\partial \hat{\mathbf{p}}'_1}{\partial \mathbf{r}_4} \\ \frac{\partial \hat{\mathbf{p}}'_2}{\partial \mathbf{r}_1} & \frac{\partial \hat{\mathbf{p}}'_2}{\partial \mathbf{r}_2} & \frac{\partial \hat{\mathbf{p}}'_2}{\partial \mathbf{r}_3} & \frac{\partial \hat{\mathbf{p}}'_2}{\partial \mathbf{r}_4} \\ \frac{\partial \hat{\mathbf{p}}'_3}{\partial \mathbf{r}_1} & \frac{\partial \hat{\mathbf{p}}'_3}{\partial \mathbf{r}_2} & \frac{\partial \hat{\mathbf{p}}'_3}{\partial \mathbf{r}_3} & \frac{\partial \hat{\mathbf{p}}'_3}{\partial \mathbf{r}_4} \end{pmatrix} \\ &= \frac{\partial \hat{\mathbf{p}}'_{i+1/i}}{\partial \mathbf{R}_{i+1}} = \end{aligned} \quad (82)$$

Now defining a 3×4 matrix $\mathcal{H}_{i+1/i} = -\frac{\partial \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{p}_{i+1}}{\partial \mathbf{R}_{i+1}}$ we can write the previous equation as

$$-\mathcal{H}_{i+1/i} \stackrel{\text{def}}{=} \frac{\partial \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{p}_{i+1}}{\partial \mathbf{R}_{i+1}} = \begin{pmatrix} h_1 & h_2 & h_3 & h_4 \\ h_4 & -h_3 & h_2 & -h_1 \\ -h_3 & -h_4 & h_1 & h_2 \end{pmatrix}, \quad (83)$$

where

$$\begin{aligned} h_1 &= 2(\hat{r}_1 p_1 + \hat{r}_4 p_2 - \hat{r}_3 p_3), \\ h_2 &= 2(\hat{r}_2 p_1 + \hat{r}_3 p_2 + \hat{r}_4 p_3), \\ h_3 &= 2(-\hat{r}_3 p_1 + \hat{r}_2 p_2 - \hat{r}_1 p_3), \\ h_4 &= 2(-\hat{r}_4 p_1 + \hat{r}_1 p_2 + \hat{r}_2 p_3). \end{aligned}$$

First replacing (83) in (77) and then substituting it together with (75) (76) in (74) and then omitting the second order terms we get

$$\begin{aligned} 0 &= \mathbf{p}'_{i+1} - \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{p}_{i+1} + (\mathbf{p}'_{0,i+1} - \mathbf{p}'_{i+1}) \\ &\quad - \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})(\mathbf{p}_{0,i+1} - \mathbf{p}_{i+1}) \\ &\quad - \mathcal{H}_{i+1/i}(\mathbf{R}_{i+1} - \hat{\mathbf{R}}_{i+1/i}). \end{aligned} \quad (84)$$

This equation can be further rearranged as

$$\begin{aligned} \mathbf{p}'_{i+1} - \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{p}_{i+1} + \mathcal{H}_{i+1/i}\hat{\mathbf{R}}_{i+1/i} \\ = \mathcal{H}_{i+1/i}\mathbf{R}_{i+1} + (\mathbf{p}'_{i+1} - \mathbf{p}'_{0,i+1}) \\ - \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})(\mathbf{p}_{i+1} - \mathbf{p}_{0,i+1}). \end{aligned} \quad (85)$$

The components of this equation are now amenable to be identified as the measurement \mathbf{z}_{i+1} and the noise of the measurement $\mathbf{n}_{i+1/i}$ as follows

$$\mathbf{z}_{i+1} = \mathbf{p}'_{i+1} - \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{p}_{i+1} + \mathcal{H}_{i+1/i}\hat{\mathbf{R}}_{i+1/i}, \quad (86)$$

$$\begin{aligned} \mathbf{n}_{i+1/i} &= (\mathbf{p}'_{i+1} - \mathbf{p}'_{0,i+1}) - \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})(\mathbf{p}_{i+1} - \mathbf{p}_{0,i+1}) \\ &= \mathbf{n}'_{i+1} - \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{n}_{i+1}. \end{aligned} \quad (87)$$

In terms of these variables we can finally write the first order linearized measurement equation more compactly

$$\mathbf{z}_{i+1} = \mathcal{H}_{i+1/i}\mathbf{R}_{i+1} + \mathbf{n}_{i+1/i}. \quad (88)$$

Here $\mathbf{n}_{i+1/i}$ is zero mean noise with covariance given by

$$\mathbf{C}_{i+1/i} = \mathbf{C}'_{i+1} + \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{C}_{i+1}\mathcal{R}^T(\hat{\mathbf{R}}_{i+1/i}). \quad (89)$$

where \mathbf{C}'_{i+1} and \mathbf{C}_{i+1} are the known covariance matrices of the noise \mathbf{n}'_{i+1} and \mathbf{n}_{i+1} respectively.

5.3.1. Rotation Estimation. Now we will describe the procedure of the estimation of a rotation expressed as a the rotor \mathbf{R} . At the beginning of the iteration or step $i = 0$, the initial state $\hat{\mathbf{R}}_0$ and an initial estimation error covariance matrix \mathcal{P}_0 are given. According to the Eq. (67), given an estimate $\hat{\mathbf{R}}_i^*$, and the covariance matrix \mathcal{P}_i at step i , it is reasonable to assign the following as a prediction of the next step $i + 1$

$$\hat{\mathbf{R}}_{i+1/i} = \hat{\mathbf{R}}_i^*, \quad (90)$$

$$\hat{\mathcal{P}}_{i+1/i} = \mathcal{P}_i. \quad (91)$$

Taking into account the measurements $\mathbf{p}_{i+1}, \mathbf{p}'_{i+1}$ and the predicted state $\hat{\mathbf{R}}_{i+1/i}$ the new measurement \mathbf{z}_{i+1} , Eq. (86) can be straightforwardly computed. Then, according Eqs. (55) and (58) the Kalman gain matrix and the estimate $\hat{\mathbf{R}}_{i+1}$ at step $i + 1$ by EKF are computed respectively as follows

$$\begin{aligned} \mathcal{K}_{i+1} &= \mathcal{P}_{i+1/i}\mathcal{H}_{i+1/i}^T \\ &\quad (\mathcal{H}_{i+1/i}\mathcal{P}_{i+1/i}\mathcal{H}_{i+1/i} + \mathbf{C}_{i+1/i})^{-1}, \end{aligned} \quad (92)$$

$$\begin{aligned} \hat{\mathbf{R}}_{i+1} &= \hat{\mathbf{R}}_{i+1/i} + \mathcal{K}_{i+1}(\mathbf{z}_{i+1} - \mathcal{H}_{i+1/i}\hat{\mathbf{R}}_{i+1/i}) \\ &= \hat{\mathbf{R}}_{i+1/i} + \mathcal{K}_{i+1}(\mathbf{p}'_{i+1} - \mathcal{R}(\hat{\mathbf{R}}_{i+1/i})\mathbf{p}_{i+1}). \end{aligned} \quad (93)$$

$\hat{\mathbf{R}}_{i+1}$ must be modified prudently to satisfy the constraint $\|\mathbf{R}\| = 1$:

$$\hat{\mathbf{R}}_{i+1}^* = \frac{\hat{\mathbf{R}}_{i+1}}{\|\hat{\mathbf{R}}_{i+1}\|} \quad (94)$$

The updating of the error covariance matrix follows according to

$$\begin{aligned} \mathcal{P}_{i+1} &= (\mathbf{I} - \mathcal{K}_{i+1}\mathcal{H}_{i+1})\mathcal{P}_{i+1/i}(\mathbf{I} - \mathcal{K}_{i+1}\mathcal{H}_{i+1})^T \\ &\quad + \mathcal{K}_{i+1}\mathbf{C}_{i+1}\mathcal{K}_{i+1}^T. \end{aligned} \quad (95)$$

Note that \mathcal{H}_{i+1} and \mathbf{C}_{i+1} are recalculated using the current estimate $\hat{\mathbf{R}}_{i+1}^*$. As far as the implementation of the filter is concerned, the measurement error covariance matrix \mathbf{C}_i might be measured prior to the operation of the filter.

6. The Motor Extended Kalman Filter

The use of the EKF filter in the motor algebra framework gives us the simultaneous estimation of translation and rotation. According to the literature there are only batch methods for the simultaneous estimation of these components [6, 15]. The motor extended Kalman filter (MEKF) turns out to be a natural extension of the rotor extended Kalman filter thanks to the multivector concept of the geometric algebra. First, let us define the noisy motion equation using lines in the motor algebra framework $\mathcal{G}_{1,3,0}^+$.

The geometric features we will consider for the measurements are 3D lines $(L^1, L^2, \dots, L^n, n \geq 2)$ which belong to an object moving in the 3D space, see Fig. 1. The rigid motion parameters between any couple of consecutive time instants $(t_0, t_1, t_2, \dots, t_N)$ are described compactly by the motor M_i . According to the Eq. (36), the motion of any line of the object is modeled by

$$L_i = M_i L_{i-1} \tilde{M}_i. \quad (96)$$

If the change of the parameters of the line in motion between the time instants t_{i-1} and t_i is described in terms of the motor velocity information $V_{i/i-1}$

$$L_i = V_{i/i-1} L_{i-1} \tilde{V}_{i/i-1}, \quad (97)$$

we can then express the recursive motion equation of the line in general as follows

$$\begin{aligned} L_i &= M_i L_{i-1} \tilde{M}_i \\ &= (V_{i/i-1} M_{i-1}) L_{i-1} (\tilde{M}_{i-1} \tilde{V}_{i/i-1}). \end{aligned} \quad (98)$$

Thus, we obtain the ideal dynamic motion model in terms of the motors

$$M_i = V_{i/i-1} M_{i-1}. \quad (99)$$

For example, suppose the motion is a screw motion with rotation of constant angular velocity ω about an axis of known line $(L_s = \bar{r} + I t_c \wedge \bar{r})$ and with constant translation velocity v_s along the axis line. If the data sampling is done at equidistant time interval, then the instant times can be represented by integers, so the motor equation reads

$$\begin{aligned} V_{i/i-1} &= V \\ &= (1 + I v_s / 2)(\cos(\omega/2) + \sin(\omega/2) L_s). \end{aligned} \quad (100)$$

Since in real work the relation between M_{i-1} and M_i is known only approximately, the real dynamic model of the noisy 3D motion is given by

$$M_i = \mathcal{V}_{i/i-1, Ml} M_{i-1} + W_i, \quad (101)$$

where the statistics of W_i is given by (49) and (50). Note that $\mathcal{V}_{i/i-1, Ml}$ means "left-multiplication matrix" of motor $\mathcal{V}_{i/i-1}$.

6.1. Linearization of the Measurement Model

Considering Eq. (45), we can easily see that the relation between the measurement \mathcal{A}_M and the state M is unfortunately nonlinear; that is why we should linearize it. Assume that the measurement \mathcal{A}_{Mi} is the true data $\mathcal{A}_{M0,i}$ contaminated by measurement noise $\mathcal{N}_{\mathcal{A}_M,i}$ with zero mean and known covariance matrix $\mathcal{C}_{\mathcal{A}_M,i}$

$$\mathcal{A}_{Mi} = \mathcal{A}_{M0,i} + \mathcal{N}_{\mathcal{A}_M,i}. \quad (102)$$

Supposing that the predicted state of M_i is $\hat{M}_{i/i}$ than according the Eq. (74) we can define a function $f_{M,i}$ depending of the variables $(\mathcal{A}_{M0,i}, M_i)$ as follows

$$f_{M,i}(\mathcal{A}_{M0,i}, M_i) = \mathcal{A}_{M0,i} M_i = 0. \quad (103)$$

This equation can be expanded into a first order Taylor series about the predicted state $(\mathcal{A}_{Mi}, \hat{M}_{i/i-1})$

$$\begin{aligned} f_{M,i}(\mathcal{A}_{M0,i}, M_i) &= f_{M,i}(\mathcal{A}_{Mi}, \hat{M}_{i/i-1}) \\ &+ \frac{\partial f_{M,i}(\mathcal{A}_{Mi}, \hat{M}_{i/i-1})}{\partial M_i} (M_i - \hat{M}_{i/i-1}) \\ &+ (\mathcal{A}_{M0,i} - \mathcal{A}_{Mi}) \frac{\partial f_{M,i}(\mathcal{A}_{Mi}, \hat{M}_{i/i-1})}{\partial \mathcal{A}_{M0,i}} \\ &+ \mathcal{O}^2 = 0. \end{aligned} \quad (104)$$

Now calling the components

$$\frac{\partial f_{M,i}(\mathcal{A}_{Mi}, \hat{M}_{i/i-1})}{\partial M_i} = \mathcal{A}_{Mi}, \quad (105)$$

$$\frac{\partial f_{M,i}(\mathcal{A}_{Mi}, \hat{M}_{i/i-1})}{\partial \mathcal{A}_{M0,i}} = \hat{M}_{i/i-1}, \quad (106)$$

omitting the second order terms \mathcal{O}^2 and considering the Eq. (102) for $\mathcal{A}_{M,i+1}$ we get

$$\begin{aligned} & \mathcal{A}_{M,i} \hat{M}_{i/i-1} + \mathcal{A}_{M,i} (M_i - \hat{M}_{i/i-1}) \\ & + (\mathcal{A}_{M0,i} - \mathcal{A}_{M,i}) \hat{M}_{i/i-1} = \mathcal{A}_{M,i} \hat{M}_{i/i-1} \\ & + \mathcal{A}_{M,i} (M_i - \hat{M}_{i/i-1}) - \mathcal{N}_{\mathcal{A}_{M,i+1}} \hat{M}_{i/i-1} = 0. \end{aligned} \quad (107)$$

or

$$\begin{aligned} & -\mathcal{A}_{M,i} M_i + \mathcal{N}_{\mathcal{A}_{M,i}} \hat{M}_{i/i-1} \\ & = \mathcal{A}_{M,i} \hat{M}_{i/i-1} - \mathcal{A}_{M,i} \hat{M}_{i/i-1} = 0 \end{aligned} \quad (108)$$

As a result we can claim that the measurement equation for the MEKF at step i is given by

$$\begin{aligned} Z_i &= -\mathcal{A}_{M,i} M_i + \mathcal{N}_{\mathcal{A}_{M,i}} \hat{M}_{i/i-1} \\ &= \mathcal{H}_i M_i + N_{Z,i} = 0, \end{aligned} \quad (109)$$

where we call $\mathcal{H}_i = -\mathcal{A}_{M,i}$ and $N_{Z,i} = \mathcal{N}_{\mathcal{A}_{M,i+1}} \hat{M}_{i/i-1}$. The covariance matrix of $N_{Z,i}$ is \mathcal{C}_i .

6.2. Enforcing a Geometric Constraint

In order to estimate the motor state, we assume first that at the beginning or step $i = 0$ the initial state $\hat{M}_{1/0}$ and the initial error covariance matrix of the estimate $\mathcal{P}_{1/0}$ are given. Now, according to Eqs. ((55), (59), (58) and (61)), the estimation equation of the motor state is given by

$$\begin{aligned} M_i^* &= \Phi_{i/i-1} \hat{M}_{i-1} + \mathcal{K}_i (Z_i - \mathcal{H}_i \Phi_{i/i-1} \hat{M}_{i-1}) \\ &= \mathcal{V}_{i/i-1, M} \hat{M}_{i-1} + \mathcal{K}_i (Z_i - \mathcal{H}_i \mathcal{V}_{i/i-1, M} \hat{M}_{i-1}) \\ &= (R_i^{*T} \quad R_i'^{*T})^T, \end{aligned} \quad (110)$$

where the optimal Kalman gain matrix \mathcal{K}_i is computed according to the formula

$$\mathcal{K}_i = \mathcal{P}_{i/i-1} \mathcal{H}_i^T (\mathcal{H}_i \mathcal{P}_{i/i-1} \mathcal{H}_i^T + \mathcal{C}_i)^{-1}. \quad (111)$$

where

$$\mathcal{P}_{i/i-1} = \Phi_{i/i-1} \mathcal{P}_i \Phi_{i/i-1}^T + \mathcal{Q}_i \quad (112)$$

and the error covariance matrix for the i -step is updated as

$$\mathcal{P}_i = \mathcal{P}_{i/i-1} - \mathcal{K}_i \mathcal{H}_i \mathcal{P}_{i/i-1} = (I - \mathcal{K}_i \mathcal{H}_i) \mathcal{P}_{i/i-1}. \quad (113)$$

Now M_i^* consists of two 4-dimensional vectors R_i^* and $R_i'^*$ which must be varied to fulfill the constraints (46) and (47). In the case of the constraint (46), the modifications can be done simply by

$$\hat{R}_i = \frac{R_i'^*}{\|R_i'^*\|}. \quad (114)$$

However, it is not that simple to satisfy the constraint (47). The constraint

$$R'^T R = 0 \quad (115)$$

tells us that R must be orthogonal to the dual rotor R' and this is valid up to a scalar, see the role of this scalar in the Eq. (120) below. Unfortunately in the practice the rotor estimate R_i^* is usually not orthogonal to the estimated dual rotor $R_i'^*$. The Fig. 6 suggests clearly how to enforce this geometric constraint in order to modify the orientation of $R_i'^*$.

In order to do so first we consider the cosine of the angle φ between estimates R_i^* and $R_i'^*$

$$\cos(\varphi) = \frac{R_i'^{*T} R_i^*}{\|R_i'^*\| \cdot \|R_i^*\|} \quad (116)$$

which can be simplified by using Eq. (114) and the unit rotor \hat{R}_i as follows

$$\cos(\varphi) = \frac{R_i'^{*T} \hat{R}_i}{\|R_i'^*\|}. \quad (117)$$

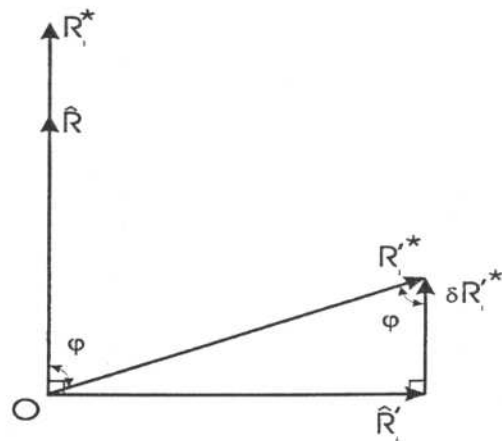


Figure 6. Constraint of $\hat{R} \hat{R}' = 0$.

Then we consider also the orthogonal variation

$$\delta \hat{R}_i^* = \|\hat{R}_i^*\| \cos(\varphi) \hat{R}_i = (R_i'^{*T} \hat{R}_i) \hat{R}_i. \quad (118)$$

It is then straightforward to build the following difference relation

$$\hat{R}_i' = k(R_i'^* - (R_i'^{*T} \hat{R}_i) \hat{R}_i). \quad (119)$$

Note that this equation is valid up to a scalar k which can be defined as

$$k = \frac{\alpha \|\hat{R}_i'\|}{\|R_i'^* - (R_i'^{*T} \hat{R}_i) \hat{R}_i\| \cdot \|\hat{R}_i^*\|}, \quad (120)$$

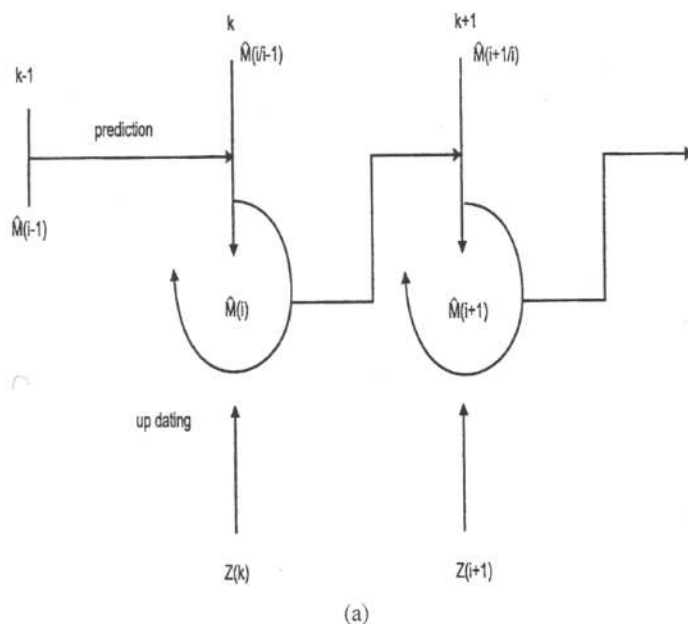
therefore

$$\hat{R}_i' = \alpha \frac{(R_i'^* - (R_i'^{*T} \hat{R}_i) \hat{R}_i) \|\hat{R}_i'\|}{\|R_i'^* - (R_i'^{*T} \hat{R}_i) \hat{R}_i\| \|\hat{R}_i^*\|} \quad (121)$$

where the scalar α can be find out thereafter by filter tuning. An illustration of the effect of the tuning will be presented in the experimental part.

6.3. Operation of the MEKF Algorithm

The processing of information by the MEKF filter can be explained very simply by considering the block diagrams presented in Fig. 7(a) and (b).



In general, during the MEKF cycle of Fig. 7(a), the updating uses a prediction and a corrected input measurement to actualize a new estimate which in turn will be modified using a geometric constraint. This cycle continues for each new measurement on and on. After few iterations the MEKF should have stabilized to the estimated proper states.

The Kalman gain K_i can be calculated before the actual estimation is carried out and it does not depend at all on the measurement Z_i . The computation cycle for K_i illustrated in Fig. 7(b) would proceed as follows

Step	Procedure
1	given $\mathcal{P}_{i-1/i-1} = \mathcal{P}_i$, \mathcal{Q}_i and $\Phi_{i/i-1}$ then $\mathcal{P}_{i/i-1}$ is computed using the Eq. (112).
2	$\mathcal{P}_{i/i-1}$, \mathcal{H}_i and \mathcal{C}_i are substituted in Eq. (111) to obtain K_i , which will be used in step 3 of the MEKF algorithm.
3	$\mathcal{P}_{i/i-1}$, K_i , and \mathcal{H}_i are substituted in Eq. (113) to determine \mathcal{P}_i , which is stored until the time of the next measurement, when the cycle is repeated

Next the MEKF algorithm illustrated in Fig. 8 will be explained. For the initialization of the MEKF we can use for the initial time instant i known values of \hat{M}_{i-1} and \mathcal{P}_{i-1} or if we do not known them simply take the trivial values $\hat{M}_{i-1} = [10000000]^T$ and $\mathcal{P}_{i-1} = I_{8 \times 8}$. After the initialization the MEKF seeks to determine \hat{M}_i for some future time instant. The computation procedure of the MEKF would proceed as follows

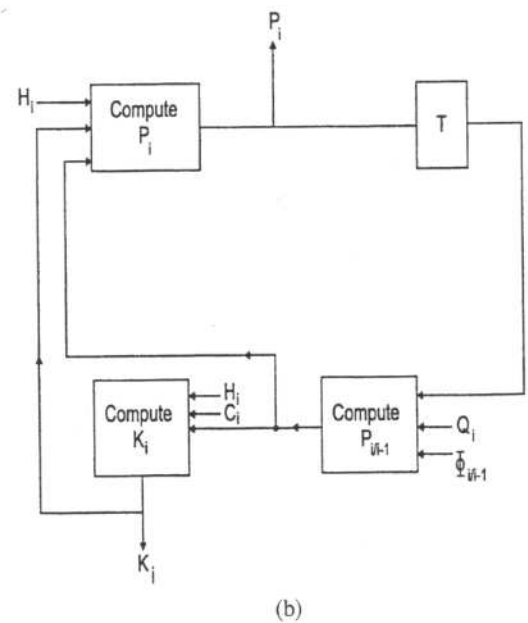


Figure 7. (a) Estimation and up-dating cycles (b) Kalman gain computation.

Step	Action	Procedure
1	Prediction	The estimated \hat{M}_{i-1} is propagated forward by premultiplying it by the discrete system model matrix $\Phi_{i/i-1}$. This gives the predicted estimate $\hat{M}_{i/i-1} = \Phi_{i/i-1} \hat{M}_{i-1}$. Then the measurement model $\mathcal{H}_i = -\mathcal{A}_{M_i}$ is linearized, see Eq. (45).
2	Estimation	$\Phi_{i/i-1} \hat{M}_{i/i-1}$ is premultiplied by $\mathcal{H}_i = -\mathcal{A}_{M_i}$ giving the estimated input measurement $\hat{Z}_i = \mathcal{H}_i \Phi_{i/i-1} \hat{M}_{i/i-1}$ which is subtracted from actual measurement Z_i to obtain the measured residual or error $e'_i = Z_i - \mathcal{H}_i \Phi_{i/i-1} \hat{M}_{i/i-1}$.
3	Correction	The error e'_i is premultiplied by the matrix \mathcal{K}_i and the result is added to $\hat{M}_{i/i-1}$ to give the current estimate M_i^* , see Eq. (110).
4	Modification	The component R_i^* of the estimate $M_i^* = (R_i^{*T}, R_i'^{*T})^T$ is modified using a geometric constrain according the Eq. (121). Then the final estimate $\hat{M}_i = (\hat{R}_i^T, \hat{R}_i'^T)^T$ is stored until the next measurement is made, at which the cycle is repeated.

The MEKF would run recursively completing these cycles until instant time N . It is necessary to mention that Kalman filter implementation is sensitive to the numerical instability. In order to overcome these problems several techniques are available as the square-root filtering and the so called U-D factorization [19].

7. Analytical Method for the Recovering of Motion Parameters

In the experimental part we will compare the performance of the MEKF algorithm with an analytical method used by Zhang and Faugeras [22]. In this section, we explain briefly this method. This procedure can also be advantageously used for estimating the initial prediction in our MEKF algorithm.

Consider $k = 1, 2, \dots, n$ lines belonging to a rigid object. The lines $L_0^k = n_0^k + Im_0^k$ are measured before the rigid motion transformation given by

$$M_i = \left(1 + I \frac{t_i}{2}\right) R_i \quad (122)$$

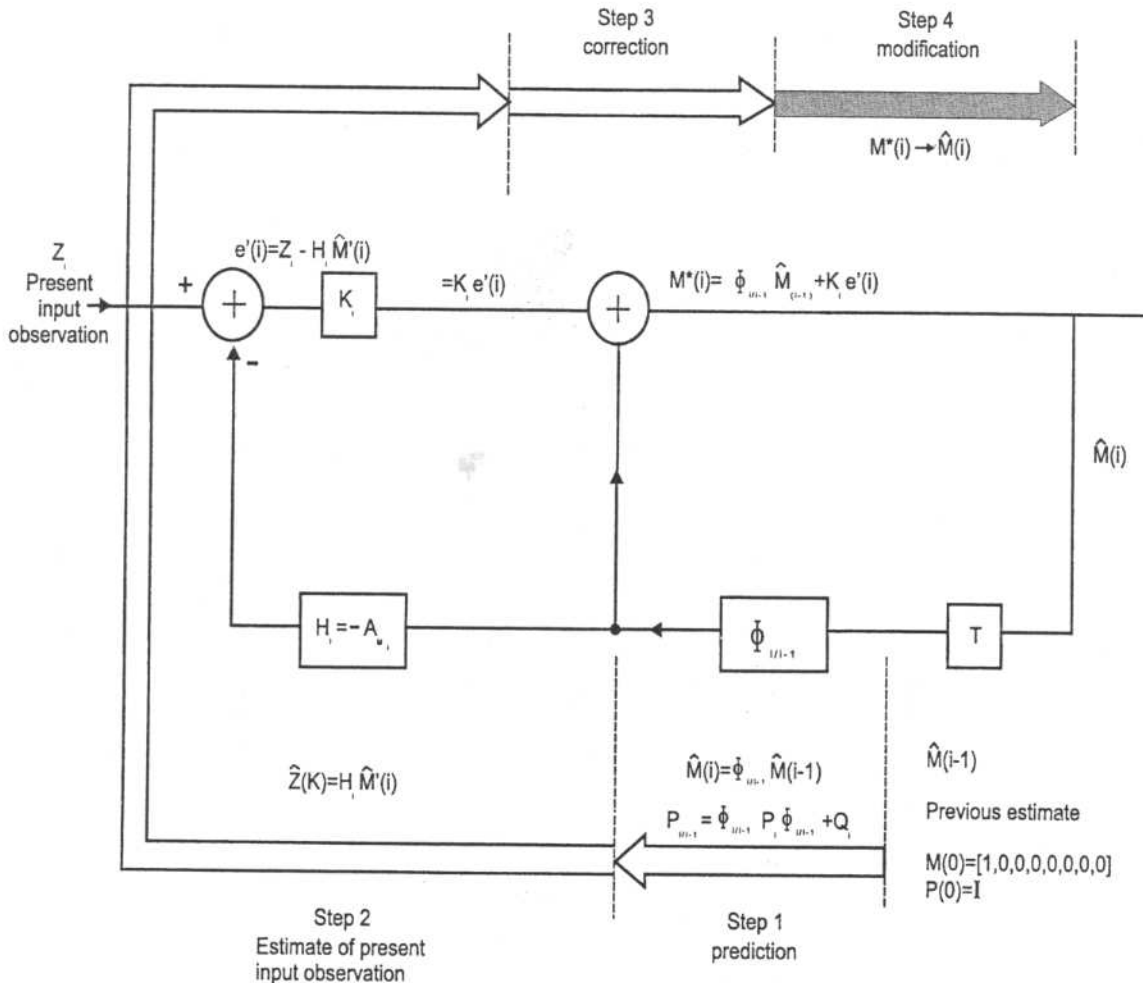


Figure 8. Representation of the MEKF algorithm.

and the lines $L_i^k = n_i^k + Im_i^k$ are measured after. According to the Eq. (34) the relation between a line $L_0 = L_0^k$ and the transformed line $L_i = L_i^k$ is given by

$$L_i = n_i + Im_i \\ = R_i n_0 \tilde{R}_i + I(R_i m_0 \tilde{R}_i + t_i \wedge (R_i n_0 \tilde{R}_i)). \quad (123)$$

The real and dual parts of this equation are

$$n_i = R_i n_0 \tilde{R}_i \quad (124)$$

$$m_i = R_i m_0 \tilde{R}_i + t_i \wedge n_i. \quad (125)$$

The method uses the noisy measurements L_0^k and L_i^k in the least square sense to estimate the best solution for the rotation and translation components. The first criterion of minimization uses the rotation of Eq. (124)

$$E_{\min} = \left\{ \sum_{k=1}^n \|n_i^k - R_i n_0^k \tilde{R}_i\|^2 \right\}_{\min}. \quad (126)$$

Multiplying from the right both sides of (124) with the rotor R_i results

$$n_i R_i - R_i n_0 = 0. \quad (127)$$

This expression in terms of linear algebra reads

$$(n_i)_{Ri} R_i - (n_0)_{Rr} R_i = \mathcal{A}_R R_i = 0. \quad (128)$$

Thus Eq. (126) can be reformulated as follows

$$E'_{\min} = \left\{ \sum_{k=1}^n R_i^T \mathcal{A}_R^{kT} \mathcal{A}_R^k R_i \right\}_{\min} = \{R_i^T \mathcal{A}_R\}_{\min}, \quad (129)$$

where

$$\mathcal{A}_R^k = \sum_{k=1}^n ((n_i^k)_{Ri} - (n_0^k)_{Rr}), \quad (130)$$

$$\mathcal{A} = \sum_{k=1}^n \mathcal{A}_R^{kT} \mathcal{A}_R^k. \quad (131)$$

Since \mathcal{A} is a symmetric matrix and $\|R_i\| = 1$, the solution to this problem is given by the smallest eigenvalue of \mathcal{A} which corresponds to the 4-dimensional vector \hat{R}_i .

Using the computed rotation \hat{R}_i the translation is straightforwardly computed with Eq. (125). In terms of linear algebra, Eq. (125) can be expressed as

$$m_i = \mathcal{R}_i m_0 - (n_i)_{\times} t_i, \quad (132)$$

where

$$\mathcal{R}_i = (R_i)_{Ri} (\tilde{R}_i)_{Rr}, \quad (133)$$

and the matrix $(n_i)_{\times}$ is the skew-symmetric matrix of n_i

$$(n_i)_{\times} = \begin{pmatrix} 0 & n_{3,i} & -n_{2,i} \\ -n_{3,i} & 0 & n_{1,i} \\ n_{2,i} & -n_{1,i} & 0 \end{pmatrix}, \quad (134)$$

which performs the outer product of the bivector n_i with another bivector. The estimate of the translation \hat{t}_i is obtained minimizing the following criterion

$$E''_{\min} = \left\{ \sum_{k=1}^n \|m_i^k - \hat{\mathcal{R}}_i m_0^k + (n_i^k)_{\times} \hat{t}_i\|^2 \right\}_{\min}. \quad (135)$$

After differentiating this criterion with respect to t_0 and setting the result equal to zero, we obtain

$$\sum_{k=1}^n 2(m_i^k - \hat{\mathcal{R}}_i m_0^k + (n_i^k)_{\times} \hat{t}_i)^T (n_i^k)_{\times} = 0. \quad (136)$$

Finally the desired \hat{t}_i can be gained solving the following equation

$$\left(\sum_{k=1}^n (n_i^k)_{\times}^T (n_i^k)_{\times} \right) \hat{t}_i = \sum_{k=1}^n (n_i^k)_{\times}^T (\hat{\mathcal{R}}_i m_0^k - m_i^k). \quad (137)$$

It is well known that at least two non-parallel lines are required to determine a unique motion displacement. This tells that the matrices \mathcal{A} and $\mathcal{B} = \sum_{k=1}^n (n_i^k)_{\times} (n_i^k)_{\times}^T$ are always of full rank if two of the lines $L_i^k (k = 1, \dots, n)$ are non-parallel.

8. Experimental Results

This section presents first a simulation to compute the motion parameters of an object moving in a screw trajectory. In this simulation the MEKF algorithm and the analytical solution are compared. The second experiment involves the use of the MEKF for the estimation of the motion parameters of a line moving in 3D space. The line measurements were taken using a stereo system attached to a 6 DOF robot arm.

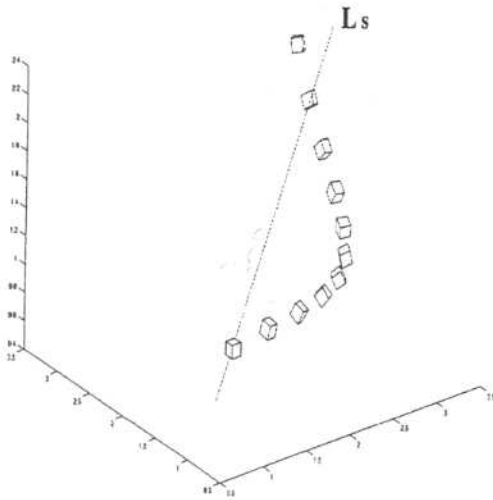


Figure 9. The object moving in 3-D with screw trajectory.

8.1. Comparison of the MEKF and Analytical Solution Algorithms

This simulation was written using MATLAB to compare both methods. The methods estimate the motion parameters of a moving object in the 3D space.

The Fig. 9 shows a rigid object moving with screw motion about the known line axis

$$\begin{aligned} L_s &= \bar{r} + It_c \wedge \bar{r} \\ &= 0.7071\gamma_2\gamma_3 + 0.3536\gamma_3\gamma_1 + 0.6124\gamma_1\gamma_2 \\ &\quad + I(-0.7418\gamma_2\gamma_3 + 0.3813\gamma_3\gamma_1 + 0.6364\gamma_1\gamma_2). \end{aligned}$$

with angular velocity $w/2 = -\phi/15$ and a constant velocity $v_s = 0.3\bar{r}$. Since the measurements are sampled at equidistant intervals, they are normalized to 1. The motion between the instant times $i-1$ to i was represented using the Eq. (100) of the motor

$$\begin{aligned} V &= (1 + Iv_{s0}/2)(\cos(\omega/2) + \sin(\omega/2)L_s) \\ &= (1 + I0.3\bar{r}/2)(\cos(-\pi/15) + \sin(-\pi/15)L_s) \\ &= 0.9832 - 0.1289\gamma_2\gamma_3 - 0.0645\gamma_3\gamma_1 \\ &\quad - 0.1117\gamma_1\gamma_2 + I(-0.0266 + 0.2367\gamma_2\gamma_3 \\ &\quad - 0.0188\gamma_3\gamma_1 - 0.0283\gamma_1\gamma_2). \end{aligned} \quad (138)$$

In the simulation the variables of the used motor

$$\begin{aligned} V_{i/i-1} &= (1 + Iv_{si}/2)(\cos(\omega_i/2) + \sin(\omega_i/2)L_s) \\ \omega_i &= \omega + n_{\omega_i}, \\ v_{si} &= v_s + n_{v_{si}}, \end{aligned} \quad (139)$$

were corrupted using noise variables n_{ω_i} and $n_{v_{si}}$ which are independent, normally distributed with zero mean and known deviations σ_ω and σ_{v_s} .

We use the Eq. (99) of the ideal dynamic motion equation of a motor expressed in terms of linear algebra

$$M_i = V_{Ml}M_{i-1}, \quad (140)$$

in this sense the ground truth of the motor trajectories $M_{0,i}$ can be computed as follows

$$\begin{aligned} M_{0,i} &= (r_{0i} \ r_{1i} \ r_{2i} \ r_{3i} \ r'_{0i} \ r'_{1i} \ r'_{2i} \ r'_{3i})^T \\ &= V_{i/i-1}M_{0,i-1}, \end{aligned} \quad (141)$$

using as initial state $M_{0,0} = (1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$.

At the i -th instant of time in the MEKF algorithm two 3D points x'_0 and y'_0 define the line L'_0 at the coordinate frame A. After the application of the motor $M_i = (V_{i/i-1})_{Ml}M_{i-1}$ this line relative to the frame A is L'_i . Note that the real part of M_i gives the rotation R_i and from the dual part R'_i using the Eq. (20) we can gain the expected translation $t_i = 2R'_i\tilde{R}_i$.

Now having two lines related to the frame A: at the initial position L'_0 and at the position at time i the line L'_i , we can simulate the noisy line observations simply adding to those lines independent Gaussian noise with zero mean and known standard deviation σ , obtaining the noisy line observations L_0 and L_i . Using these noisy line observations both methods estimated the motion parameters with different degree of accuracy.

The MEKF algorithm was tuned according to Eq. (121) in order to improve its accuracy and the convergence rate of the estimate. The Fig. 10 shows the eight components of the motor parameters for a motion trajectory of 20 time instants. The initial prediction of the MEKF was estimated using the analytical solution. This type of initial prediction (using the analytical method) works very well, because in all eight parameters the estimations align themselves quickly to the ground truth. We can clearly see that the MEKF algorithm follows the ground truth remarkably better than the analytical method.

8.2. Estimation of the Relative Positioning of an Robot End-Effector

In this experiment we applied the MEKF algorithm to estimate the relative motion between the end-joint of a Staubli RX90 robot arm and 3D reference line belonging to a rigid object. The 3D line parameters were recovered during the arm movement using a stereo

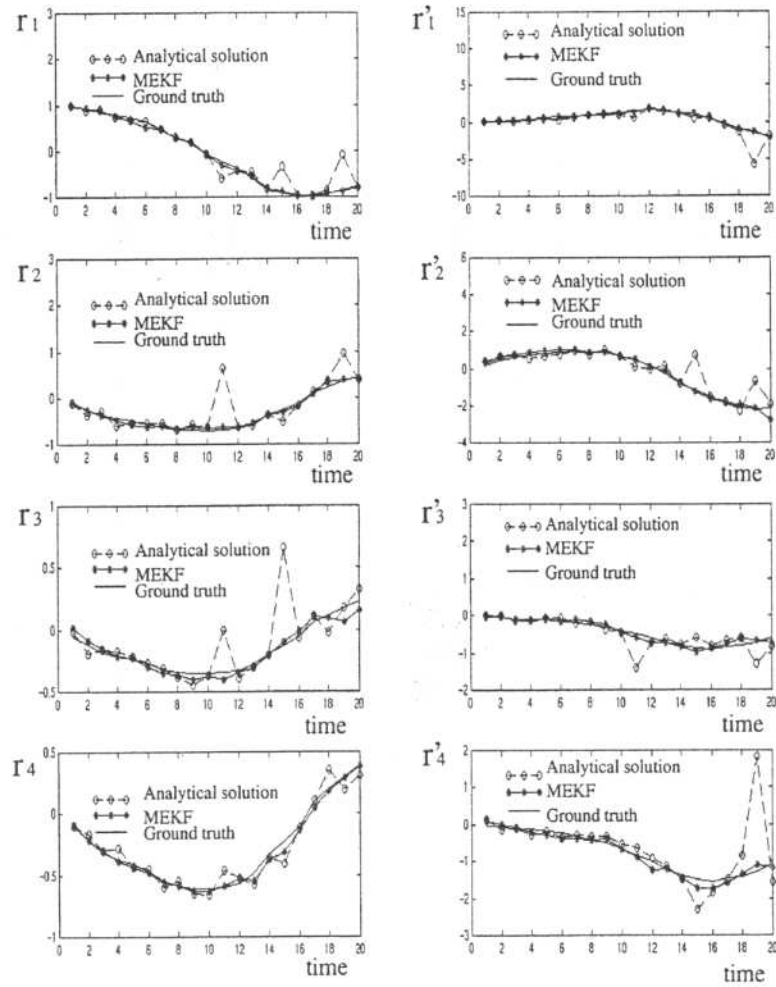


Figure 10. The estimation of the motor parameters for the simulated object motion.

vision system. This approach can be used for several industrial applications like maneuvering and grasping.

The physical setup of the experiment is shown in Fig. 11. The system looks at a couple of lines lying on

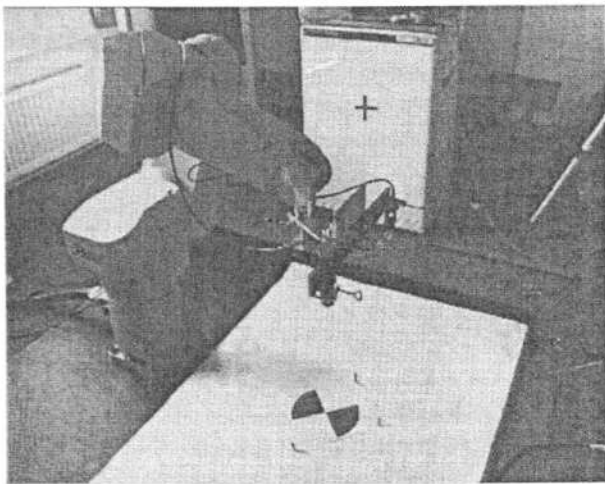


Figure 11. The physical setup of the experiment.

the floor and moves in the 3D space always conserving those lines in field of view. The main task for the system is to estimate automatically the relative motion between the floor lines and its end-joint. The visual system consists of two grey-scale CCD 640×480 cameras fastened to the last joint of the robot arm. The Staubli robot arm has six joints which can be controlled by six variables (x , y , z , roll, pitch and yaw). The coordinates (x , y , z) describe the position of the tool coordinate system T of the end-joint, referred to the coordinate system W of the robot. The orientation of the end-joint is described by the variables (roll, pitch, yaw) in terms of Euler angles. Since in practice we need three views in order to reconstruct a 3D line, using a two cameras stereo vision system we can create a virtual third one simply moving a bit the stereo system as it is illustrated in Fig. 12. The movement of the robot arm is carried out and controlled by the relative position and orientation between the tool coordinate system T and the base system W . The camera calibration procedure obtains

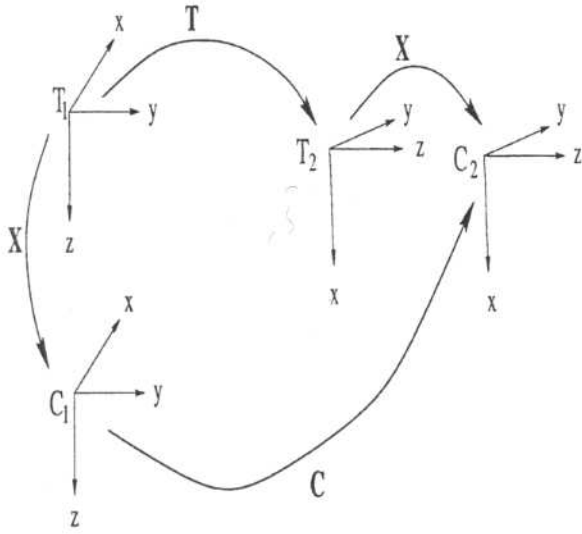


Figure 12. The relationship between the tool system T and the camera system C .

the projective transformation matrix \mathbf{P} which relates up to scalar the visual 3D space to the image plane. The coordinate system of the camera C at the end-joint is related to the tool system frame T via a certain transformation \mathbf{X} which was computed using a hand-eye calibration procedure [18]. When the tool system T is transformed from T_1 to T_2 by the transformation \mathbf{T} , the camera system C will be transformed from C_1 to C_2 by a certain transformation \mathbf{C}

$$\mathbf{C} = \mathbf{X}\mathbf{T}\mathbf{X}^{-1}. \quad (142)$$

Since the motion of the robot arm specified by the transformation \mathbf{T} is known, we can compare it with the relative motion \mathbf{C} . The relative motion of the frame W and the frame C is a screw motion with constant angular velocity $\omega = -\pi/90$ and a constant translation velocity $v_s = 0.2$. The axis line L_s is parallel to the z axis of the system C and the point $(1.5, 0, 0)$ touches these axis line. In the motor algebra $\mathcal{G}_{3,0,1}^+$ the axis line is given by

$$\begin{aligned} L_s &= \gamma_1\gamma_2 + I(1.5\gamma_2\gamma_3) \wedge (\gamma_1\gamma_2) \\ &= \gamma_1\gamma_2 + I1.5\gamma_3\gamma_1. \end{aligned} \quad (143)$$

Similarly as in the Eq. (138) the motor \mathbf{V} is calculated as follows

$$\begin{aligned} \mathbf{V} &= (1 + Iv_{s0}/2)(\cos(\omega/2) + \sin(\omega/2)L_s) \\ &= 0.9994 - 0.0349\gamma_1\gamma_2 \\ &\quad + I(0.0035 - 0.0523\gamma_3\gamma_1 + 0.0999\gamma_1\gamma_2). \end{aligned} \quad (144)$$

Table 1. Reconstructed 3-D lines.

Time	Line	A point on the line			Direction		
0	1	(0.000	3.087	-2.327)	(-0.345	0.937	-0.027)
	2	(0.556	0.000	-2.250)	(0.941	0.336	0.023)
1	1	(1.125	0.000	-2.027)	(-0.404	0.914	0.013)
	2	(0.701	0.000	-2.049)	(0.915	0.401	0.029)
2	1	(1.111	0.000	-1.82)	(-0.462	0.886	0.017)
	2	(0.794	0.000	-1.83)	(0.880	0.471	0.055)
⋮							
14	1	(0.018	0.000	0.648)	(-0.971	0.236	-0.036)
	2	(1.103	0.000	0.538)	(0.241	0.965	0.103)
15	1	(-0.680	0.000	0.753)	(-0.986	0.159	-0.025)
	2	(0.000	-6.341	0.783)	(0.171	0.985	-0.003)

The motor \mathbf{M}_{i+1} expressed in terms of linear algebra is given by

$$\mathbf{M}_i = \mathbf{V}_{M_i}\mathbf{M}_{i-1}, \quad (145)$$

initialized with $\mathbf{M}_0 = (1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)^T$. The reconstructed 3D lines listed in Table 1 were used to estimate the relative motion between the end-joint and the object on the floor using the MEKF algorithm.

The procedure followed in this experiment is summarized below.

Step	Procedure
1	Cameras calibration to obtain \mathbf{P} for each camera.
2	Hand-eye calibration to obtain \mathbf{X} for each camera.
3	Robot arm movement images taken at constant sample rate, see Figs. 13 and 14.
4	Extraction of 2D lines from the images using the Hough transform, see Figs. 13 and 14.
5	3D line reconstruction using matched lines of three images, see Table 1.
6	Estimation of the motion using the 3D lines observations and the MEKF algorithm, see Fig. 15

The algorithm for the motion estimation runs online recursively following the steps 3 to 6.

The Fig. 15 presents the eight estimated parameters of the motion for 15 instant times. We can see clearly that after almost 4 observations the MEKF algorithm starts to follow almost perfectly the ground truth of the eight parameters. This real experiment as well as the previous simulation confirm that the MEKF algo-

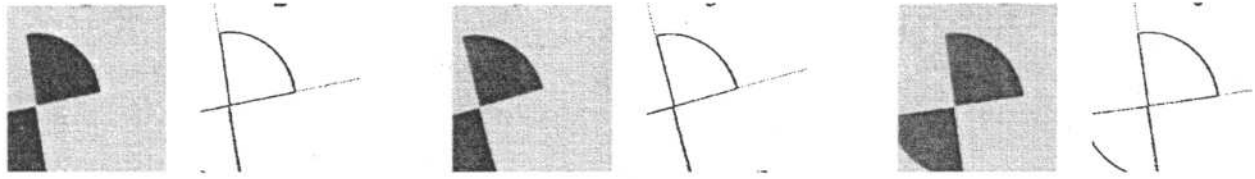


Figure 13. A stereo triplet of a sample object at time $i = 0$ with its edge images overlapped by extracted 2-D lines.

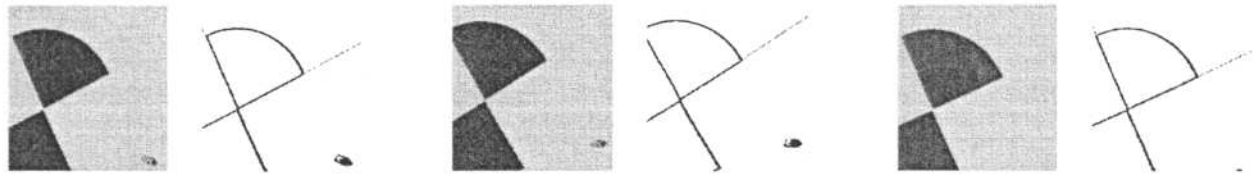


Figure 14. A stereo triplet of a sample object at time $i = 4$ with its edge images overlapped by extracted 2-D lines.

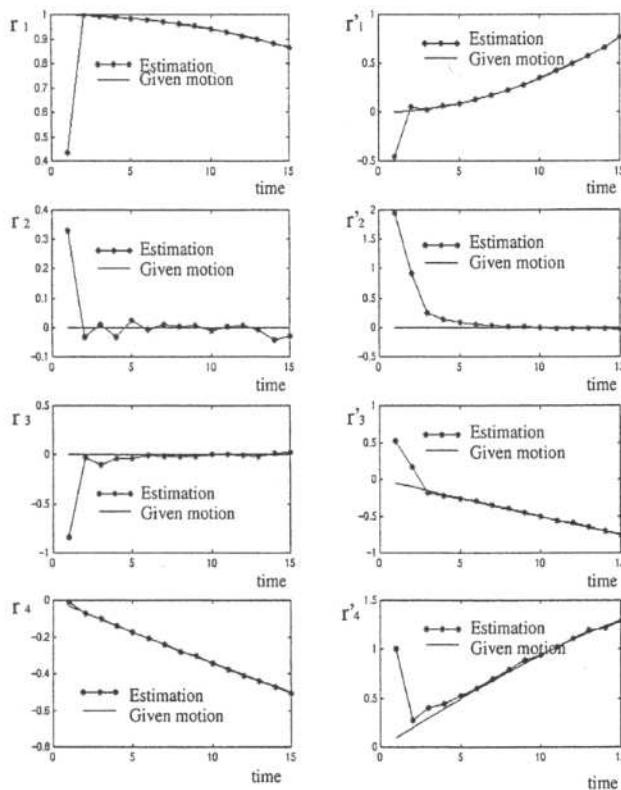


Figure 15. The estimated motor parameters by the MEKF for the visual guided robot system.

rithm is an appropriate tool for the estimation of screw transformations using line observations.

9. Conclusion

In this paper we modeled the motion of lines in the 4D space using the motor algebra. This kind of modeling linearize the 3D Euclidean rigid motion transformation.

The model of motion of lines using motors is very appealing for the design of an extended Kalman filter for the motion estimation. For the design of the filter we started with the rotor extended Kalman filter. As a natural extension of it we set the theoretic foundations of the motor extended Kalman filter (MEKF).

The MEKF algorithm has the virtue of estimating the rotation and the translation transformations simultaneously. Since the most of recursive algorithms in the literature compute the translation and rotation transformations separately we can claim that this is one of the most important advantages of the MEKF. Additionally using the modeling of the lines in the motor algebra we could linearize the nonlinear measurement model which does not face singularities. The dynamic motion model using motors as states is useful to formulate effectively and to compute the screw motion of a line. In the algorithm of MEKF, we modified the step of the estimation to satisfy certain geometric constraints, which made the estimation converge faster to a proper motor state. Tests with simulated data confirmed that the tuning of the MEKF parameters improve substantially the MEKF capabilities.

We compared the MEKF with an analytical method using simulated data. The comparison showed that the MEKF is a better estimation approach for the dynamic motion problem and due to its nature it can be used as online approach as opposed to batch methods. We presented also a real application of visual guided robot manipulation. This experiment aims to show the MEKF as an online algorithm for real applications. The vision system was calibrated beforehand using controlled robot movements and an effective hand-eye calibration method. Thereafter the system is ready to supply

the 3D line coordinates to the MEKF. The recovery of the parameters of the 3D lines was carried out using a stereo vision system, techniques of Hough transform and shape filtering for the line matching. Using the 3D lines the MEKF algorithm estimated efficiently the relative motion between its end-joint and a 3D reference line. Since the primer aim of this paper is to show the MEKF algorithm working online, we are sure there will be more accurate and faster methods for camera calibration and hand-eye calibration for future applications, therefore the use of the MEKF will be even more attractive for the estimation of screw motions using 3D line observations. Our future research includes the use of MEKF for robot localization, obstacle avoidance for mobile robots and control of binocular or trinocular heads.

References

1. K.S. Arun, T.S. Huang, and S.D. Blostein, "Least-Squares fitting of two 3D point sets," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 9, No. 5, pp. 698–700, 1987.
2. A.J. Azarbayejani, H. Bradley, and A. Pentland, "Recursive estimation of structure and motion using relative orientation constraints," In *IEEE Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, June 1993, pp. 294–299.
3. I.Y. Bar-Itzhack and Y. Oshman, "Attitude determination from vector observations: Quaternion estimation," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 21, No. 1, pp. 128–135, 1985.
4. E. Bayro-Corrochano, "The geometry and algebra of kinematics," In *Geometric computing with clifford algebra*, G. Sommer (Ed.), Springer-Verlag, 2000.
5. E. Bayro-Corrochano, S. Buchholz, and G. Sommer, "Self-organizing clifford neural network," in *Proceedings of the International Conference of Neural Networks, ICNN'96*, June 3–6, 1996, Washington D.C., USA, Vol. 1, pp. 120–125.
6. E. Bayro-Corrochano, K. Daniilidis, and G. Sommer, "Motor algebra for 3D kinematics: The case of the hand-eye calibration," *Journal of Mathematical Imaging and Vision*, October 2000, Vol. 13(??), pp. 79–99.
7. J. Laseby and E. Bayro-Corrochano, "Analysis and computation of projective invariants from multiple views in the geometric algebra framework," *IJPRAI*, Vol. 13, No. 8, 1999, pp. 1105–1121.
8. W.K. Clifford, "Preliminary sketch of bi-quaternions," in *Proc. London Math. Soc.*, Vol. 4, pp. 381–395, 1873.
9. W.K. Clifford, "Applications of Grassmann's extensive algebra," *Am. J. Math.* Vol. 1, pp. 350–358, 1878.
10. K. Daniilidis and E. Bayro-Corrochano, "The dual quaternion approach to hand-eye calibration," *IEEE Proceedings of ICPR'96 Viena, Austria*, Vol. I, Aug. 1996, pp. 318–322.
11. H. Grassmann, "Der Ort der Hamilton'schen Quaternionen in der Ausdehnungslehre," *Math. Ann.*, Vol. 12, p. 375, 1877.
12. D. Hestenes, "Space-time algebra," *Gordon and Breach*, 1966.
13. D. Hestenes, *New Foundations for Classical Mechanics*, D. Reidel: Dordrecht, 1986.
14. D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*, D. Reidel: Dordrecht, 1984.
15. R. Horaud and F. Dornaika, "Hand-eye calibration," *Intern. Journal of Robotics Research*, Vol. 14, pp. 195–210, 1995.
16. E.J. Lefferts, F.L. Markley, and M.D. Shuster, "Kalman filtering for spacecraft attitude estimation," *AIAA Journal on Guidance, Control, and Dynamics*, Vol. 5, pp. 417–429, 1982.
17. R.E. Kalman, "A new approach to linear filtering and prediction problems," *Trans ASME Journal of Basic Engineering*, Vol. 82, pp. 35–45, 1960.
18. S. Kunze, "Ein Hand-Auge-System zur visuell basierten Lokalisierung und Identifikation von Objekten," Ms. Thesis, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik, 1999.
19. P. Maybeck, *Stochastic Models, Estimation and Control*, Vol. 1, Academic Press: New York, 1979.
20. R. Sabata and J.K. Aggarwal, "Estimation of motion from a pair of range images: A review," *CVGIP: Image Understanding*, Vol. 54, pp. 309–324, 1991.
21. H.W. Sorenson, "Kalman filtering techniques," in *Advances in Control Systems Theory and Applications*, C.T. Leondes (Ed.), Vol. 3, Academic Press: New York, pp. 218–292, 1966.
22. Z. Zhang and O. Faugeras, *3D Dynamic Scene Analysis*. Springer-Verlag: Berlin 1992.



Eduardo Bayro-Corrochano gained his Ph.D. in cognitive computer science in 1993 from the University of Wales at Cardiff. From 1995 to 1999 he was researcher and lecturer at the Institute for Computer Science, Christian Albrechts University, Kiel, Germany, working on the applications of Clifford geometric algebra to cognitive systems.

He is a contributor, an editor and an author of books on applications of Clifford geometric algebra in computer science and engineering. He is also reviewer in various journals. In September 1997 he acted as program chair in the Int. Workshop Algebraic Frames for the Perception-Action Cycle, AFPAC'97, held in Kiel, Germany. In July 1999 together with Garret Sobczyk, he organized and chaired the Int. Workshop on Applications of Geometric Algebra in Computer Vision, Robotics, Computer Science and Engineering AGACSE'99 during the 5th International Conference on Clifford Algebras and their Applications in Mathematical Physics, held in Ixtapa-Zihuatanejo, Mexico.

His current research interests include geometric neural networks, intelligent control, computer vision, visually guided robotics, geometry in color image processing and the use of bivector Lie algebras for perception of invariants and robot maneuvering. In his research and academic activities he promotes indefatigably a geometric approach for the design and implementation of real time artificial perception-action systems.



Yiwen Zhang gained his M.S. degree in Electronic and Electrical Engineering in 1994 from the Fudan University of Shanghai, P.R.

China. From 1995 to 1997 he worked in China as a consultant engineer in computer networking. Since 1998 he has been pursuing his Ph.D. at the Institute for Computer Science, Christian Albrechts University, Kiel, Germany, working on the applications of Clifford geometric algebra for visual guided robotics.

His research interests involve Kalman filter techniques, multivector estimation, intelligent control, image processing and robotics.