Visual Position Tracking using Dual Quaternions with Hand-Eye Motion Constraints

Olsson, Tomas; Bengtsson, Johan; Robertsson, Anders; Johansson, Rolf

Published in:
Proceedings. ICRA '03.

DOI:
10.1109/ROBOT.2003.1242130

2003

Proceedings of the 2003 IEEE
International Conference on Robotics & Automation
Taipei, Taiwan, September 14-19, 2003

Visual Position Tracking using Dual Quaternions with Hand-Eye Motion Constraints

Tomas Olsson, Johan Bengtsson, Anders Robertsson, Rolf Johansson

Department of Automatic Control, Lund Institute of Technology, Lund University, SE-221 00 Lund Sweden;
E-mail: {Tomas.Olsson, Johan.Bengtsson, Anders.Robertsson, Rolf.Johansson}@control.lth.se

Abstract—**In this paper a method for contour-based rigid body tracking with simultaneous camera calibration is developed. The method works for a single eye-in-hand camera with unknown hand-eye transformation, viewing a stationary object with unknown position. The method uses dual quaternions to express the relationship between the camera- and end-effector screws. It is shown how using the measured motion of the robot end-effector can improve the accuracy of the estimation, even if the relative position and orientation between sensor and actuator is completely unknown.**

The method is evaluated in simulations on images from a real-time 3D rendering system. The system is shown to be able to track the pose of rigid objects and changes in intrinsic camera parameters, using only rough initial values for the parameters. The method is finally validated in an experiment using real images from a camera mounted on an industrial robot.

I. Introduction

A. Visual position tracking

Tracking and estimating the position of objects using measurements from one or several cameras has been an active research topic for many years. A special case is tracking of the position and orientation of rigid objects. Many methods for rigid body tracking work by minimizing some measure of the image space error as a function of the unknown position and orientation parameters. The minimization can for instance be performed using standard non-linear optimization methods such as Gauss-Newton or Levenberg-Marquardt. Another option is to use Kalman filtering techniques [1], [2].

The position and orientation can be parameterized in different ways, such as roll-pitch-yaw angles [1], quaternions or dual quaternions [2]. There are also various ways to measure the image space error, the most common measurements are the positions of point features [1] or line features [2], or point-to-contour errors [3], [4]. The point-to-contour method has a major advantage in that it does not require exact matching of features, only the error in the normal direction at a number of points on a contour. This only requires a one-dimensional search for features (edges).

In [3] it is shown that not only the position and orientation can be estimated, but also the intrinsic parameters of the camera (focal length, aspect ratio and principal point). In [4], on the other hand, it is pointed out that the problem of simultaneously tracking position and intrinsic parameters is ill-conditioned when the points of the object lie on a plane parallel to the image plane. This causes the Jacobian matrix, relating errors in position- and intrinsic parameters to image errors, to lose rank. Because of noise, this problem extends also to positions where the relative depth of the object points in the camera is small. A multi-camera tracking system is suggested as a possible solution to this problem.

B. Quaternions and dual quaternions

Unit quaternions [5], [6], [2] is a common represention of rotations. Similarly to real quaternions, dual quaternions are defined as $\check{\mathbf{q}} = (\check{q}^0, \check{\vec{\mathbf{q}}})$, where $\check{q}^0 = q^0 + \varepsilon q'^0$ is a dual number with $\varepsilon^2 = 0$, and where $\check{\vec{\mathbf{q}}} = \vec{\mathbf{q}} + \varepsilon\vec{\mathbf{q}}'$ is a dual vector. The dual quaternion operations are

$$\check{\mathbf{q}}_1 + \check{\mathbf{q}}_2 = (\check{q}_1^0 + \check{q}_2^0, \check{\vec{\mathbf{q}}}_1 + \check{\vec{\mathbf{q}}}_2) \quad (1)$$

$$k\check{\mathbf{q}} = (k\check{q}^0, k\check{\vec{\mathbf{q}}}) \quad (2)$$

$$\check{\mathbf{q}}_1\check{\mathbf{q}}_2 = (\check{q}_1^0\check{q}_2^0 - \check{\vec{\mathbf{q}}}_1^T\check{\vec{\mathbf{q}}}_2, \check{q}_1^0\check{\vec{\mathbf{q}}}_2 + \check{q}_2^0\check{\vec{\mathbf{q}}}_1 + \check{\vec{\mathbf{q}}}_1 \times \check{\vec{\mathbf{q}}}_2). \quad (3)$$

We will often write the dual quaternion as the sum of the real and dual parts $\mathbf{q} + \varepsilon\mathbf{q}'$. Its norm is given by $\|\check{\mathbf{q}}\|^2 = \check{\mathbf{q}}\bar{\check{\mathbf{q}}}$ with $\bar{\check{\mathbf{q}}} = \bar{\mathbf{q}} + \varepsilon\bar{\mathbf{q}}'$, and the unity conditions become

$$\mathbf{q}\bar{\mathbf{q}} = 1, \quad \bar{\mathbf{q}}\mathbf{q}' + \bar{\mathbf{q}}'\mathbf{q} = 0. \quad (4)$$

Unit dual quaternions can be used to represent general rigid transformations including translations, similarly to the way rotations can be represented by real quaternions. It can be shown, see [6], that the rigid transformation of a line through the point $\vec{\mathbf{p}}$, represented by its direction $\vec{\mathbf{n}}$ and moment $\vec{\mathbf{m}} = \vec{\mathbf{p}} \times \vec{\mathbf{n}}$, is given by $\check{\mathbf{q}}(\mathbf{n} + \varepsilon\mathbf{m})\bar{\check{\mathbf{q}}}$, where $\vec{\mathbf{n}}$ and $\vec{\mathbf{m}}$ are expressed as quaternions $\mathbf{n} = (0, \vec{\mathbf{n}})$ and $\mathbf{m} = (0, \vec{\mathbf{m}})$, respectively. The dual quaternion itself is $\mathbf{q} + \varepsilon\mathbf{q}'$, where $\mathbf{q}$ is the quaternion describing the rotation, and where $\mathbf{q}' = \mathbf{t}\mathbf{q}/2$ with $\mathbf{t} = (0, \vec{\mathbf{t}})$ being the translation.

C. Screws and robot motion constraints

Screws: According to Chasles' theorem [5], a general rigid transformation can be modeled as a rotation about

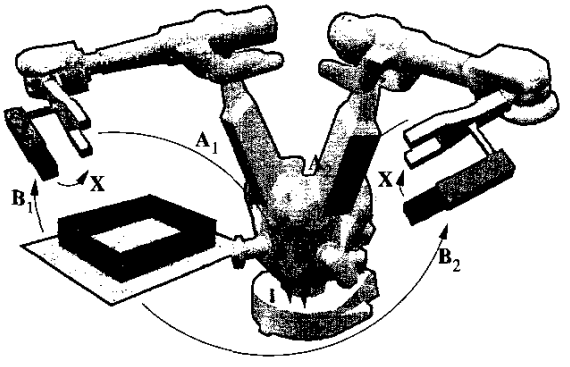

Fig. 1. Two different positions of the robot and relevant transformations.

an axis not through the origin and a translation along the rotation axis. The parameters of the screw are the direction $\vec{\mathbf{n}}$ and the moment $\vec{\mathbf{m}}$ of the screw axis line, the rotation angle θ, and the translation (pitch) d along $\vec{\mathbf{n}}$. Together with the constraints $\vec{\mathbf{n}}^T\vec{\mathbf{n}} = 1$ and $\vec{\mathbf{n}}^T\vec{\mathbf{m}} = 0$ these parameters constitute the six degrees of freedom of a rigid transformation. It can be shown that the dual quaternion corresponding to the screw with parameters $\vec{\mathbf{n}}$, $\vec{\mathbf{m}}$, θ and d can be written as

$$\check{\mathbf{q}} = (\cos(\check{\theta}/2), \sin(\check{\theta}/2)\check{\vec{\mathbf{l}}}), \tag{5}$$

where the dual angle is $\check{\theta} = \theta + \varepsilon d$, and the line is given by $\check{\vec{\mathbf{l}}} = \vec{\mathbf{n}} + \varepsilon\vec{\mathbf{m}}$.

Robot motion constraints: The well known hand-eye equation

$$\mathbf{AX} = \mathbf{XB}, \tag{6}$$

with $\mathbf{A} = \mathbf{A}_2^{-1}\mathbf{A}_1$ and $\mathbf{B} = \mathbf{B}_2\mathbf{B}_1^{-1}$ from Fig. 1, can be written using dual quaternions as

$$\check{\mathbf{a}} = \check{\mathbf{q}}\check{\mathbf{b}}\bar{\check{\mathbf{q}}}. \tag{7}$$

In [6], it is shown that the scalar parts of $\check{\mathbf{a}}$ and $\check{\mathbf{b}}$ are equal, which can easily be shown as follows

$$\begin{aligned} Sc(\check{\mathbf{a}}) &= \frac{1}{2}(\check{\mathbf{a}} + \bar{\check{\mathbf{a}}}) = \frac{1}{2}(\check{\mathbf{q}}\check{\mathbf{b}}\bar{\check{\mathbf{q}}} + \check{\mathbf{q}}\bar{\check{\mathbf{b}}}\bar{\check{\mathbf{q}}}) = \\ &= \frac{1}{2}\check{\mathbf{q}}(\check{\mathbf{b}} + \bar{\check{\mathbf{b}}})\bar{\check{\mathbf{q}}} = Sc(\check{\mathbf{b}})\check{\mathbf{q}}\bar{\check{\mathbf{q}}} = Sc(\check{\mathbf{b}}). \end{aligned} \tag{8}$$

In terms of the screw parameters, Eq. (8) means that the angle and pitch of the camera screw and the robot end-effector screw must be equal [6]. This is known as the Screw Congruence Theorem, see [7].

D. Problem formulation

The purpose of this paper is to develop methods for real-time rigid body tracking with simultaneous calibration and tracking of intrinsic parameters. We intend to show that a dual quaternion parameterization of the object pose, together with measurements of the robot motion, can be used to formulate constraints on the estimated motion. The constraints can be expressed as linear equations in the states.

II. Modeling

Consider a manipulator with a single camera attached to its end-effector, viewing a stationary object. We assume that only rough initial values of the intrinsic camera parameters and the position/orientation of the object are known, but that a CAD model of the object is available. The motion of the robot end-effector is related to the motion of the camera through the hand-eye equation (6), where the relative sensor-actuator pose $\mathbf{X}$ is unknown. We assume that the camera can be modeled as a four parameter pinhole camera

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & u_0 \\ 0 & \gamma f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{9}$$

with λ corresponding to the depth of point (X, Y, Z) in the camera. The parameters to be estimated are f, γ, u_0, v_0, and some parameterizations of $\mathbf{R} \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$.

A. Extended Kalman Filtering (EKF)

The motion of the system can be written as a non-linear discrete-time dynamic system

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) \tag{10}$$

$$\mathbf{g}(\mathbf{p}_k, \mathbf{x}_k) = 0 \tag{11}$$

with $\mathbf{x}_k \in \mathbb{R}^n$ the state of the system, $\mathbf{p}_k \in \mathbb{R}^m$ a vector of measured outputs, $\mathbf{f}$ a known function describing the system dynamics, and $\mathbf{g}$ a known function relating the state to the output. The state vector is chosen as

$$\mathbf{x} = \begin{pmatrix} \mathbf{q}^T & \mathbf{q}'^T & f & \gamma & u_0 & v_0 & \omega^T & \mathbf{v}^T \end{pmatrix}^T \tag{12}$$

where $\mathbf{q}, \mathbf{q}' \in \mathbb{R}^4$ is the vector representation of the object-camera dual quaternion $\check{\mathbf{q}} = \mathbf{q} + \varepsilon\mathbf{q}'$, and $\mathbf{v}$, $\omega \in \mathbb{R}^3$ are the velocity and angular velocity.

Measurement model: In the function $\mathbf{g}(\mathbf{p}_k, \mathbf{x}_k)$ we have all the measurement equations, and the constraints on the state vector. For a point feature, denoted by index i, the measurement would be its image coordinates, denoted by $\mathbf{p}_{k,i} = (u_i, v_i)^T \stackrel{\text{def}}{=} \mathbf{h}_i(\mathbf{x})$, which is known from Eq. (9) to be related to the states by the equations

$$\begin{aligned} \hat{\mathbf{h}}_i(\mathbf{x}) &= \begin{pmatrix} \hat{u}_i \\ \hat{v}_i \\ \hat{w}_i \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{r}_x^T(\mathbf{q}) & t_x(\check{\mathbf{q}}) \\ \mathbf{r}_y^T(\mathbf{q}) & t_y(\check{\mathbf{q}}) \\ \mathbf{r}_z^T(\mathbf{q}) & t_z(\check{\mathbf{q}}) \end{pmatrix} \begin{pmatrix} \mathbf{X}_i \\ 1 \end{pmatrix} = \\ &= \begin{pmatrix} f(\mathbf{r}_x^T(\mathbf{q})\mathbf{X}_i + t_x(\check{\mathbf{q}})) + u_0(\mathbf{r}_z^T(\mathbf{q})\mathbf{X}_i + t_z(\check{\mathbf{q}})) \\ \gamma f(\mathbf{r}_y^T(\mathbf{q})\mathbf{X}_i + t_y(\check{\mathbf{q}})) + v_0(\mathbf{r}_z^T(\mathbf{q})\mathbf{X}_i + t_z(\check{\mathbf{q}})) \\ \mathbf{r}_z^T(\mathbf{q})\mathbf{X}_i + t_z(\check{\mathbf{q}}) \end{pmatrix} \end{aligned} \tag{13}$$

where $\mathbf{K}$ is the matrix of intrinsic parameters in Eq. (9). The image space coordinates are given by

$$\mathbf{h}_i(\mathbf{x}) = \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} \hat{u}_i/\hat{w}_i \\ \hat{v}_i/\hat{w}_i \end{pmatrix}. \quad (14)$$

The rotation matrix can be calculated directly from the unit quaternion $\mathbf{q}$, and the translation can be obtained from $\check{\mathbf{q}}$ as

$$\mathbf{t} = 2\mathbf{q}'\bar{\mathbf{q}}. \quad (15)$$

If $\mathbf{p}_{k,i}$ are the measured image coordinates, we can write the measurement equation for this point as

$$\mathbf{g}_i(\mathbf{p}_{k,i}, \mathbf{x}_k) = \mathbf{p}_{k,i} - \mathbf{h}_i(\mathbf{x}_k) = 0 \quad (16)$$

This equation can be linearized around the predicted state $\mathbf{x}_k^{(p)}$, which gives the approximation

$$\begin{aligned} \mathbf{g}_i(\mathbf{p}_{k,i}, \mathbf{x}_k) &\approx \mathbf{g}_i(\mathbf{p}_{k,i}, \mathbf{x}_k^{(p)}) + \frac{\partial \mathbf{g}_i}{\partial \mathbf{x}}(\mathbf{p}_{k,i}, \mathbf{x}_k^{(p)})(\mathbf{x}_k - \mathbf{x}_k^{(p)}) \\ &= \mathbf{p}_{k,i} - \mathbf{h}_i(\mathbf{x}_k^{(p)}) + \frac{\partial \mathbf{g}_i}{\partial \mathbf{x}}(\mathbf{p}_{k,i}, \mathbf{x}_k^{(p)})(\mathbf{x}_k - \mathbf{x}_k^{(p)}) \approx 0 \quad (17) \end{aligned}$$

In our system however, the only image measurements available are the point-to-contour error in the predicted (local) normal direction of the contour, which can be approximated with the normal component of the error

$$\mathbf{g}_i^{(n)} = \mathbf{n}_i^T(\mathbf{p}_{k,i} - \mathbf{h}_i(\mathbf{x}_k^{(p)})) \quad (18)$$

Eq. (17) can then be rewritten as

$$\begin{aligned} 0 &= \mathbf{g}_i^{(n)}(\mathbf{p}_{k,i}, \mathbf{x}_k) \approx \mathbf{n}_i^T(\mathbf{p}_{k,i} - \mathbf{h}_i(\mathbf{x}_k^{(p)})) + \\ &+ \mathbf{n}_i^T \frac{\partial \mathbf{g}_i}{\partial \mathbf{x}}(\mathbf{p}_{k,i}, \mathbf{x}_k^{(p)})(\mathbf{x}_k - \mathbf{x}_k^{(p)}) \quad (19) \end{aligned}$$

which can be expressed on linear form as

$$\begin{aligned} \mathbf{y}_{k,i} &= \mathbf{C}_{k,i}\mathbf{x}_k \quad (20) \\ \mathbf{C}_{k,i} &= \mathbf{n}_i^T \frac{\partial \mathbf{g}_i}{\partial \mathbf{x}}(\mathbf{p}_{k,i}, \mathbf{x}_k^{(p)}) = -\mathbf{n}_i^T \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}}(\mathbf{p}_{k,i}\mathbf{x}_k^{(p)}) \\ \mathbf{y}_{k,i} &= -\mathbf{n}_i^T \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}}(\mathbf{p}_{k,i}, \mathbf{x}_k^{(p)})\mathbf{x}_k^{(p)} - \mathbf{n}_i^T(\mathbf{p}_{k,i} - \mathbf{h}_i(\mathbf{x}_k^{(p)})) \end{aligned}$$

The Jacobian of of $\mathbf{h}_i$ can be calculated by direct differentiation of Eq. (13) with respect to the elements of $\mathbf{x}$, combined with the equations

$$\begin{pmatrix} u_i' \\ v_i' \end{pmatrix} = \begin{pmatrix} \frac{\hat{u}_i'}{\hat{w}_i} - \frac{\hat{u}_i\hat{w}_i'}{\hat{w}_i^2} \\ \frac{\hat{v}_i'}{\hat{w}_i} - \frac{\hat{v}_i\hat{w}_i'}{\hat{w}_i^2} \end{pmatrix} \quad (21)$$

where $\hat{u}_i'$ denotes the differentiation of $\hat{u}_i$ with respect to the relevant quantity.

The constraints on the dual quaternion in Eq. (4) can be written on vector form

$$\mathbf{q}^T\mathbf{q} = 1, \quad \mathbf{q}^T\mathbf{q}' = 0 \quad (22)$$

and included by linearizing around the prediction $\mathbf{x}_k^{(p)}$, which gives us two linear equations in $\mathbf{q}$ and $\mathbf{q}'$

$$\begin{aligned} 1 &\approx \mathbf{q}^{(p)T}\mathbf{q}^{(p)} + 2\mathbf{q}^{(p)T}(\mathbf{q} - \mathbf{q}^{(p)}) \quad (23) \\ 0 &\approx -\mathbf{q}^{(p)T}\mathbf{q}'^{(p)} + \mathbf{q}'^{(p)T}\mathbf{q} + \mathbf{q}^{(p)T}\mathbf{q}', \quad (24) \end{aligned}$$

which can be included among the output equations.

Including the robot motion constraints from Eq. (8) is also straightforward. Consider two different robot poses, represented by the the dual quaternions $\check{\mathbf{q}}_{B_1}$ and $\check{\mathbf{q}}_{B_2}$, and the corresponding relative object-camera poses $\check{\mathbf{q}}_{A_1}$ and $\check{\mathbf{q}}_{A_2}$. We know from Eq. (8) that the scalar parts of $\check{\mathbf{q}}_A = \check{\mathbf{q}}_{A_2}\check{\mathbf{q}}_{A_1}$ and $\check{\mathbf{q}}_B = \check{\mathbf{q}}_{B_2}\check{\bar{\mathbf{q}}}_{B_1}$ must be equal. Define the scalar part of the relative robot pose $\check{\mathbf{q}}_B$ as $\check{q}_B^{(0)} = q_B^{(0)} + \varepsilon q_B'^{(0)}$, which can be calculated directly from the forward kinematics of the robot. The scalar part of $\check{\mathbf{q}}_A$ can be seen from Eq. (3) to be $\mathbf{q}_{A_1}^T\mathbf{q}_{A_2} + \varepsilon(\mathbf{q}_{A_1}'^T\mathbf{q}_{A_2} + \mathbf{q}_{A_1}^T\mathbf{q}_{A_2}')$, with the quaternions written on vector form. Setting the scalar parts equal gives us two more linear equations

$$\begin{aligned} \mathbf{q}_{A_1}^T\mathbf{q}_{A_2} &= q_B^{(0)} \quad (25) \\ \mathbf{q}_{A_1}'^T\mathbf{q}_{A_2} + \mathbf{q}_{A_1}^T\mathbf{q}_{A_2}' &= q_B'^{(0)}, \quad (26) \end{aligned}$$

which can also be added to the system measurement equation, which can now be formulated as

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \delta_k \quad (27)$$

where δ_k is a sequence of uncorrelated Gaussian noise, and the vector $\mathbf{y}_k$ and time-varying matrix $\mathbf{C}_k$ are obtained by stacking equations (19) for each edge search point, and adding constraints from Eq. (23)–(24) and (25)–(26). Any number of robot motion constraints can be added to the measurement equation. In general each position used gives us two independent constraints on the pose, meaning that three positions are sufficient to completely constrain the estimated pose. This can be compared to the problem of hand-eye calibration, where it is well known that three positions are necessary for the calculation of the hand-eye transformation [6].

State dynamics model: In the function $\mathbf{f}$ in the state update equation (11), we update the estimate of the dual quaternion using the equations

$$\begin{aligned} \dot{\mathbf{q}} &= \frac{1}{2}\omega\mathbf{q} = \frac{1}{2}(0, \vec{\omega}_k)\mathbf{q} \quad (28) \\ \dot{\mathbf{q}}' &= \frac{1}{2}\dot{\mathbf{t}}\mathbf{q} + \frac{1}{2}\mathbf{t}\dot{\mathbf{q}} = \frac{1}{2}\mathbf{v}\mathbf{q} + \frac{1}{4}\mathbf{t}\omega\mathbf{q} \quad (29) \end{aligned}$$

where the details can be found in [2]. Discretizing Eqs. (28) and (29) using sample time h, the noise-free state update equation becomes

$$\begin{pmatrix} \mathbf{q}_{k+1} \\ \mathbf{q}'_{k+1} \\ \check{\mathbf{K}}_{k+1} \\ \omega_{k+1} \\ \mathbf{v}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \frac{h}{2}\mathbf{Q}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \frac{h}{4}\mathbf{T}_k\mathbf{Q}_k & \frac{h}{2}\mathbf{Q}_k \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{q}_k \\ \mathbf{q}'_k \\ \check{\mathbf{K}}_k \\ \omega_k \\ \mathbf{v}_k \end{pmatrix} \quad (30)$$

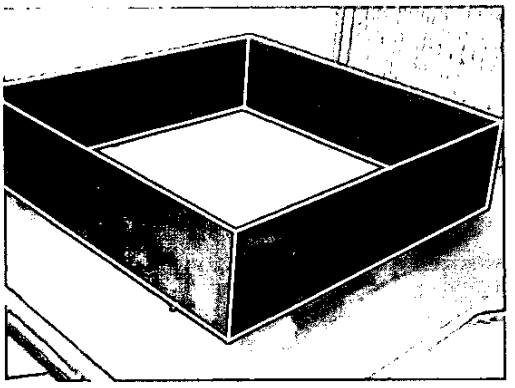

Fig. 2. Example of image with superimposed object model, where hidden features have been removed.

where $\tilde{\mathbf{K}} = (f, \gamma, u_0, v_0)^T$, and where the matrices $\mathbf{Q}_k$ and $\mathbf{T}_k$ correspond to the quaternion multiplications with $\mathbf{q}_k = (q_0, q_1, q_2, q_3)$ and $\mathbf{t}_k = (0, t_x, t_y, t_z) = 2\mathbf{q}_k'\bar{\mathbf{q}}_k$ in Eqs. (28) and (29). Adding uncorrelated Gaussian noise ε_k, Eq. (30) can be written as

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \varepsilon_k. \tag{31}$$

The linearized equations in (30) and (27) can then be used to recursively update the state estimate using an Extended Kalman filter. The equations for the EKF can for instance be found in [8], [1], [2].

B. Object modeling and feature selection/localization

The object models consist of a number of planar surfaces connected at their edges, see Fig. 2. No assumptions are made about the shape of the planar surfaces, although in the experiments we use an object with only straight edges. At each step in the tracking visible object edges are selected, based on the predicted object pose and a pre-generated Binary Search Partitioning (BSP) tree description of the object, see [9]. The BSP tree recursively divides the surfaces in the object into "in front" and "behind", until we have a perfect front-to-back ordering. The surfaces are then processed front-to-back, each surface is clipped against all surfaces in front of it, and a number of search points are selected on each visible edge. The image position measurements are then obtained from a one-dimensional edge localization in the local edge normal direction at each point. The edges are found from the convolution with a differentiated Gauss kernel, at three different scales. To increase robustness only points where a clear single edge is detected are used by the tracker.

III. Experiments

The algorithm is first evaluated in simulations using images generated using an image-generation program based on OpenGL, making it possible to simulate phenomena such as occlusion, specular reflections and noise from a cluttered background.

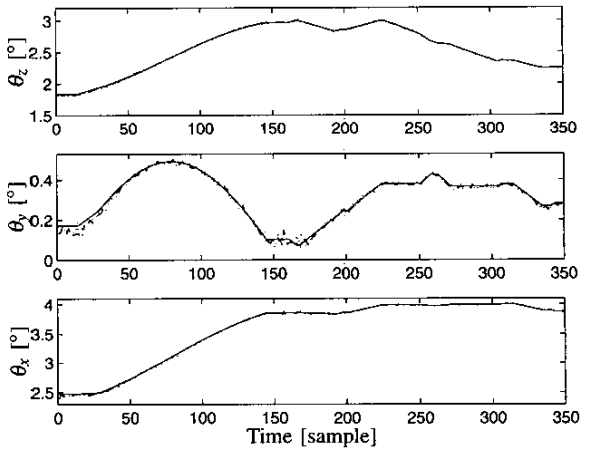

Fig. 3. Tracking of θ. The diagram shows the real orientation (*solid*), estimated orientation using four constraints (*dashed*), and estimated orientation using two constraints (*dotted.*)

The experiments are performed in two steps. The tracker is initialized with a poor initial guess for the intrinsic camera parameters, which is used to get a very rough estimate of the object-camera pose. We then run the tracker for a little over a second with the robot stationary to get a good initial estimate of the state. During the initialization phase the robot motion constraints are not used, since they would require a good initial guess for the object pose. When the state estimate has converged, the tracker is started, using the initial estimate of the state as $\check{\mathbf{q}}_{A_1}$ to constrain the position estimate according to Eq. (25)–(26).

IV. Experimental results

In this section the presented methods will be validated. There will be a comparison between when only using Eqs. (23) and (24) and when also using Eqs. (25) and (26). The first will be referred to as the *two constraint case* and the latter as the *four constraints case*. In the study we have looked at the mean of the absolute estimation error, which will be denoted with Δ. The number of edge search points varied between 100 and 250 during the motion .

A. Visual position tracking

Figs. 3 and 4 show the result of tracking the orientation θ and the translation $\mathbf{t}$. Both with four and with two constraints the tracking of the position is satisfactory. There are some differences in the tracking accuracy, see Table I. We see that with four constraints the mean error in the estimation of θ and $\mathbf{t}$ is reduced. Table I shows results using different conditions and number of constraints in the estimation of the object pose. For case 1 the only noise is from the image measurements. For case 2 extra noise $\in N(0,3)$ was added. The initial state covariance, $\mathbf{P}_0$, and

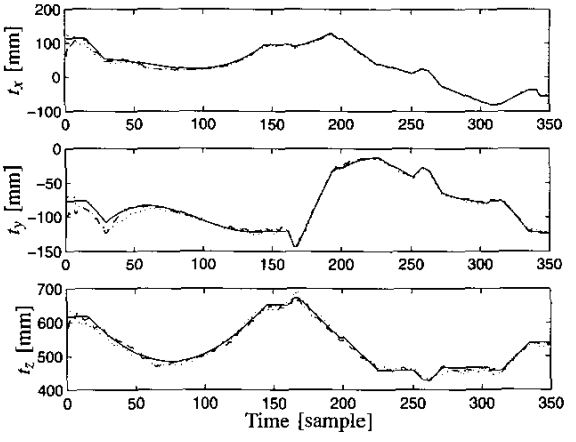

Fig. 4. Tracking of t. The diagram shows the real translation (*solid*), estimated translation using four constraints (*dashed*), and estimated translation using two constraints (*dotted*.)

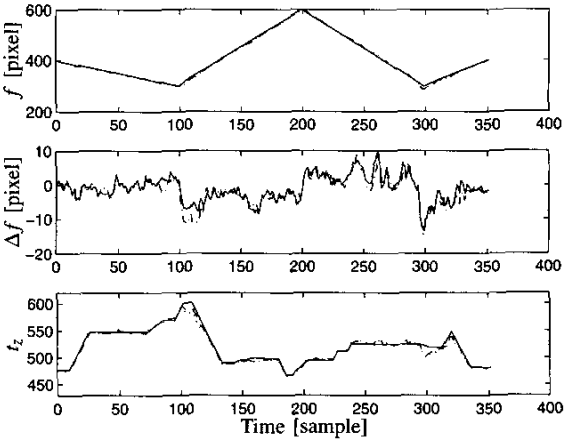

Fig. 5. Experiment where f is varying between 300 and 600 pixels. The diagram shows the real values (*solid*), estimation using four constraints (*dashed*), and estimation using two constraints (*dotted*.)

state noise covariance, $\mathbf{Q}$, for case 1 and 2 were set to

$$\mathbf{P}_0 = \text{diag}(0.1^2 \cdot \mathbf{1}_{1\times 8} \;\; 50^2 \;\; 0.1^2 \;\; 40^2 \;\; 40^2 \;\; \mathbf{0}_{1\times 6})$$
$$\mathbf{Q} = \text{diag}(\mathbf{0}_{1\times 8} \;\; 3^2 \;\; 0.01^2 \;\; 0.2^2 \;\; 0.2^2 \;\; 0.1^2 \cdot \mathbf{1}_{1\times 6}).$$

The output noise variance was set to $E(\delta_k^2) = 1$ in case 1 and 3 and to $E(\delta_k^2) = 3^2$ in case 2 and 4.

B. Varying focal length.

Fig. 5 shows results of when the focal length was varied between 300 and 600 pixels. Still the tracking of the focal length was successful, and the effect on the depth estimation was negligible.

C. Incorrect initial values

Figs. 6 and 7 show results of when the tracker starts with incorrect initial values, both for the intrinsic parameters and for the pose of the object. After approximately 30 samples the intrinsic parameters and the position have converged to their correct value. The intrinsic camera parameters in this experiment were $f = 400$, $\gamma = 1.0$, $u_0 = 320$, and $v_0 = 240$.

TABLE I

MEAN TRACKING ERRORS USING TWO AND FOUR CONSTRAINTS

Case	1		2	
# constr.	4	2	4	2
$\Delta\theta_z$ (°)	0.122	0.216	0.292	0.541
$\Delta\theta_y$ (°)	0.317	0.313	0.529	0.632
$\Delta\theta_x$ (°)	0.130	0.381	0.221	0.777
Δt_x (mm)	2.105	3.406	2.434	3.310
Δt_y (mm)	2.390	2.093	2.322	2.878
Δt_z (mm)	4.857	4.926	5.704	7.228
$\|\Delta\theta\|$ (°)	0.364	0.538	0.643	1.138
$\|\Delta t\|$ (mm)	5.808	6.345	6.622	8.455

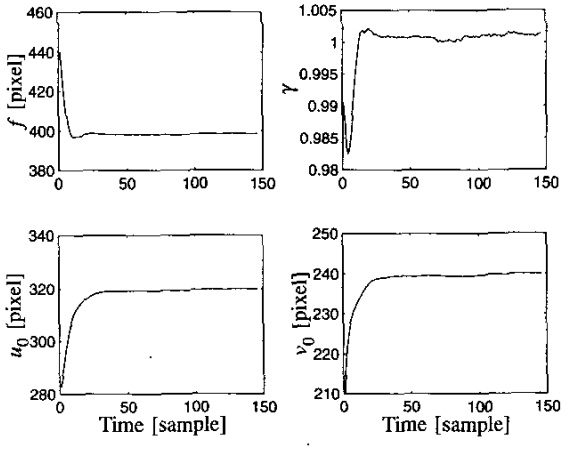

Fig. 6. Transient responses of estimates from incorrect initial values.

D. Real world experiments.

Fig. 8 show the results of an experiment using images from a Sony DFW-V300 640x480 pixels digital camera, see Fig. 2 for an example image. The camera was mounted on an ABB Irb2000 industrial robot. The top figures show the estimated focal length and principal point, which should be compared with the values $f = 1020$ pixels, $u_0 = 344$ pixels and $v_0 = 215$ pixels from an offline camera calibration. The lower figure shows the estimated position of the camera, where the lines indicate the direction of the camera z-axis.

V. DISCUSSION

The use of the robot motion constraints showed an improvement in the estimation of the parameters, even

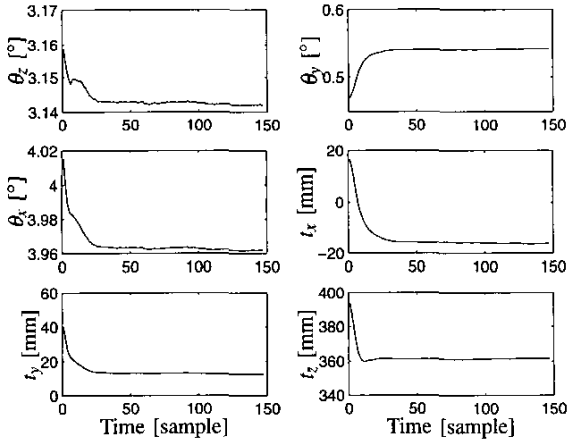

Fig. 7. Transient responses of estimates from incorrect initial values.

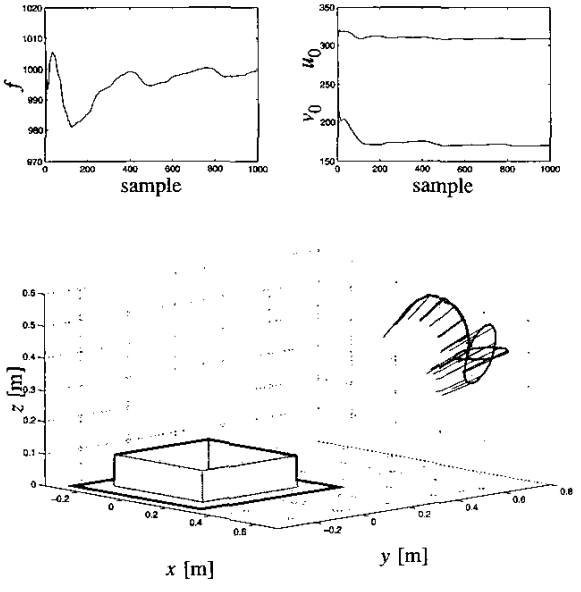

Fig. 8. Estimated focal length, principal point, and trajectory in the real world experiment.

though the hand-eye transformation was unknown. Additionally, it is also intuitively reasonable that the extra constraints should improve the robustness of the tracking against other error sources, such as errors due to the edge detector locking on to false edges. Experiments with multiple constraints show that three or more positions will constrain the position estimation completely, allowing for tracking of very fast motions.

The system is capable of performing a total calibration of all relevant parameters, based on only rough initial values. The method works well also on real world images with low intensity edges. The use of dual quaternions and robot motion constraints in this type of tracker has not previously been demonstrated. The fact that we are able to track even during changes in the intrinsic parameters is an advantage, for instance in vision-based control. This gives the practical advantage of allowing the system to dynamically change its field of view, allowing a wider range of motions.

One apparent drawback with our EKF-based algorithm is that the updating of the state covariance estimates is very time consuming when the number of outputs is large. Effectively, this limits the feasible number of edge search points to a value which is lower than for optimization based methods. The effective sampling interval varied between 66 ms and 91 ms during the simulation, where the variation can be explained by the varying number of search points. With some optimizations it is reasonably to assume that 30 frames/second is achievable for the real-world system, especially since there is no need for the relatively time consuming 3D rendering.

VI. Conclusion

We have developed methods for real-time rigid body tracking with simultaneous calibration and tracking of intrinsic parameters. We have shown that a dual quaternion parameterization can be used to formulate linear robot motion constraints on the estimated states, and that the extra constraints can help to reduce the tracking error. The method has been validated in simulations and experiments on an industrial robot.

VII. References

[1] V. Lippiello, B. Siciliano, and L. Villani, "Objects motion estimation via BSP tree modeling and Kalman filtering of stereo images," in *IEEE Int. Conference on Robotics and Automation*, Washington D.C., 2002, pp. 2968–2973.

[2] J.S. Goddard, *Pose and motion estimation from vision using dual quaternion-based extended Kalman filtering*, Ph.D. thesis, University of Tennessee, Knoxville, TN, 1997.

[3] T. Drummond and R. Cipolla, "Real-time tracking of complex structures with on-line camera calibration," in *British Machine Vision Conference*, 1999, vol. 2, pp. 574–583.

[4] F. Martin and R. Horaud, "Multiple camera tracking of rigid objects," *International Journal of Robotics Research*, vol. 21, no. 2, pp. 97–113, February 2002.

[5] R.M. Murray, Z. Li, and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, FL, 1994.

[6] K. Daniilidis, "Hand-eye calibration using dual quaternions," *International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.

[7] H.H. Chen, "A screw motion approach to uniqueness analysis of head-eye geometry," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 145–151.

[8] A. Gelb, Ed., *Applied Optimal Control*, The M.I.T. Press, Cambridge, Massachusetts, 1974.

[9] A. van Dam, L. Foley, S. Feiner, and J. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, Boston, MA, 2nd edition, 1991.