ORNL/CP-96358

CONF-980117--

Pose and Motion Estimation Using Dual Quaternion-Based Extended Kalman Filtering

J.S. Goddard[a] and M.A. Abidi[b]

[a]Oak Ridge National Laboratory, Oak Ridge, TN

[b]University of Tennessee, Knoxville, TN

ABSTRACT

A solution to the remote three-dimensional (3–D) measurement problem is presented for a dynamic system given a sequence of two-dimensional (2–D) intensity images of a moving object. The 3–D transformation is modeled as a nonlinear stochastic system with the state estimate providing the six-degree-of-freedom motion and position values as well as structure. The stochastic model uses the iterated extended Kalman filter (IEKF) as a nonlinear estimator and a screw representation of the 3–D transformation based on dual quaternions. Dual quaternions, whose elements are dual numbers, provide a means to represent both rotation and translation in a unified notation. Linear object features, represented as dual vectors, are transformed using the dual quaternion transformation and are then projected to linear features in the image plane. The method has been implemented and tested with both simulated and actual experimental data. Simulation results are provided, along with comparisons to a point-based IEKF method using rotation and translation, to show the relative advantages of this method. Experimental results from testing using a camera mounted on the end effector of a robot arm are also given.

Keywords: pose estimation, motion estimation, iterated extended Kalman filter, dual quaternion, line features

1. INTRODUCTION

Estimation of relative 3–D position and orientation (pose) and structure as well as relative motion between two reference frames is an important problem in robotic guidance, manipulation, and assembly as well as in other areas such as photogrammetry, tracking, and object recognition. The general problem presented here is to locate an object and measure its relative motion in three dimensions given a sequence of 2–D intensity images of the object whose position and orientation are known relative to a base reference frame. The 3–D transformation is modeled as a nonlinear stochastic system with the state estimate providing the six-degree-of-freedom motion and position values. The stochastic model uses the IEKF as an estimator and a screw representation of the 3–D transformation based on dual quaternions. Previous solutions have used point-based image features in estimating the structure, pose, and motion. This work, instead, uses image line features as measurement inputs for the estimation. Line features are present in many scenes and objects to a greater extent than point features. They may be more visible than points as well under a wider range of lighting and environmental conditions. Also, straightforward techniques such as the Hough transform and line fitting to edges are available to extract the lines from the images.

Point-based methods of pose estimation rely on the identification and location of feature points on a target object from a 2–D image of the scene. Three- and four-point coplanar targets have been directly used for pose determination with closed-form solutions demonstrated.[1,2] A 3–D location method based on dual-number quaternions has been described by Walker[3] and Phong.[4] Phong uses line features or lines defined by pairs of corresponding points as correspondences between the image and the 3–D object. Real-time methods are intended for those applications requiring fast response for control or needing low computational overhead for operation with low-cost hardware. Abidi and Chandra[1] describe a new, fast, closed-form algorithm for relative pose determination based on the volume measurement of tetrahedra. Other methods of pose determination use line features in determining the pose.[5,6] A number of real-time estimation methods and applications using Kalman filtering have also been described. The problems being solved include not only pose estimation but also motion and 3–D structure of a rigid object. For real-time applications, extended Kalman filtering is the most widely used method of recursive estimation. Ayache

Further author information -
J.S.G.: E-mail: sgo@ornl.gov; Telephone: 423-574-9034
M.A.A.: E-mail: abidi@abidi.engr.utk.edu; Telephone: 423-974-5454

DISCLAIMER

and Faugeras provide a method for representing uncertainty in measurements of points, lines, and planes used in building visual maps of the environment for a mobile robot.[7] A recursive method of estimating motion, structure, and focal length is given by Azarbayejani and Pentland[8] using an extended Kalman filter (EKF). Quaternions are used to represent rotation indirectly. Broida et al.[9] also present a motion and structure estimation solution based on the IEKF. Point-based image features are inputs. A rotational quaternion is estimated directly as part of the state. A Kalman filter approach is used by Lee and Kay for 3–D pose and motion estimation from stereo images.[10] Westmore and Wilson describe the use of an EKF to provide an estimate of the position of a camera mounted on a robot endpoint.[11] Wang and Wilson describe the estimation of both relative position and orientation of a moving object using an EKF based on point features.[12]

Section 2 describes the dual quaternion 3–D transformation representation. The line-based feature recovery from 2–D images is given in Section 3. This section includes the perspective projection of 3–D lines to the image plane and the parameterization of the projected lines in the image plane. Section 4 develops the estimation method using the IEKF. The system and measurement model is given for the pose and structure estimation. Section 5 presents an implementation of the method in a robotic arm application where a single camera is mounted on the end effector of the robot. Results are shown for simulations as well as for actual tests on a robot arm. Section 6 gives a summary and conclusions.

2. DUAL QUATERNION 3–D TRANSFORMATION REPRESENTATION

2.1. Quaternions

A quaternion is a four-component number consisting of a scalar part and three orthogonal parts. Quaternions are an extension of complex numbers to R^4 with three imaginary parts. Formally, a quaternion q can be defined as

$$q = q_0 + q_1\vec{i} + q_2\vec{j} + q_3\vec{k} \tag{1}$$

where each of the q_i is a real number and $\vec{i}$, $\vec{j}$, and $\vec{k}$ are orthogonal imaginary unit vectors. From the above definition, the class of quaternions can be seen to include scalars, ordinary complex numbers, and three-element spatial vectors. Additional detailed properties of quaternions are given by Horn[13] and Chou.[14]

2.2. Dual Numbers

A dual number is defined as

$$d = a + \epsilon b \tag{2}$$

where a, b are real numbers and ϵ is defined as $\epsilon^2 = 0$. The standard arithmetic operations (addition, multiplication, etc.) may be defined. Dual numbers were first proposed by Clifford.[15] Additional operations and other properties of dual numbers are given by Walker.[3] One application for dual numbers is in the representation of a skew angle between two 3–D lines, $\hat{\theta} = \theta + \epsilon d$, where θ is the angle between the lines and d is the minimum distance between the lines.

2.3. Representation of 3–D Rotation and Translation by Dual Number Quaternions

Quaternions are limited to representing rotation in a full 3–D transformation. Translation must be dealt with separately. Dual number quaternions, however, provide a framework that may be used to represent both rotation and translation. Dual number quaternions correspond directly to the screw 3–D transformation representation shown pictorially in Fig. 1. The required parameters include the screw axis, the screw angle, and the screw pitch. The screw axis is described by a line in 3–D space that has direction $\vec{l}$ passing through point $\vec{p}$. A 3–D line is defined by the two three-element vectors $\vec{l}$ and $\vec{m}$ where $\vec{m} = \vec{p} \times \vec{l}$. While six parameters are used in this definition, only four degrees of freedom are present due to the constraints $\vec{l} \cdot \vec{m} = 0$ and $\|\vec{l}\| = 1$.[16] This particular formulation leads to the dual quaternion representation of the line,

$$\hat{\vec{l}} = \vec{l} + \epsilon\vec{m}. \tag{3}$$

The eight parameters in this equation directly correspond to the screw transformation representation. The unit dual quaternion $\hat{q} = r + \epsilon s$ representing the 3–D transformation can be expressed as

$$\hat{q} = \begin{bmatrix} \cos(\hat{\theta}/2) \\ \sin(\hat{\theta}/2)\hat{\vec{l}} \end{bmatrix} \tag{4}$$

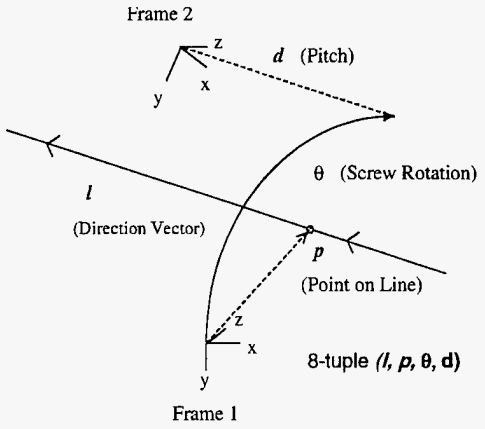

Figure 1. Diagram showing screw form of 3–D transformation between two reference frames. Eight parameters are needed to specify the complete transformation.

where $\hat{\vec{l}}$ is a dual vector that represents the screw axis about which the coordinate system has rotated and translated as given by $\hat{\theta}$.[3,16] The resulting full 3–D transformation of a line can be expressed simply as

$$\hat{l}' = \hat{q}\hat{l}\hat{q}^*. \tag{5}$$

3. RECOVERY OF FEATURE INFORMATION

3.1. Perspective Projection of 3–D Lines

A 3–D line is transformed to the camera reference frame using Eq. (5). Perspective projection is then applied to this transformed line. The projected line lies in a plane defined by the 3–D line and the center of projection that intersects the image plane at $z = -\lambda$. The result is an equation of the projected line in the $z = -\lambda$ plane,

$$m_x x_i + m_y y_i = m_z \lambda, \tag{6}$$

where x_i and y_i are the image plane coordinates. From the normalized form, the direction vector of the image line is

$$\vec{l}_i = \begin{bmatrix} -\frac{m_y}{\sqrt{m_x^2+m_y^2}} \\ \frac{m_x}{\sqrt{m_x^2+m_y^2}} \\ 0 \end{bmatrix} \tag{7}$$

while the $\vec{m}$ vector portion of the image line is

$$\vec{m}_i = \frac{\lambda}{\sqrt{m_x^2+m_y^2}}\vec{m}. \tag{8}$$

Note that the projected 2–D image plane line as given by $\vec{m}_i$ and $\vec{l}_i$ is determined entirely by the $\vec{m}$ vector components of the 3–D object line.

3.2. Representation of Lines in the Image Plane

The result of the perspective projection for 3–D object lines is a set of coplanar lines located in the image plane. Since the lines are restricted to a known plane, only two degrees of freedom are present. The parameterization of the image line is an x,y point called the line point. A line point is defined as the intersection of the line feature with a line passing through the image origin perpendicular to the line feature. Figure 2 illustrates the definition of the line point on the image plane. The line point is unique for all lines except for those lines that pass through the origin. For these cases, it is assumed the lines approach the origin with a distance δ but do not actually pass through the origin. The line point has the advantage of minimum state representation and a simple distance measure as well as

being continuous for all lines. The line point is calculated from the dual vector image line in terms of the 3–D line dual-vector components

$$x_{lp} = \lambda \frac{m_x m_z}{m_x^2 + m_y^2} \tag{9}$$

$$y_{lp} = \lambda \frac{m_y m_z}{m_x^2 + m_y^2}. \tag{10}$$

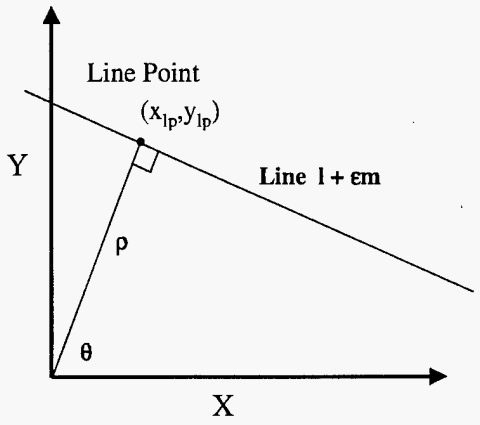

Figure 2. Line point for a line in 2–D image plane is defined as the intersection point of the line and a perpendicular line passing through the origin.

4. ESTIMATION OF POSE AND MOTION

The perspective projection equations for lines from 3–D to 2–D are nonlinear. Calculation of relative position and orientation from the 3–D coordinates of corresponding points in two reference frames is also nonlinear.[2,13] For the minimum three points needed to calculate a pose, up to four solutions are possible. Solutions with four or more points are overconstrained with a minimum error criterion used to determine the result. Motion estimation requires a dynamic system model that is nonlinear in the rotational dynamics. Direct linear estimation techniques, such as the standard Kalman filter, are therefore not applicable. The IEKF, however, is highly suitable as a nonlinear estimator.[17]

4.1. System Model

The state assignment estimates the transformation between the camera and the object reference frames and the first derivatives of this transformation. The assignment is based on the dual quaternion representation of the 3–D transformation. Similar to the approach given by Broida,[9] the state variable assignment with a known object geometry is

$$\vec{s} = [t_x \;\; t_y \;\; t_z \;\; q_0 \;\; q_1 \;\; q_2 \;\; q_3 \;\; v_x \;\; v_y \;\; v_z \;\; \omega_x \;\; \omega_y \;\; \omega_z]^T. \tag{11}$$

Thirteen state variables are present: t_i and v_i terms are the linear translation and linear velocity, respectively; q_i is the rotational quaternion; and ω_i is the rotational velocity in each axis. Translation, rather than the dual part of the dual quaternion, is estimated in the state vector since the dual part can readily be calculated from the translation and the rotational real quaternion as given by

$$\hat{q} = q + \epsilon \frac{t}{2} q \tag{12}$$

where t is a vector quaternion formed from the translation vector and a zero scalar part. The first derivative is

$$\dot{\hat{q}} = \dot{q} + \epsilon (\frac{\dot{t}}{2} q + \frac{t}{2} \dot{q}). \tag{13}$$

With the linear velocity as state variables, the state update for the linear translation becomes very simple. The first derivative of the translation is also continuous for all real dynamic translations. Chou[14] gives the relation between quaternion angular velocity and the spatial angular velocity,

$$\Omega = \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = 2\dot{q}q^*. \tag{14}$$

Ω is a vector quaternion where the vector portion is the angular velocity about the axes. Solving for $\dot{q}$ gives

$$\dot{q} = \frac{1}{2}\Omega q. \tag{15}$$

Since the quaternion has four parameters to represent rotation, an additional degree of freedom is present. As a result, normalization of the quaternion to unit magnitude after each iteration is performed.

4.2. State Propagation and Measurement Update Model

The discrete EKF system model defines a function ϕ and additive noise $\vec{w}$ to calculate the next state in terms of the present state given by

$$\vec{s}_k = \phi_{k-1}(\vec{s}_{k-1}) + \vec{w}_{k-1}. \tag{16}$$

$\vec{w}$ is assumed to be additive Gaussian noise with zero mean and covariance matrix Q_{k-1}. The state transition function ϕ extrapolates from the state at time interval $k-1$ to the next state at time interval k. The linear and angular velocities are assumed constant so that $\omega_i(t_k) = \omega_i(t_{k-1})$ and $v_i(t_k) = v_i(t_{k-1})$.

The quaternion propagation in time is described by Eq. (15). The solution when all ω_i are constant is, after simplification,

$$q(t) = \left[\cos(\frac{\|\Omega\|(t-t_k)}{2})I + \frac{2}{\|\Omega\|}\sin(\frac{\|\Omega\|(t-t_k)}{2})\bar{\Omega}\right] q(t_k) = Q_{tran}q(t_k). \tag{17}$$

The complete state transition function is

$$\phi_k(\vec{s}) = [t_x + \tau v_x \quad t_y + \tau v_y \quad t_z + \tau v_z \quad Q_{tran}q(t_k) \quad v_x \quad v_y \quad v_z \quad \omega_x \quad \omega_y \quad \omega_z]^T. \tag{18}$$

The measurement update equation for the EKF is

$$\vec{z}_k = h_k(\vec{s}_k) + \vec{v}_k. \tag{19}$$

The measurement noise $\vec{v}_k$ is a zero mean, Gaussian sequence with covariance matrix R_k. h_k comprises the perspective projection functions given in Section 3.1. Since the dual quaternion operation transforms lines to lines, the given model features from the object are lines represented as dual vector quaternions. The measured $\vec{z}_k$ components are the line points of the 2–D image plane lines projected from the 3–D object lines.

4.3. Linearization of the State Transition Function

The linearized state transition matrix is computed as the partial derivative of the state transition function with respect to each state variable and is evaluated at time $t = t_k$. The time dependency requires that it be computed at each state update. Formally, Φ_k is defined as

$$\Phi_k(\tilde{x}_k(-)) = \left.\frac{\partial\phi_k(\vec{x})}{\partial\vec{x}}\right|_{\vec{x}=\tilde{x}_k(-)}. \tag{20}$$

Φ_k is a matrix of dimension $n \times n$ where n is the number of state variables. In the problem being addressed here, $n = 13$. This matrix is

$$\Phi_k = \begin{bmatrix} 1 & 0 & 0 & \cdots & \tau & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & \tau & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & \tau & 0 & 0 & 0 \\ 0 & 0 & 0 & & 0 & 0 & 0 & & & \\ \vdots & & & Q_{tran} & & & \vdots & & Q_\omega & \\ 0 & \cdots & & & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdots & & & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \cdots & & & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \cdots & & & & & \cdots & 1 & 0 & 0 \\ 0 & \cdots & & & & & \cdots & 0 & 1 & 0 \\ 0 & \cdots & & & & & \cdots & 0 & 0 & 1 \end{bmatrix}. \tag{21}$$

Q_{tran} is defined above in Eq. (17). Q_ω is defined as

$$Q_\omega = \left. \frac{\partial Q_{tran} q(t_k)}{\partial \vec{\omega}} \right|_{\vec{\omega}=\vec{\omega}_k} \tag{22}$$

where $\omega = [\omega_x \ \omega_y \ \omega_z]^T$. The partial derivative of $Q_{tran}q(t_k)$ with respect to each ω_i is

$$\begin{aligned} \frac{\partial Q_{tran} q(t_k)}{\partial \omega_i} &= \left(\frac{-\omega_i \tau}{2|\omega|} \sin(\frac{|\omega|\tau}{2}) + \left(\frac{-\omega_i \tau}{|\omega|^2} \cos(\frac{|\omega|\tau}{2}) - \frac{\omega_i}{|\omega|^3} \sin(\frac{|\omega|\tau}{2}) \right) \bar{\Omega} + \right. \\ &\quad \left. + \frac{1}{|\omega|} \sin(\frac{|\omega|\tau}{2}) \frac{\partial \bar{\Omega}}{\partial \omega_i} \right) q(t_k). \end{aligned} \tag{23}$$

4.4. Linearization of the Measurement Function

The linearized measurement matrix is the partial derivative of the measurement function with respect to each state variable and is evaluated at time $t = t_k$. Thus, it is time dependent and must be calculated at each measurement update step. Formally, H_k is defined as

$$H_k(\tilde{s}_k(-)) = \left. \frac{\partial h(\vec{s})}{\partial \vec{s}} \right|_{\vec{s}=\tilde{S}_k(-).} \tag{24}$$

H_k is a $2i$-by-n matrix where i is the number of measured lines and n is the number of states. For each measured line, the partial derivative of each line point coordinate is calculated

$$\frac{\partial x_{lp}}{\partial s_i} = \frac{\partial x_{lp}}{\partial m_x} \frac{\partial m_x}{\partial s_i} + \frac{\partial x_{lp}}{\partial m_y} \frac{\partial m_y}{\partial s_i} + \frac{\partial x_{lp}}{\partial m_z} \frac{\partial m_z}{\partial s_i} \tag{25}$$

and

$$\frac{\partial y_{lp}}{\partial s_i} = \frac{\partial y_{lp}}{\partial m_x} \frac{\partial m_x}{\partial s_i} + \frac{\partial y_{lp}}{\partial m_y} \frac{\partial m_y}{\partial s_i} + \frac{\partial y_{lp}}{\partial m_z} \frac{\partial m_z}{\partial s_i} \tag{26}$$

where s_i is a state variable. Expanding each line, $l + \epsilon m$, and solving for m gives

$$m = r m_m r^* + \frac{1}{2} r l_m r^* t^* + \frac{1}{2} t r l_m r^*. \tag{27}$$

The quaternion multiplications may be replaced by the corresponding matrix forms to give

$$m = \overset{+}{M}_r \overset{-}{M}_r^* \, m_m + \frac{1}{2} (\overset{+}{M}_t + \overset{-}{M}_{t^*}) \, \overset{+}{M}_r \overset{-}{M}_{r^*} \, l_m. \tag{28}$$

Let $R = \overset{+}{M}_r \overset{-}{M}_r^*$ and $M_{t+t^*} = \frac{1}{2}(\overset{+}{M}_t + \overset{-}{M}_{t^*})$. R has the standard 3-by-3 rotation matrix in the lower right submatrix. The partial derivatives of m with respect to each state variable are

$$\frac{\partial m}{\partial t_i} = \frac{\partial M_{t+t^*}}{\partial t_i} R l_m \tag{29}$$

where t_i is one of the three translation variables;

$$\frac{\partial m}{\partial q_i} = \frac{\partial R}{\partial q_i} m_m + M_{t+t^*} \frac{\partial R}{\partial q_i} l_m \tag{30}$$

where q_i is one of the four rotational quaternion variables;

$$\frac{\partial m}{\partial v_i} = \frac{\partial m}{\partial \omega_i} = 0 \tag{31}$$

where v_i and ω_i are the three linear velocities and three angular velocities, respectively. l_m and m_m are quaternions that compose the dual quaternion representation of the object model and are known from the geometric description of the model. The partial derivatives of the image points x_{lp} and y_{lp} with respect to each element of $\vec{m}$ are given by

$$\frac{\partial x_{lp}}{\partial m_x} = \frac{\lambda m_z}{(m_x^2 + m_y^2)} - 2\frac{\lambda m_x^2 m_z}{(m_x^2 + m_y^2)^2}, \qquad \frac{\partial y_{lp}}{\partial m_x} = -2\frac{\lambda m_x m_y m_z}{(m_x^2 + m_y^2)^2}, \tag{32}$$

$$\frac{\partial x_{lp}}{\partial m_y} = -2\frac{\lambda m_x m_y m_z}{(m_x^2 + m_y^2)^2}, \qquad \frac{\partial y_{lp}}{\partial m_y} = \frac{\lambda m_z}{(m_x^2 + m_y^2)} - 2\frac{\lambda m_y^2 m_z}{(m_x^2 + m_y^2)^2}, \tag{33}$$

$$\frac{\partial x_{lp}}{\partial m_z} = \frac{\lambda m_x}{(m_x^2 + m_y^2)}, \qquad \frac{\partial y_{lp}}{\partial m_z} = \frac{\lambda m_y}{(m_x^2 + m_y^2).} \tag{34}$$

4.5. Iterated Extended Kalman Filter Representation

The system and measurement models of the IEKF have been given above. An initial state estimate is required based on prior knowledge. An initial error covariance matrix, P_0, that is dependent on the prior knowledge must also be specified. Process and measurement noise covariance matrices, Q_k and R_k, respectively, are required. After each iteration, the derived partial derivatives with respect to each state variable are calculated to determine the linearized equations. With each updated state estimate, a new linearization is then performed about this state. Several iterations about the updated state may then be needed to achieve convergence.

5. EXPERIMENTAL RESULTS

This section provides an evaluation for the line-based method through both simulation and actual physical system testing. A dynamic vision application for the approach is defined where a camera provides the visual feedback for a robot performing a task. Simulation testing measures the accuracy of the estimation under an assumed noise distribution and magnitude. Speed of convergence, mean square error, and stability are presented. Actual test results measuring relative motion and position from a robot arm are also shown. Relative accuracy as well as noise characterization is provided.

5.1. Simulation Tests

Simulation testing was performed to evaluate the performance of the pose estimation method under a variety of conditions. The results are compared with corresponding results from a point-based extended Kalman filtering method using an identical target to illustrate the performance differences. An ideal camera model is used in the simulation. Perspective projection is assumed for the camera with a known effective focal length. Noise of an assumed magnitude and distribution is added to the image feature locations before processing. A target object consisting of four coplanar points in a rectangular pattern is simulated with individual feature points. Pairs of these points, when extracted from the image plane, are connected together to form lines.

Initial conditions requiring specification include the initial state, $\vec{s}_0$, and the error covariance matrix, P_0. The state vector may be considered a collection of Gaussian random variables with covariance P_0. The initial state is a sample taken from each random variable. Process noise given by the covariance matrix Q is also specified as an initial condition for the simulations, remaining constant throughout. Similarly, measurement noise given by the covariance matrix R is initially specified as a constant.

Table 1. Actual and assumed initial states of the EKF for the four-point simulation tests. The initial states are the same for both the dual quaternion method and the point method.

State	Translation $x\ y\ z$ (mm.)	Quaternion $q_0\ q_1\ q_2\ q_3$	Linear Velocity $x\ y\ z$ (mm./sec.)	Rotational Velocity $x\ y\ z$ (rad./sec.)
True Initial State	10 10 1000	1 0 0 0	-5 2 -5	-0.03 0.05 -0.2
Initial State Estimate	0 0 990	0.9998 0.01 0.01 0.01	0 0 0	0 0 0

5.1.1. Test Results

Simulation results give the calculated root mean square (RMS) error over time for the model trajectory. Table 1 shows the initial state conditions for the simulation. The initial states are the same for both methods. In all cases, the off-diagonal terms of the covariance matrices are zero. For these tests, the simulated sample interval is 0.1 second. The error covariance values were held constant over the entire measurement interval.

The input noise is a Gaussian with a standard deviation of 0.02 mm. This noise level corresponds to approximately 4% of the image size in the image. Based on this noise level, the measurement error covariance matrix has diagonal elements of 0.0004 mm.^{2} for each measured variable. Although the dynamic model is the same for both the dual quaternion method tests and the comparison method tests, tuning and stabilization adjustments were required for both methods. Nonzero process noise quantities were needed even though the dynamic model assumed no process uncertainty. Stability was a significant problem for the point reference method with this level of noise. Estimate errors occasionally became unbounded when no process noise was used. Stability was not as great a concern for the dual quaternion method since the errors did not become unbounded, reaching approximately 10% error in the z-translation parameter. Significant improvements in errors were achieved, however, by adjustment of these values.

Figures 3 through 5 show examples of the calculated RMS error over 100 sample runs for both the dual quaternion line method and the quaternion point method. In these state variables, the dual quaternion method has lower RMS error over the test time interval. The square root of the corresponding mean diagonal element from the calculated covariance error matrix is also shown in each figure. In these tests, the covariance matrix values correspond, generally, closely to the actual error values for the dual quaternion method while large differences are present for the point method.

5.2. Robot Arm Tests

Experimental testing has been performed using a robot arm equipped with an area camera and a computer system for processing images and controlling the arm. Tests were performed with controlled relative motion between an object and the camera.

5.2.1. Experimental Setup

The experimental setup consists of a Mitsubishi RV-E2 six-degree-of-freedom robot arm with a Cidtec camera used for image acquisition. The pose estimation is performed on a 166 Mhz. Pentium PC. The robot is controlled remotely via RS232 to the robot controller. The remote robot control consists of position, velocity, and acceleration commands.

5.2.2. Measurement Model

The measurement model for the experiment consists of the functions given in Fig. 6. First, the image of the target object is acquired by the camera and the frame grabber. Next, the object edges are extracted and the contour is formed. Straight lines are then calculated from the contour with length filtering performed to eliminate short lines that could arise from extraneous features. Collinear lines are also merged into one line. The resulting lines are used as inputs to the filter after conversion to the line point parameterization. Correspondence to the known geometric model is established at this stage.

In the experimental tests, an initial estimate of the relative pose between the camera and the object is provided by calculating pose using the four-point coplanar pose calculation method of Abidi and Chandra.[1] The initial velocities are set to zero as in the simulation tests.

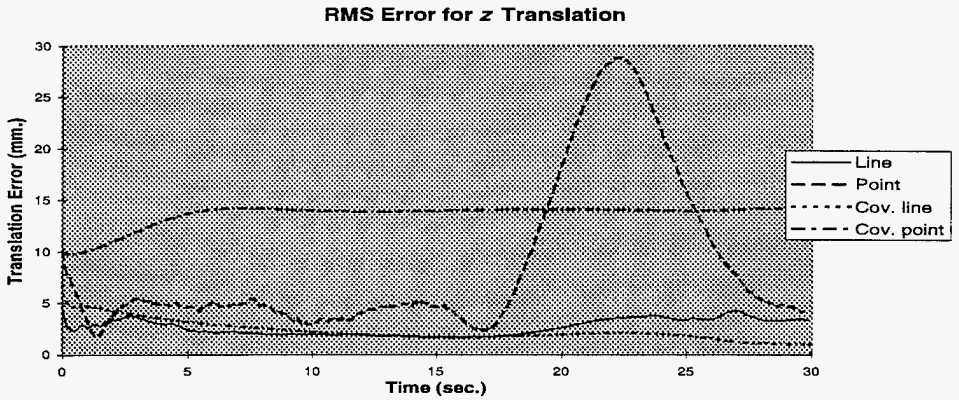

Figure 3. t_z translation pose estimate simulation results with a four-point target showing linear translation RMS estimation error over time for the line method and the comparison point method calculated from 100 sample runs. The square root of the predicted error covariance from the Kalman filter for both methods is also shown.

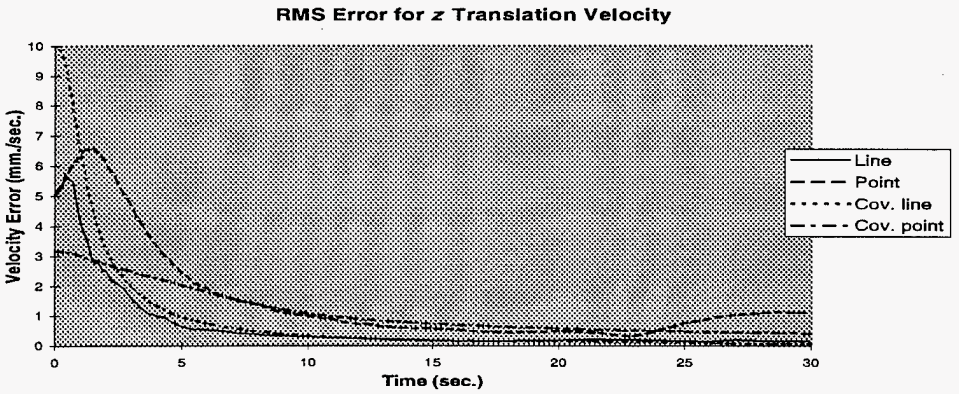

Figure 4. v_z translation velocity pose estimate simulation results with a four-point target showing linear translation RMS estimation velocity error over time for the line method and the comparison point method calculated from 100 sample runs. The square root of the predicted error covariance from the Kalman filter for both methods is also shown.

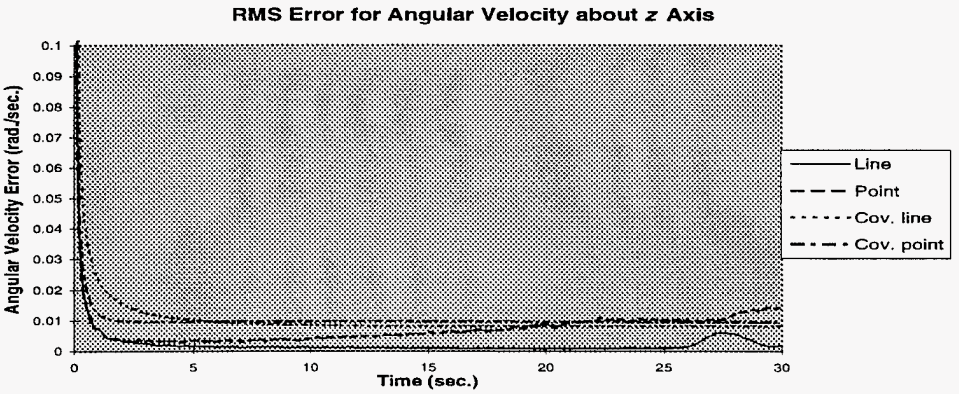

Figure 5. ω_z angular velocity pose estimate simulation results with a four-point target showing angular RMS estimation velocity error over time for the line method and the comparison point method calculated from 100 sample runs. The square root of the predicted error covariance from the Kalman filter for both methods is also shown.

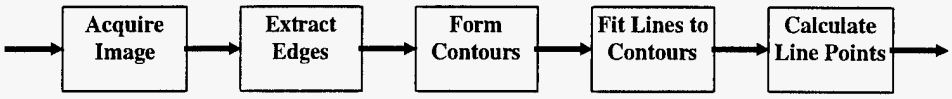

Figure 6. Measurement functions applied to obtain line point measurements from camera image in robot arm experiments.

5.2.3. Target Motion Results

The tests were performed with the target moving in a circular pattern around the z axis at constant speed. The camera and robot arm were directly above the target and stationary. The motion is confined to the x-y plane. Sample period is one second. Figures 7 through 10 show the estimated response for some selected state variables over time. The x and y translations vary in a sinusoidal pattern consistent with constant circular motion. The z-axis

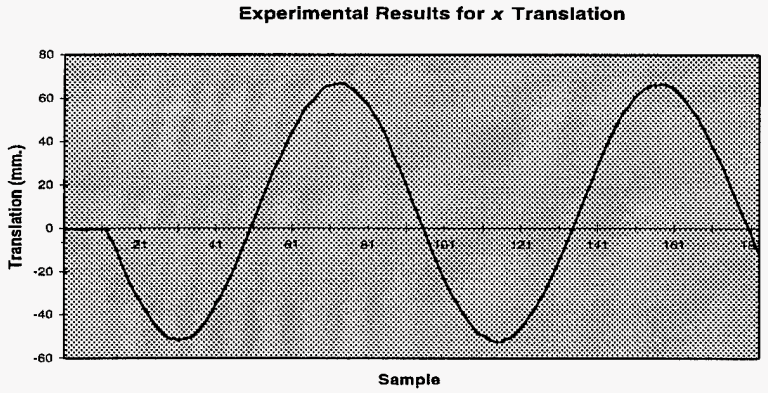

Figure 7. t_x translation pose estimate from experimental testing for constant target motion about the z axis. The estimate shows a sinusoidal variation corresponding to the true x variation of the target.

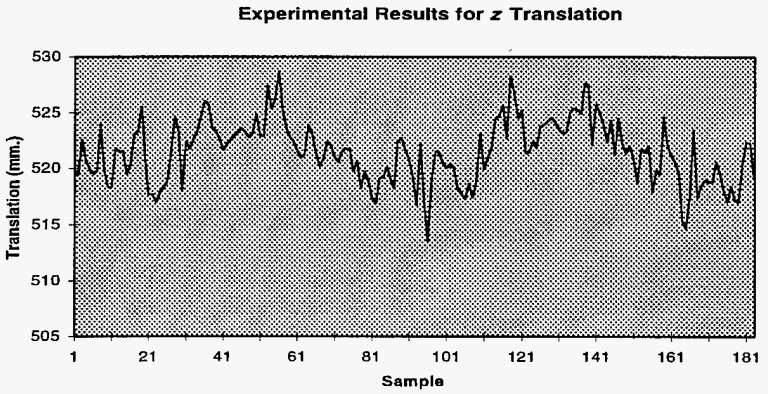

Figure 8. t_z translation pose estimate from experimental testing for constant target motion about z axis. The estimate shows a noisy variation about a mean z value with a small sinusoidal component. The true target z value is nominally constant.

position is seen to vary with a small sinusoidal component over the test interval along with some apparent noise. The motion was nominally in the x-y plane of the target and parallel to the image plane of the camera. Minimal out-of-plane motion results in the observed sinusoidal z variation with time. In this test, the z variation is seen to be about 5 mm. The quaternion estimates are also consistent with the actual motion. For motion about the z axis, only q_0 and q_3 should vary significantly in magnitude. The results for q_0 in Fig. 9 demonstrate this behavior. The z linear velocity estimate, however, shows significant noise, masking the true velocity component which is near zero. Angular velocity estimates show significant noise in the x and y components although the mean appears to remain

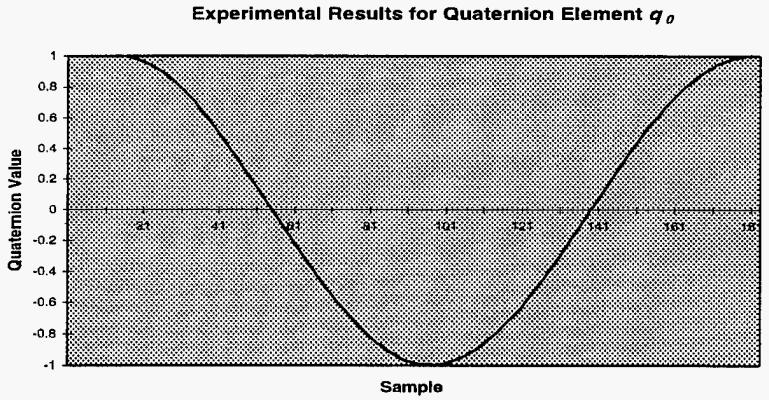

Figure 9. q_0 quaternion pose estimate from experimental testing for constant target motion about the z axis. The estimate shows a full 360 degree sinusoidal variation corresponding to the true q_0 variation of the target as it moves through two complete revolutions.

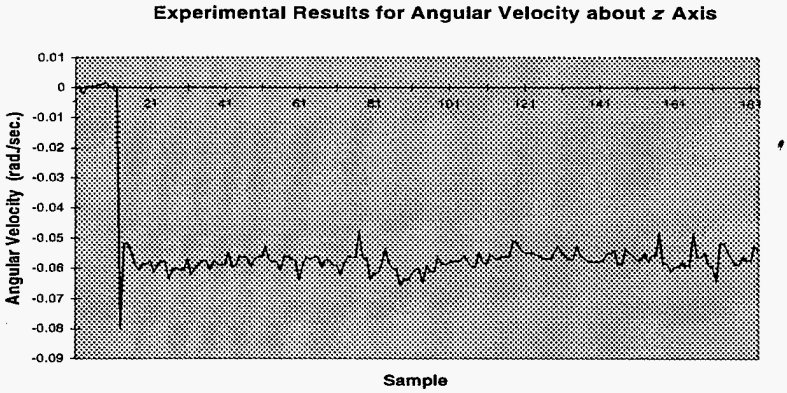

Figure 10. ω_z angular velocity pose estimate from experimental testing for constant target motion about z axis. The estimate shows a noisy variation about a true mean ω_z velocity value of -0.06 rad./sec.

near zero. The angular z velocity in Fig. 10, however, gives an accurate estimate of the true rotational speed. The 0.06 rad./sec. velocity corresponds to a revolution time of 105 seconds. The z angular velocity results also show the fast transient response of the initial startup from no motion to the constant rotational speed.

6. CONCLUSIONS

A new pose, motion, and structure estimation method has been developed. This method uses an imaging technique with a single area-based camera along with a reference object or scene to calculate estimates for relative six-degree-of-freedom position and orientation as well as the associated velocity estimates. Intended for real-time use, this method can be applied to robot vision, control, and assembly tasks. The system model for the method is based on line features, a dual quaternion parameterization for the 3–D transformation, and the IEKF. The IEKF is used as an estimator for the highly nonlinear imaging model.

The experimental results obtained from simulation and actual robot arm testing have shown that this method estimates 3–D pose parameters more accurately than previous point-based extended Kalman filtering within the central region of a camera view. In particular, the range and range motion estimation has been shown to be measured more accurately with faster settling times than with point methods of real-time estimation. The robot arm tests demonstrate that the method is accurate and stable.

REFERENCES

1. M. Abidi and T. Chandra, "A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**, pp. 534–538, May 1995.
2. M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM* **24**, pp. 381–395, June 1981.
3. M. Walker, L. Shao, and R. Volz, "Estimating 3–D location parameters using dual number quaternions," *CVGIP: Image Understanding* **54**, pp. 358–367, November 1991.
4. T. Phong, R. Horaud, A. Yassine, and D. Pham, "Optimal estimation of object pose from a single perspective view," in *International Conference on Computer Vision*, pp. 534–539, February 1993.
5. D. Lowe, *Perceptual Organization and Visual Recognition*, Kluwer, Boston, MA, 1985.
6. R. Kumar, "Determination of camera location and orientation," in *Proceedings: Image Understanding Workshop*, pp. 870–879, May 1989.
7. N. Ayache and O. Faugeras, "Building, registrating, and fusing noisy visual maps," *International Journal of Robotics Research* **7**, pp. 45–65, December 1988.
8. A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**, pp. 562–575, June 1995.
9. T. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive 3–D motion estimation from a monocular image sequence," *IEEE Transactions on Aerospace and Electronic Systems* **26**, pp. 639–656, July 1990.
10. S. Lee and Y. Kay, "A Kalman filter approach for accurate 3–D motion estimation from a sequence of stereo images," *CVGIP: Image Understanding* **54**, pp. 244–258, September 1991.
11. D. Westmore and W. Wilson, "Direct dynamic control of a robot using an end-point mounted camera and Kalman filter position estimation," in *Proceedings IEEE International Conference on Robotics and Automation*, pp. 2376–2384, April 1991.
12. J. Wang and W. Wilson, "3d relative position and orientation estimation using Kalman filter for robot control," in *Proceedings IEEE International Conference on Robotics and Automation*, pp. 2638–2645, May 1992.
13. B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America* **4**, pp. 629–642, April 1987.
14. J. Chou, "Quaternion kinematic and dynamic differential equations," *IEEE Transactions on Robotics and Automation* **8**, pp. 53–64, February 1992.
15. W. Clifford, "Preliminary sketch of bi-quaternions," *Proceedings of the London Mathematical Society* **4**, pp. 381–395, 1873.
16. K. Daniilidis and E. Bayro-Corrochano, "The dual quaternion approach to hand-eye calibration," in *Proceedings IEEE International Conference on Pattern Recognition*, p. A7E.6, 1996.
17. A. Gelb, ed., *Applied Optimal Estimation*, The MIT Press, Cambridge, Massachusetts, 1974.

Report Number (14) ORNL/CP--96358
CONF-980117--

Publ. Date (11) 199801
Sponsor Code (18) DOE, XF
JC Category (19) UC-900, DOE/ER

DOE