

Recursive 3-D Motion Estimation from a Monocular Image Sequence

T. J. BROIDA
Hughes Aircraft Co.

S. CHANDRASHEKHAR, Student, IEEE

R. CHELLAPPA, Senior Member, IEEE
Signal and Image Processing Institute

The problem considered here involves the design and application of a recursive algorithm to a sequence of images of a moving object to estimate both its structure and kinematics. The object is assumed to be rigid, and its motion is assumed to be "smooth" in the sense that it can be modeled by retaining an arbitrary number of terms in the appropriate Taylor series expansions. Translational motion involves a standard rectilinear model, while rotational motion is described with quaternions. Neglected terms of the Taylor series are modeled as process noise. A state-space model is constructed, incorporating both kinematic and structural states, and recursive techniques are used to estimate the state vector as a function of time.

A set of object match points is assumed to be available, consisting of fixed features on the object, the image plane coordinates of which have been extracted from successive images in the sequence. The measured data are the noisy image plane coordinates of this set (or of a subset of this set) of object match points, taken from each image in the sequence. High image plane noise levels (up to ~ 10 percent of the object image size) are allowed. The problem is formulated as a parameter estimation and tracking problem, which can use an arbitrarily large number of images in a sequence. The recursive estimation is done using an iterated extended Kalman filter (IEKF), initialized with the output of a batch algorithm run on the first few frames. Approximate Cramér-Rao lower bounds on the error covariance of the batch estimate are used as the initial state estimate error covariance of the IEKF. The performance of the recursive estimator is illustrated using both real and synthetic image sequences.

Manuscript received April 10, 1989; revised November 4, 1989 and January 3, 1990.

IEEE Log No. 36730.

This work was partially supported by the Office of Naval Research under the grant N00014-89-5-1598.

Authors' addresses: T. J. Broida, Hughes Aircraft Co., Radar Systems Group, Los Angeles, CA 90009; S. Chandrashekar and R. Chellappa, Signal and Image Processing Institute, Dep't. of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-0272.

0018-9251/90/0700-0639 \$1.00 © 1990 IEEE

I. INTRODUCTION

Two fundamentally distinct approaches exist for obtaining measurement data for motion estimation. Optical flow methods represent motion in the image plane as sampled, continuous velocity fields [1-3]. Work such as [4, 5] demonstrates the usefulness of this type of approach, in terms of the "low-level" problems of motion recognition and segmentation of scenes into their moving and stationary components, when the camera is itself moving.

Feature based methods rely on the recognition of the same set of correspondence points—points that arise from the same feature of the object—in two or more images. Then, various approaches can be taken towards estimation of object motion. The approach presented here is based on the use of discrete features.

A substantial amount of work has been devoted to methods for estimating object motion based on a short sequence of images. Many existing methods define object motion to be a single rigid body transformation that takes an object from its spatial position at the time of one image to its position at the time of a second image. This transformation can be represented by a single translational increment, followed or preceded by a rotational increment about a single axis. Since the time between images can be measured, the result is an approximation of translational and rotational velocities.

Much existing work begins by assuming that the problems of extracting feature points and establishing the required correspondences have been accomplished. Roach and Aggarwal [6] develop a system of nonlinear equations that relate the measured image plane coordinates, the (x, y, z) coordinates on the object, and the camera position parameters. Numerical techniques for solving the nonlinear equations are discussed, as well as methods for choosing good initial conditions and handling noise in the data.

Assuming that rotational increments are small, Tsai and Huang, in [7, 8], give an elegant analysis of the motion estimation problem that addresses both the computational and theoretical problems. In [7] it is shown that the eight "pure parameters" relating the spatial positions of the object match points in successive images are unique, given several constraints on the spatial relationships of the match points. Using a singular value decomposition of the "pure parameters" matrix, the conditions under which the motion parameter estimates are unique are discussed in [8].

Fang and Huang [9] have implemented the approach, and present detailed experimental results. They report successful experiments in extracting features and establishing match point correspondence when the rotation and scale change between successive images is small.

In [10], a modification to the pure projective geometry approach is made. This involves introducing a

regularization term so as to find the feasible rotation with minimum magnitude. Experiments conducted for more (up to 30) match points show that for large numbers of points (more than about 20) the pure projective geometry method works as well as the regularization approach is not improved significantly by the use of more points. An attempt to use more than two images in the sequence is also made, by averaging the results of each of the successive pairs. This is shown to give no significant improvement.

It is widely held that most existing schemes for motion estimation perform very poorly when the data (image coordinates of match points) are noisy [6, 9–11]. In [6, 7, 9] smoothing is achieved by using a larger number of match points in each image. This gives some improvement [9, 10], but as discussed in [12], using additional match points in each frame introduces additional unknown parameters, which limits the amount of new information that can be incorporated by this means. The alternative is to use a larger number of image frames. To use an arbitrary number of frames, the number of unknown model parameters should not be a function of the number of image frames used. The object motion model presented here has this property; models used in [6, 7, 9, 10] do not. The models used in [13, 14] are quite similar to those used in our research, but differ in certain ways: [13] assumes known structure, [14] uses specific models for different situations (e.g., number of feature points, number of image frames). In [15] an integrated spatio-temporal approach is developed, which can be tailored to specific applications such as automated land vehicle guidance. In [16] this rotational modeling is addressed by combining multiple solutions of the two-view problem using a local conservation of angular momentum (LCAM) model. However, the real image experiments reported there involve stereo imagery only.

The use of a larger number of images in the sequence allows significant smoothing to be achieved. In addition, the problem of forming parameter estimates in the presence of match point occlusion or temporary object occlusion can also be addressed in a natural manner, since the evolution of the model parameters with time is the basis of the general approach. Further, the recursive solution could be a starting point towards solving the problem of finding the same set of object match points in successive image frames, since match point locations and image coordinates can be extrapolated to the next image based on model parameter estimates.

Our basic approach has been to develop models for the motion of an object, the time-evolution of this motion, and the observation of the object, and to formulate the problem as one of recursive state estimation. Fig. 1 illustrates the basic models for motion, structure, and the observation of the object. First, the use of truncated Taylor series

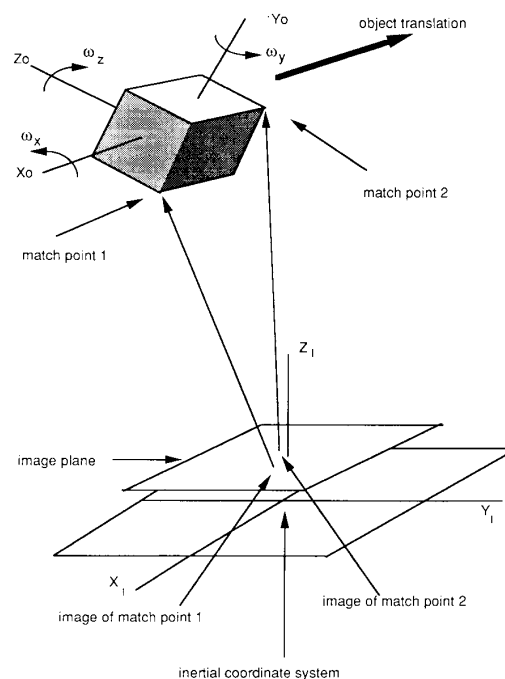


Fig. 1. Conceptual illustration of object motion and imaging.

allows the incorporation of as many derivatives as desired in modeling motion, both translational and rotational. The choice of standard rectilinear states for translational motion results in linear time-evolution. However, rotational motion is more complex, and unavoidably nonlinear. Fortunately, the use of quaternions allows at least a closed form propagation in time of rotational motion with either constant velocity or constant precessional rate.

Next, an object-centered coordinate system is defined. Translation is that of the origin of this coordinate system, and rotation is about the origin of this system. Object feature points are constant in this coordinate system, since the object is assumed to be rigid. The camera-centered coordinate system is assumed to be inertial (nonaccelerating). The inertial (camera-centered) coordinates of an object feature point are just the inertial coordinates of the origin of the object-centered system added to the object-system coordinates of the feature point. These concepts are made precise in the next section. Using the central projection imaging model, the image of each point is then simply the ratio of its inertial x and y coordinates to its z coordinate, multiplied by the focal length of the imaging system.

Finally, based on N frames of data, with noisy image coordinates of M features in each frame, we form estimates of the unknown parameters (elements of the state vector) in the above models. The plant equation is based on the motion model, and indicates how the parameters to be estimated evolve in time.

The measurement equation is based on the central projection model of image formation. The nonlinear nature of both equations precludes a simple Kalman filtering approach, and hence an approximate nonlinear filtering approach is used. An iterated extended Kalman filter (IEKF) [17] is designed for the problem, and its performance on real and synthetic images is studied. Experiments show that the IEKF performs very well when the initial guess supplied to it is obtained by running a batch estimation algorithm on the first few images in the sequence [18], and the approximate Cramér-Rao lower bounds [19] on the batch estimate are used as the error covariance of the initial guess.

Here, feature point correspondences are assumed to be available; a method for obtaining them is suggested in the Appendix, but it has not been implemented so far in our research. A similar method has been implemented by Dickmanns [20], who employs “constrained correlation” to obtain the match points.

Previous work by Broida and Chellappa in this area is discussed in [18, 21–23]. In [21] a one-dimensional (1-D) image of a two-dimensional object (2-D) undergoing 2-D motion was examined, to explore the properties of central projection imaging and the viability of the object/motion modeling approach. Some knowledge of object structure was assumed, and a recursive solution method was used on simulated data. The favorable results presented there were extended to a 2-D image of a 3-D object, undergoing 3-D motion, and the various models were more fully developed—this research was reported in a workshop [18]. A recursive solution was applied to simulated imagery involving pure translation and unknown structure. In [22], rotational motion was included, and a batch approach was applied to synthetic data; the experiments involved known object structure. In [23], the general case of unknown structure and motion (both translational and rotational) was addressed, and a batch method was shown to be effective in two experiments involving real imagery. Here we deal with the more general case of motion involving both translation and rotation, assuming no knowledge about the structure of the object undergoing motion.

II. MODELS

The fundamental model of this paper is that the motion of the rigid object during the observation period is smooth enough so that it can be represented by a dynamic model of relatively low dimensionality. The constraint imposed on the motion is that some finite time derivative (say, the n th) of the variation in each kinematic attribute be constant. That is, a constant first derivative implies constant velocity, constant second derivative implies constant

acceleration, etc. Further, the model allows a different value of n for rotation and translation.

A. Imaging Model

A central projection imaging model is assumed, defined by

$$h : S \mapsto P \quad (1)$$

where

$$s = (x, y, z)^T \in S = \{(x, y, z)^T \in R^3, z > 0\} \quad (2)$$

is a spatial point coordinate,

$$p = (X, Y)^T \in P \subset R^2 \quad (3)$$

is an image plane point coordinate. The space P is nominally a finite rectangle, corresponding to the image plane of a camera. Then,

$$X = f \cdot \frac{x}{z} + n_X; \quad Y = f \cdot \frac{y}{z} + n_Y \quad (4)$$

map spatial coordinates to noisy image coordinates, where the camera focal length f is set to unity w.l.o.g. The terms n_X and n_Y are the image plane noise components, discussed below. Thus, the measurement model for a single point $s = (x, y, z)^T$ is

$$p = \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} x/z \\ y/z \end{pmatrix} + \begin{pmatrix} n_X \\ n_Y \end{pmatrix} = h[s] + n. \quad (5)$$

The measured image coordinates of the match points are assumed to consist of the image coordinates of the true feature positions corrupted by additive independent zero mean Gaussian noise.

B. Object and Motion Model

An object-centered coordinate system is defined. The origin of this object-centered coordinate system is not observed, in general. Object structure is then defined as the coordinates of the object match points in the object-centered coordinate frame. These positions are constant in time due to the rigidity assumption. Object translational kinematics are defined to be the position and motion of the origin of the object-centered coordinate frame with respect to the camera-centered (inertial) coordinate frame. Object rotational kinematics are defined to be the object angular position and motion about the origin of the object-centered frame expressed in the inertial camera-centered coordinate system. Object structure and translational kinematics can only be known to within a global scale factor, unless absolute *a priori* data is available about the object and/or its translational kinematics. Object rotational kinematics are not subject to this scale factor. The scale factor comes about because, as can be seen from the mapping h defined in (5), any constant multiple of all spatial coordinates $(x, y, z)^T$ results in the same image.

The following model results. Let $s_{iO} = (x_i, y_i, z_i)^T$ be the object-centered coordinates of match point i . Let $s_R(t) = (x_R(t), y_R(t), z_R(t))^T$ be the camera-centered (inertial) coordinates of the origin of the translating object reference frame (not observed directly). Let $R(t)$ be the 3×3 coordinate transformation matrix that aligns the object coordinate axes with the camera coordinate axes, changing with time (rotating object). Let $s_i(t)$ be the spatial, camera-centered coordinates of match point i at time t .

The object motion model is then given by

$$s_i(t) = s_R(t) + R(t)s_{iO}. \quad (6)$$

At time t_k the image plane measurements of the match points are, from (5),

$$p_i(t_k) = h[s_i(t_k)] + n(t_k) \quad (7)$$

which can be written as

$$\begin{aligned} p_i(t_k) &= (X_i, Y_i)_k^T \\ &= h[s_R(t_k) + R(t_k)s_{iO}] + n(t_k) \end{aligned} \quad (8)$$

where $i = 1, \dots, M$ for M object match points. This cannot be simplified further, because h is nonlinear.

C. Translational Motion Model

Since the spatial coordinates of the object-centered reference frame $s_R(t)$ can be written in terms of an arbitrary number of derivatives, a variety of modeling options are available. Assuming it can be accurately modeled by a constant n th derivative,

$$s_R(t) = s_R(t_0) + \sum_{k=1}^n \frac{\partial^{(k)} s_R(t)}{\partial t^{(k)}} \bigg|_{t=t_0} \frac{(t-t_0)^k}{k!}. \quad (9)$$

Thus, the translational motion during the observation period is modeled by a finite number ($3n$) of parameters, which are simply the nonzero derivatives at a single point in time.

D. Rotational Motion Model

Quaternions, described for example in [24–26], can be used to propagate the transformation matrix $R(t)$ in time, with the rotation of the object coordinate frame represented by the object rotation rates about its (x, y, z) axes, $\omega_t = (\omega_x, \omega_y, \omega_z)_t$. With this approach, the rotation matrix $R(t)$ can be written in terms of the unit quaternion $\mathbf{q}(t) = (q_1(t), q_2(t), q_3(t), q_4(t))^T$ as discussed in [24–26]. Suppressing the time dependency,

$R =$

$$\begin{pmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{pmatrix}$$

or, more briefly,

$$R(t) = R[\mathbf{q}(t)]. \quad (10)$$

The unit quaternion $\mathbf{q} = (q_1, q_2, q_3, q_4)^T$ is related to “standard” expressions of the angular relation between coordinate systems by the relation

$$(q_1, q_2, q_3, q_4)^T = (n_1 \sin \theta/2, n_2 \sin \theta/2, n_3 \sin \theta/2, \cos \theta/2)^T \quad (11)$$

where (n_1, n_2, n_3) is the axis of rotation, and θ is the angle about the axis that aligns the coordinate axes of the rotating coordinate system with those of the reference system. It is termed a unit quaternion because $|\mathbf{q}| = 1$. The quaternion \mathbf{q} propagates in time according to the differential equation [25, 26]

$$\dot{\mathbf{q}}(t) = \Omega(\omega_t)\mathbf{q}(t); \quad \mathbf{q}(t_0) = \mathbf{q}_0 \quad (12)$$

where

$$\Omega(\omega_t) = \frac{1}{2} \begin{pmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{pmatrix}_t. \quad (13)$$

The solution to (12) when ω is constant is simply

$$\mathbf{q}(t) = \exp[\Omega \cdot (t - t_0)]\mathbf{q}(t_0). \quad (14)$$

It also might be noted that the matrix $2i\Omega/|\omega|$ is unitary, where $i = \sqrt{-1}$. As a result, the power series expansion for the matrix exponential can be reduced to [26]

$$\mathbf{q}(t) = \left[\cos(|\omega|(t - t_0)/2)I_4 + \frac{2}{|\omega|} \sin(|\omega|(t - t_0)/2)\Omega \right] \mathbf{q}(t_0) \quad (15)$$

which can be further reduced by assuming that the coordinate systems are aligned at $t_0 = 0$, so that $\mathbf{q}(0) = (0, 0, 0, 1)^T$, and

$$\mathbf{q}(t) = \left(\frac{\omega_x}{|\omega|} \sin(|\omega|t/2), \frac{\omega_y}{|\omega|} \sin(|\omega|t/2), \frac{\omega_z}{|\omega|} \sin(|\omega|t/2), \cos(|\omega|t/2) \right)^T.$$

However, this solution is valid only when ω is constant. In general, the image of the match points at time t_k can be written, from (8),

$$p_i(t_k) = h[s_R(t_k) + R[\mathbf{q}(\omega(t_k), t_k)]s_{iO}] + n(t_k). \quad (16)$$

The reason for resorting to quaternions is that the differential equation of (12) describing their time-propagation is much simpler than the analogous system for propagating Euler angles, even when higher derivatives are involved. In addition, the body rates ω are more meaningful, intuitively, than are the Euler rates, since the latter are defined about nonorthogonal axes. As discussed in [27], motion involving constant precessional rate can also be expressed in closed form

with quaternions. In order to represent higher order rotational derivatives, ω could be expanded in Taylor series as in (9), and the differential equation of (12) could be integrated numerically, with time-varying ω . However, Cramér-Rao lower bound results [19] indicate that very large amounts of accurate data would be required for such higher order rotational motion.

III. FORMULATION FOR RECURSIVE SOLUTION

The use of a recursive procedure for parameter estimation is not viewed as an alternative to the batch procedure. Instead, an estimation "system" is envisioned, in which a batch method is applied to an initial set of data, that is used to initialize a recursive procedure. The recursive algorithm is then used to "track" the object and its motion. The value of this type of approach is that recursive methods usually require much less computation time for each new set of data (each new image, for example), they continually compute state estimates indexed to the current time, based on all past data, and can readily extrapolate the state estimates ahead in time to aid in preprocessing the next set of data.

Another aspect of recursive filtering, in this context, is that approximate filters are required, due to the nonlinear nature of both the plant and the measurement processes. On the one hand, this means that the state estimates are not sufficient statistics for the past data, as they are in the linear case, so there is an inefficiency in the use of data, with degradation of state estimates. On the other hand, if the models are only approximations to the true plant and measurement processes, then the "limited memory" of this type of recursion can be advantageous, since "small" deviations from the models are tolerated, and tracked, without the need for explicit modeling. Thus, if a constant velocity model is used for the object motion, but there is in reality some deviation from this model (trajectory perturbations, small random accelerations, small constant accelerations, etc.), then a recursive algorithm will track the actual object kinematics, and "forget" about data taken at earlier times. A batch method would give the best "straight-line" (constant velocity) estimate for all of the data, and would not preferentially weight the more recent data. Of course, large deviations from the models can also lead to filter divergence and serious estimate biases. The notions couched here in terms of small and large are made somewhat more precise in [17], in terms of a comparison of the severity of nonlinearities and modeling errors with the process noise and measurement noise statistics. Essentially, if errors of any sort are on the order of the corresponding noise covariance, they are not "severe," since their effect is quite literally "lost in the noise."

The general problem can be posed for solution by a recursive algorithm, by separating the statement of

the problem into a plant model (continuous time) and a measurement model (discrete time). The problem statement for a recursive solution is then given by [17, 28]

$$\dot{s}_t = f(s) + G_t w_t \quad (17)$$

as the plant model, and

$$p(t_k) = h[s(t_k)] + v(t_k) \quad (18)$$

as the measurement model. The continuous time state vector is s , of dimension $d \times 1$; w_t is a vector, $r \times 1$, of temporally white Gaussian random processes, $w_t \sim \mathcal{N}(0, Q_t)$. G_t is a deterministic $d \times r$ matrix mapping the process noise into state space. The measurement $p(t_k)$ is a vector of dimension m , consisting of the nonlinear mapping $h[\cdot]$ from d -dimensional state space to m -dimensional observation space, corrupted by additive, zero-mean, Gaussian noise, temporally white, $v(t_k) \sim \mathcal{N}(0, R_k)$.

The statement of (17) is heuristic, since a rigorous treatment of the continuous time white noise w_t requires the stochastic differential equation [17]

$$ds = f(s) dt + G_t d\beta_t \quad (19)$$

where β_t is a vector of Wiener processes. In this paper, we have dealt with the case of motion involving constant translational and rotational velocities. With this assumption, and assuming M object feature points, the following set of states is chosen. Let $s(t)$ consist of the scalar elements

$$s(t) = \begin{pmatrix} x_R(t)/z_R(t) \\ y_R(t)/z_R(t) \\ \dot{x}/z_R(t) \\ \dot{y}/z_R(t) \\ \dot{z}/z_R(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \\ \omega_x \\ \omega_y \\ \omega_z \\ x_1/z_R(t) \\ y_1/z_R(t) \\ z_1/z_R(t) \\ \vdots \\ x_M/z_R(t) \\ y_M/z_R(t) \\ z_M/z_R(t) \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \\ s_{10} \\ s_{11} \\ s_{12} \\ s_{13} \\ s_{14} \\ s_{15} \\ \vdots \\ s_{3M+10} \\ s_{3M+11} \\ s_{3M+12} \end{pmatrix}_t \quad (20)$$

The origin of the object-centered coordinate system is

expressed as

$$s_R(t) = (x_R(t), y_R(t), z_R(t))^T.$$

The scalar terms x_i, y_i, z_i are the coordinates of the i th feature point in the object-centered system. The time derivative of $s(t)$ is

$$\dot{s}(t) = \begin{pmatrix} \dot{x}/z_R(t) - [x_R(t)/z_R(t)][\dot{z}/z_R(t)] \\ \dot{y}/z_R(t) - [y_R(t)/z_R(t)][\dot{z}/z_R(t)] \\ -[\dot{x}/z_R(t)][\dot{z}/z_R(t)] \\ -[\dot{y}/z_R(t)][\dot{z}/z_R(t)] \\ -[\dot{z}/z_R(t)]^2 \\ 0.5(\omega_z q_2 - \omega_y q_3 + \omega_x q_4) \\ 0.5(-\omega_z q_1 + \omega_x q_3 + \omega_y q_4) \\ 0.5(\omega_y q_1 - \omega_x q_2 + \omega_z q_4) \\ 0.5(-\omega_x q_1 - \omega_y q_2 + \omega_z q_3) \\ 0 \\ 0 \\ 0 \\ -[x_1/z_R(t)][\dot{z}/z_R(t)] \\ -[y_1/z_R(t)][\dot{z}/z_R(t)] \\ -[z_1/z_R(t)][\dot{z}/z_R(t)] \\ \vdots \\ -[x_M/z_R(t)][\dot{z}/z_R(t)] \\ -[y_M/z_R(t)][\dot{z}/z_R(t)] \\ -[z_M/z_R(t)][\dot{z}/z_R(t)] \end{pmatrix} = \begin{pmatrix} s_3 - s_1 s_5 \\ s_4 - s_2 s_5 \\ -s_3 s_5 \\ -s_4 s_5 \\ -s_5^2 \\ 0.5(s_{12} s_7 - s_{11} s_8 + s_{10} s_9) \\ 0.5(-s_{12} s_6 + s_{10} s_8 + s_{11} s_9) \\ 0.5(s_{11} s_6 - s_{10} s_7 + s_{12} s_9) \\ 0.5(-s_{10} s_6 - s_{11} s_7 - s_{12} s_8) \\ 0 \\ 0 \\ 0 \\ -s_{13} s_5 \\ -s_{14} s_5 \\ -s_{15} s_5 \\ \vdots \\ -s_{3M+10} s_5 \\ -s_{3M+11} s_5 \\ -s_{3M+12} s_5 \end{pmatrix} \quad (21)$$

The plant equation is bilinear in the states, involving no trigonometric functions, which is one advantage resulting from the use of quaternions here.

The differential equation of (17) includes a process noise vector $G_i w_i$, which represents either random or unmodeled deterministic deviations from the given plant model. A precise characterization of the process noise vector should be addressed in future work, but for practical purposes it can be treated as a set of "tuning parameters" [28], to be manipulated to account for mismodeling and to keep the filter gains high enough to track the object motion.

This choice of states has the effect that at any time t , both the motion and the structure of the object are referenced to the image plane. In the paper on batch estimation [23] it is shown that normalizing the states at the selected reference time to eliminate a free scale factor is necessary. If not normalized, the search algorithm simply diverges along this extra degree of freedom, with somewhat unpredictable results. The same phenomenon occurs in the recursive approach; however, since each iteration of the recursive algorithm can be viewed as a separate search step, repeated normalization is necessary, and this takes the form of dividing all affected states by $z_R(t)$, which acts as a continuous normalizing factor. In addition, there is another degree of freedom in the system due to the fact that the constant velocity rotation constrains the origin of the object-centered coordinate system only to lie on a line (the axis of rotation); its position on the axis is arbitrary. In our implementation, this is handled by treating the z coordinate of the first feature point as a constant equal to zero. This issue is discussed in detail in [19].

One effect of this choice of states is that initializing the state estimates becomes straightforward. The states $s_1(t)$ and $s_2(t)$ are in fact the normalized image plane coordinates of the origin of the object-centered coordinate frame at time t , as would be estimated by an initial batch procedure [19]. The observed image plane match point coordinates are perturbations to this location. The vector-valued measurement function ($2M \times 1$) of (18) is then, from (5),

$$p(t_k) = h[s(t_k)] + v(t_k) = \begin{pmatrix} \frac{s_1(t_k) + R_x(s, 1, t_k)}{1 + R_z(s, 1, t_k)} \\ \frac{s_2(t_k) + R_y(s, 1, t_k)}{1 + R_z(s, 1, t_k)} \\ \frac{s_1(t_k) + R_x(s, 2, t_k)}{1 + R_z(s, 2, t_k)} \\ \frac{s_2(t_k) + R_y(s, 2, t_k)}{1 + R_z(s, 2, t_k)} \\ \vdots \\ \frac{s_1(t_k) + R_x(s, M, t_k)}{1 + R_z(s, M, t_k)} \\ \frac{s_2(t_k) + R_y(s, M, t_k)}{1 + R_z(s, M, t_k)} \end{pmatrix}_{t_k} + v(t_k) = \begin{pmatrix} X_1(t_k) \\ Y_1(t_k) \\ X_2(t_k) \\ Y_2(t_k) \\ \vdots \\ X_M(t_k) \\ Y_M(t_k) \end{pmatrix} + v(t_k). \quad (22)$$

The abbreviations represent components of matrix-vector products, for example the scalar terms $R_x(s, i, t_k)$ refers to the x -component of the product of the rotation matrix with the normalized (x, y, z) coordinates of the i th match point, $R(s, t_k) \cdot (s_{3i+10}, s_{3i+11}, s_{3i+12})^T$. For example, denoting the rs component of $R(s, t_k)$ as R_{rs} ,

$$\begin{aligned} R_x(s, 1, t_k) &= R_{11}s_{13} + R_{12}s_{14} + R_{13}s_{15} \\ &= (q_1^2 - q_2^2 - q_3^2 + q_4^2)x_1/z_R(t) \\ &\quad + 2(q_1q_2 - q_3q_4)y_1/z_R(t) \\ &\quad + 2(q_1q_3 + q_2q_4)z_1/z_R(t). \end{aligned} \quad (23)$$

The 1s in the denominators of (22) can be written as $z_R(t)/z_R(t)$, corresponding to the terms $x_R(t)/z_R(t)$ and $y_R(t)/z_R(t)$ that are states $s_1(t)$ and $s_2(t)$; the numerator and denominator of each component of h are then homogeneous in $1/z_R(t)$. This problem formulation is appropriate for solution with a filter such as an IEKF. The mathematical details of the IEKF used in this research are discussed in Section IV.

IV. APPROXIMATE NONLINEAR FILTERS

A brief description of the linear Kalman filter is given first, to establish notations and to motivate some of the approximations used in the nonlinear cases. The plant and measurement models for the linear Kalman filter are as above, in (17) and (18), with the vector valued plant function $f(\cdot)$ (of dimension $d \times 1$) being reduced to a $d \times d$ matrix F , and the measurement function $h[\cdot]$ of dimension $m \times 1$ being reduced to a matrix $H(k)$ of dimension $m \times d$. The continuous time plant (or signal) model evolves in time according to

$$\dot{\mathbf{x}}(t) = F\mathbf{x}(t) + G_t\mathbf{w}_t. \quad (24)$$

The discrete plant model is then given by

$$\mathbf{x}(k+1) = \Phi_{k+1,k}\mathbf{x}(k) + G_k\mathbf{w}_k \quad (25)$$

where $\Phi_{k+1,k} = \exp[(t_{k+1} - t_k)F]$, and the discrete measurement model is

$$\mathbf{z}(k) = H(k)\mathbf{x}(k) + \mathbf{v}(k) \quad (26)$$

where the noise terms \mathbf{w} and \mathbf{v} are zero-mean Gaussian random vectors with covariances $\text{cov}(\mathbf{v}_k) = R_k$, and $\text{cov}(\mathbf{w}_k) = Q_k$.

Following [17], the estimate $\hat{\mathbf{x}}(k+1|k)$ denotes the predicted (extrapolated) estimate, just after a time update, while the estimate $\hat{\mathbf{x}}(k|k)$ denotes the smoothed (filtered) estimate, just after a measurement update. The linear Kalman filter measurement update equation is

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + K(k)[\mathbf{z}(k) - H(k)\hat{\mathbf{x}}(k|k-1)] \quad (27)$$

where the measurement is $\mathbf{z}(k)$. The gain sequence is computed as

$$K(k) = P(k|k-1)H(k)^T \times [H(k)P(k|k-1)H(k)^T + R_k]^{-1}. \quad (28)$$

The error covariance matrix of the predicted state estimates $P(k|k-1)$ is computed as

$$P(k|k-1) = \Phi_{k,k-1}P(k-1|k-1)\Phi_{k,k-1}^T + G_kQ_kG_k^T. \quad (29)$$

The smoothed covariance matrix is

$$P(k|k) = [I - K(k)H(k)]P(k|k-1) \quad (30)$$

and the time update for the state estimate is

$$\hat{\mathbf{x}}(k+1|k) = \Phi_{k+1,k}\hat{\mathbf{x}}(k|k). \quad (31)$$

The iteration is initialized with

$$P(0|0) = P_0 = E\{(\mathbf{x}(0) - \mu_{x0})(\mathbf{x}(0) - \mu_{x0})^T\} \quad (32)$$

and

$$\hat{\mathbf{x}}(0|0) = E\{\mathbf{x}(0)\} = \mu_{x0} = \hat{\mathbf{x}}(0). \quad (33)$$

In the case of the extended Kalman filter (EKF), the measurement update equation is simply

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + K(k)[\mathbf{z}(k) - \mathbf{h}[\hat{\mathbf{x}}(k|k-1)]] \quad (34)$$

where $\mathbf{h}(\mathbf{x})$ reverts to $H\mathbf{x}$ in the linear case. If the measurement function is nonlinear, define the linearized measurement functions as the $2M \times d$ matrix

$$H^x(k) = \left. \frac{\partial \mathbf{h}[\mathbf{x}]}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)}. \quad (35)$$

In our case, with the state vector \mathbf{s} as in (20), $\mathbf{h}[\mathbf{s}]$ (after dropping the indices \mathbf{s} , t_k and k) is given by

$$\mathbf{h}[\mathbf{s}] = \begin{pmatrix} \frac{s_1 + R_x(1)}{1 + R_z(1)} \\ \frac{s_2 + R_y(1)}{1 + R_z(1)} \\ \frac{s_1 + R_x(2)}{1 + R_z(2)} \\ \frac{s_2 + R_y(2)}{1 + R_z(2)} \\ \vdots \\ \frac{s_1 + R_x(M)}{1 + R_z(M)} \\ \frac{s_2 + R_y(M)}{1 + R_z(M)} \end{pmatrix} \quad (36)$$

where s_1 and s_2 are the first two components of \mathbf{s} , and R_x , R_y , etc. are expanded as in (23). We can then

write

$$H(s) = H^s(k) = \begin{pmatrix} P_1 & 0 & W_1 & 0 & S_1 & 0 & 0 & \dots & 0 \\ P_2 & 0 & W_2 & 0 & 0 & S_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_M & 0 & W_M & 0 & 0 & 0 & 0 & \dots & S_M \end{pmatrix} \quad (37)$$

where

$$P_i = \begin{pmatrix} \frac{1}{1+R_z(i)} & 0 \\ 0 & \frac{1}{1+R_z(i)} \end{pmatrix}, \quad 1 \leq i \leq M \quad (38)$$

the W_i submatrices are of dimension 2×4 with elements

$$W_i(1, k) = \frac{(1+R_z(i)) \frac{\partial R_x(i)}{\partial q_k} - (s_1 + R_x(i)) \frac{\partial R_z(i)}{\partial q_k}}{(1+R_z(i))^2}, \quad 1 \leq k \leq 4 \quad (39)$$

$$W_i(2, k) = \frac{(1+R_z(i)) \frac{\partial R_y(i)}{\partial q_k} - (s_2 + R_y(i)) \frac{\partial R_z(i)}{\partial q_k}}{(1+R_z(i))^2}, \quad 1 \leq k \leq 4 \quad (40)$$

and the S_i submatrices are of size 2×3 with elements

$$S_i(1, 1) = \frac{(1+R_z(i))R_{11} - (s_1 + R_x(i))R_{31}}{(1+R_z(i))^2} \quad (41)$$

$$S_i(1, 2) = \frac{(1+R_z(i))R_{12} - (s_1 + R_x(i))R_{32}}{(1+R_z(i))^2} \quad (42)$$

$$S_i(1, 3) = \frac{(1+R_z(i))R_{13} - (s_1 + R_x(i))R_{33}}{(1+R_z(i))^2} \quad (43)$$

$$S_i(2, 1) = \frac{(1+R_z(i))R_{21} - (s_2 + R_y(i))R_{31}}{(1+R_z(i))^2} \quad (44)$$

$$S_i(2, 2) = \frac{(1+R_z(i))R_{22} - (s_2 + R_y(i))R_{32}}{(1+R_z(i))^2} \quad (45)$$

$$S_i(2, 3) = \frac{(1+R_z(i))R_{23} - (s_2 + R_y(i))R_{33}}{(1+R_z(i))^2} \quad (46)$$

The gain for the EKF is then computed as

$$K(k) = P(k | k-1) H^s(k)^T \times [H^s(k) P(k | k-1) H^s(k)^T + R_k]^{-1} \quad (47)$$

If $h[\cdot]$ is "close" to linear in the states, and state estimate errors are not too large, the EKF can be reasonably effective. However, if $h[\cdot]$ is highly nonlinear, the EKF may diverge. Similarly, if the errors in the state estimate are large, the effect

of the nonlinearity becomes more severe, and divergence is likely. The IEKF is a slightly more sophisticated approximation wherein the mode is used as an approximation to the mean of the posterior density given the measurements, during an iterated measurement update. This iteration is the key feature of the IEKF.

The measurement function $h[\cdot]$ (ratios of functions of states) is highly nonlinear, which requires the iteration of the measurement function, as follows. As discussed above, the measurement is of dimension $2M$, consisting of the X and Y image plane coordinates of M feature points. The iterated measurement update equation is

$$\begin{aligned} \hat{x}(k | k)_{n+1} &= \hat{x}(k | k-1) + K(k)_n \\ &\times \left[z(k) - h[\hat{x}(k | k)_n] - H^s(k)_n \right. \\ &\quad \left. \times \{ \hat{x}(k | k-1) - \hat{x}(k | k)_n \} \right] \quad (48) \end{aligned}$$

(n is the index for the local iteration, and k is the time index) where the iteration is started with

$$\hat{x}(k | k)_0 = \hat{x}(k | k-1). \quad (49)$$

In our implementation, we deviate slightly from the traditional IEKF by including a quaternion normalization step immediately after the measurement update. This is done to keep the norm of the quaternion vector equal to unity. An analysis of the effects of such a step on the performance of a similar recursive estimator is done in [29], wherein the authors conclude that "the estimation errors are not affected by the normalization operation". This normalization is a result of an extra degree of freedom. Rotational motion involves three parameters ω_x , ω_y , and ω_z , while a quaternion has four elements. A "continuous normalization" can be achieved by dividing through by a fourth (cosine) term; this results in a representation of rotational motion known as Gibbs parameters [26]. Unfortunately, Gibbs parameters do not evolve in time in a way conducive to simple modeling such as (12) and (13).

The gain for the IEKF is included in the iteration as

$$K(k)_{n+1} = \hat{P}(k | k-1) H^s(k)_n^T \times [H^s(k)_n \hat{P}(k | k-1) H^s(k)_n^T + R_k]^{-1} \quad (50)$$

and the approximate measurement function is re-evaluated at

$$H^s(k)_n = \left. \frac{\partial h[x]}{\partial x} \right|_{x=\hat{x}(k|k)_n} \quad (51)$$

Finally, after the iteration is found to yield no further improvement (successive iterations differ by ϵ , or the maximum number of iterations (from 3 to 5 in

these cases) is exceeded), the approximate smoothed covariance is computed as

$$\hat{P}(k | k)_n = [I - K(k)_n H^x(k)_n] \hat{P}(k | k - 1). \quad (52)$$

The states are propagated in time by numerical integration of the first order system defining the plant. That is,

$$\hat{\mathbf{x}}(t_{k+1} | t_k) = \int_{t_k}^{t_{k+1}} \hat{\mathbf{x}}(\tau | t_k) d\tau + \hat{\mathbf{x}}(t_k | t_k). \quad (53)$$

The covariance may be propagated in time by computing an approximate state transition matrix, such as

$$F(\mathbf{x}_t) = \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}_t) \quad (54)$$

and by integrating

$$\begin{aligned} \hat{P}(t | t_k) &= F(\hat{\mathbf{x}}(t | t_k)) \hat{P}(t | t_k) \\ &+ \hat{P}(t | t_k) F(\hat{\mathbf{x}}(t | t_k))^T + G_t Q_t G_t^T \end{aligned} \quad (55)$$

as

$$\hat{P}(t_{k+1} | t_k) = \int_{t_k}^{t_{k+1}} \hat{P}(t | t_k) dt + \hat{P}(t_k | t_k). \quad (56)$$

The term $G_t Q_t G_t^T$ is the covariance of the vector $G_t \mathbf{w}_t$ given above.

For our problem, with \mathbf{s} as in (20) and $\mathbf{f}(\mathbf{s})$ as in (21), we have

$$F(\mathbf{s}) = \begin{pmatrix} F_1 & 0 & 0 \\ 0 & F_2 & 0 \\ F_3 & 0 & F_4 \end{pmatrix} \quad (57)$$

$$F_1 = \begin{pmatrix} -s_5 & 0 & 1 & 0 & -s_1 \\ 0 & -s_5 & 0 & 1 & -s_2 \\ 0 & 0 & -s_5 & 0 & -s_3 \\ 0 & 0 & 0 & -s_5 & -s_4 \\ 0 & 0 & 0 & 0 & -2s_5 \end{pmatrix} \quad (58)$$

$$F_2 = 0.5 \begin{pmatrix} 0 & s_{12} & -s_{11} & s_{10} & s_9 & -s_8 & s_7 \\ -s_{12} & 0 & s_{10} & s_{11} & s_8 & s_9 & -s_6 \\ s_{11} & -s_{10} & 0 & s_{12} & -s_7 & s_6 & s_9 \\ -s_{10} & -s_{11} & -s_{12} & 0 & -s_6 & -s_7 & -s_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(59)

$$F_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & -s_{13} \\ 0 & 0 & 0 & 0 & -s_{14} \\ 0 & 0 & 0 & 0 & -s_{15} \\ 0 & 0 & 0 & 0 & -s_{16} \\ 0 & 0 & 0 & 0 & -s_{17} \\ 0 & 0 & 0 & 0 & -s_{18} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & -s_{3M+12} \end{pmatrix} \quad (60)$$

$$F_4 = \begin{pmatrix} -s_5 & 0 & 0 & 0 & \cdots & 0 \\ 0 & -s_5 & 0 & 0 & \cdots & 0 \\ 0 & 0 & -s_5 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & & -s_5 \end{pmatrix}. \quad (61)$$

V. EXPERIMENTAL RESULTS

The performance of the recursive algorithm was tested on simulated as well as real image sequences. The algorithm is the same for both cases, but the performance analysis is done differently because the "ground truth" is known only for the experiments using simulated data. The next two sections discuss the set-up and the results of experiments using simulated and real data. This is followed by a section describing the details of the implementation of the IEKF (such as parameter selection) and related numerical issues.

A. Experiments with Simulated Imagery

The object whose motion is to be studied is a rigid transparent cube of side 4 units. The axes and origin of the object-centered coordinate system are chosen to coincide with the physical axes and centroid, respectively, of the hypothetical cube. The corners of the cube are chosen as the feature points. The simulated measurements are generated by the following steps.

1) The spatial (absolute) 3-D coordinates of the feature points are computed for the desired number of frames using the motion model and predetermined values of the motion parameters, assuming a suitable initial starting point for the moving object.

2) The image locations of the feature points are computed using the central projection model.

3) The "noise" is added to the image coordinates of the feature points by quantizing them to the desired resolution. This is equivalent to adding uniformly distributed noise, instead of Gaussian noise. As discussed in [16], feature point detection errors can be modeled as digitization noise. The Gaussian noise

model, though useful for deriving theoretical results, seems inappropriate for this purpose.

The recursive formulation can easily be modified to handle self-occlusion (the disappearance of some feature points due to the motion of the object), but for simplicity it has been assumed that no feature point is missing in any of the frames in the image sequence.

Experimental results for four different cases are reported below. The measurement noise level indicated in each case is the ratio of the standard deviation of the quantization error to that of the signal (expressed as a percentage). The signal standard deviation is defined to be the root-mean-square distance of the feature points from their centroid, computed during the frame when the object is farthest from the camera. The initial state estimates mentioned in Cases 1–3 are obtained by a batch procedure; details are provided in a later section and in the Appendix.

The errors in the output estimates of the IEKF in each frame are displayed graphically. It must be noted that the position, structure and translational velocity estimates, and therefore the corresponding errors, are normalized by the (time-varying) z coordinate of the center of the object-centered coordinate system. In the cases discussed, 4 feature points were tracked over 100 frames, and the following parameters were used to generate the motion: $\omega_x = \omega_y = \omega_z = 0.2$; $v_x = 0.25$; $v_y = 0.2$; $v_z = 0.15$. The object is assumed to be at location (0,0,10) at start. Focal length and sampling period are both assumed to be unity. It is assumed that the feature points have been matched over all the frames. There is no special reason for selecting 4 points, except that it seems unreasonable to expect many more feature point correspondences. In general, the greater the number of matched feature points in each frame, the better will be the performance of the IEKF in terms of speed of convergence and estimation accuracy.

Case 1: (Fig. 2) A moderately low measurement noise level of 2.5 percent was used in this case. A crude initial state estimate (with errors of 20 percent or more in some states) was used. As the figures indicate, the position and velocity estimates converge quite well, requiring about 30 frames for satisfactory convergence. Most of the structure parameters seem to have a small but constant steady-state error, and the attitude parameters (i.e., the quaternions) exhibit sinusoidal oscillations of small magnitude about the correct values. This is probably due to the fact that different combinations of structure and attitude parameters can result in the same spatial position of the feature points, which means that any large errors in the initial structure estimate can cause corresponding errors in the attitude estimates. This problem can be solved by providing more accurate initial structure estimates, or by imposing additional constraints on the structure. Knowledge about the structure of the object

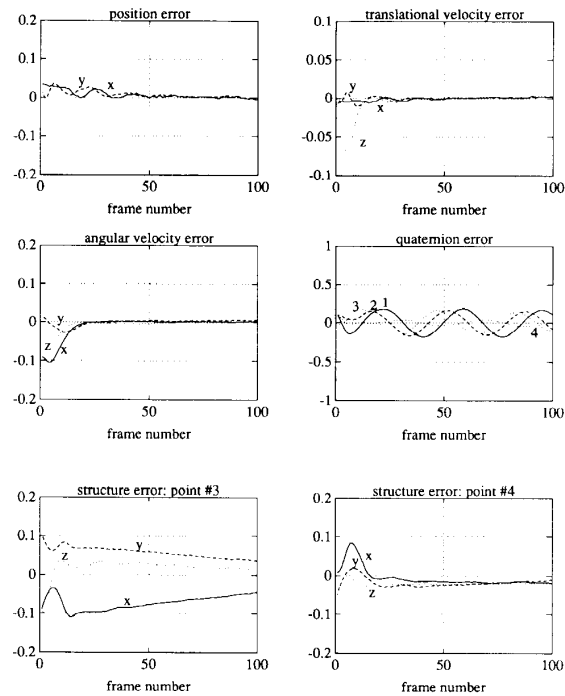


Fig. 2. Estimation errors: Case 1. In this and subsequent figures, labels (x,y,z and 1,2,3,4) are used to indicate vector component corresponding to each plot.

can also be used for this purpose.

Case 2: (Fig. 3) The measurement noise level here was fairly high (10 percent), and the initial guess was crude as in Case 1. The results are similar to those obtained for Case 1, except that the convergence is slower. For instance, the angular velocities converge in about 50 iterations, compared with 25 iterations required for Case 1. This is mainly due to the higher measurement noise.

The above two experiments demonstrate the basic convergence properties of the IEKF, given an inaccurate initial estimate and noisy observations. In actual practice, a much better initial guess can be obtained, resulting in a greatly improved performance, as shown by the next experiment.

Case 3: (Fig. 4) In this case, the measurement noise was the same as in Case 1, but a much more accurate initial state estimate was used. As the graphs show, the filter "locks on" to the motion almost immediately, and tracks it faithfully. The price we pay for this excellent performance of the IEKF is the greater amount of time spent in getting the (more accurate) initial estimate. All the same, this case is the one of greatest practical interest, since it demonstrates the ability of the IEKF to track the motion effectively given a good initial guess (which can be obtained by a batch algorithm, as described in a later section).

Case 4: (Fig. 5) In this case all the initial values of the state vector were set (except the quaternions,

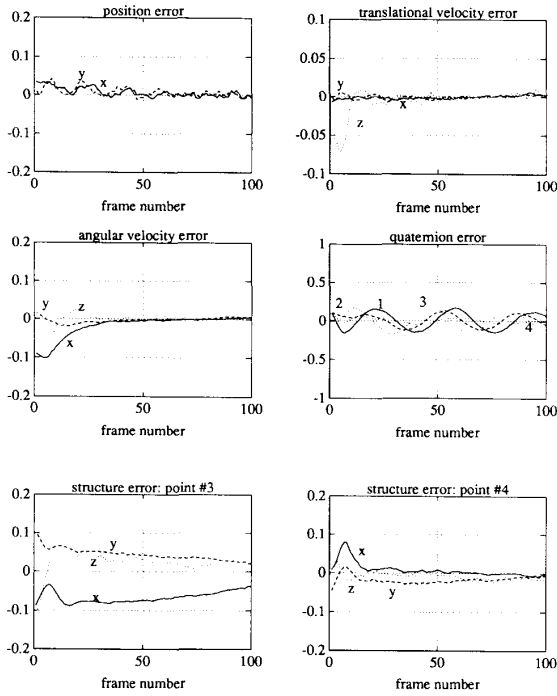


Fig. 3. Estimation errors: Case 2.

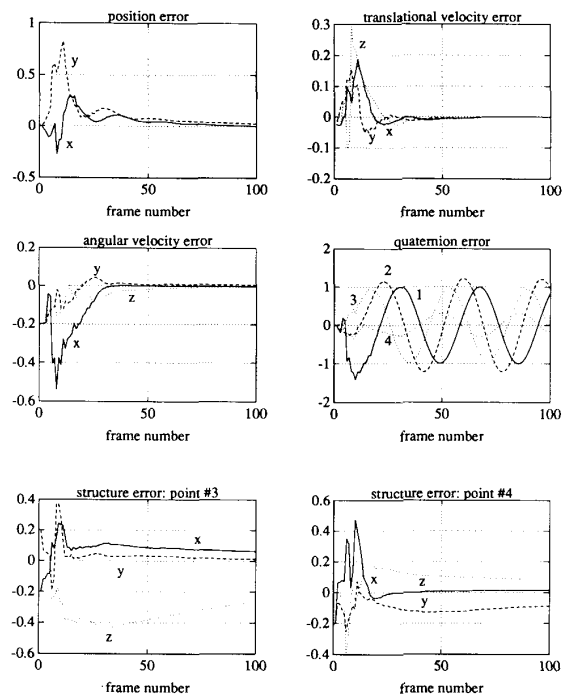


Fig. 5. Estimation errors: Case 4.

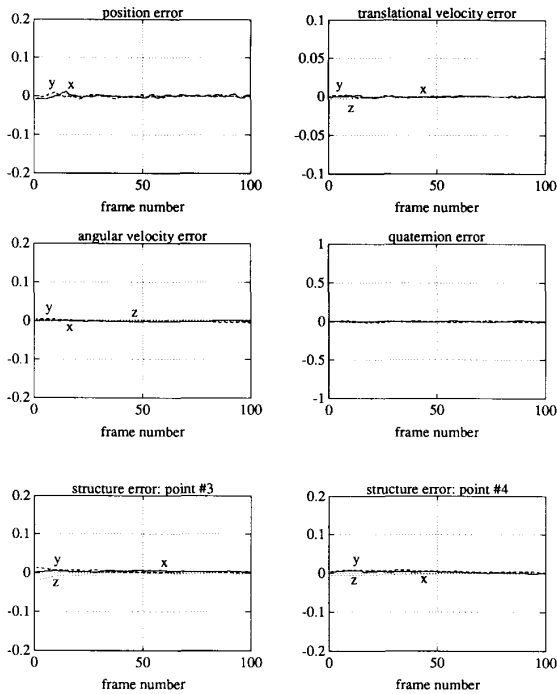


Fig. 4. Estimation errors: Case 3.

which can be trivially initialized if it is assumed that the object-centered and the inertial coordinate systems are aligned at the start of the experiment). Noise-free measurements were used. This case demonstrates

the performance of the IEKF in the absence of any information about the initial conditions. The velocity and position states of the IEKF, after an initial period of wild fluctuations, converge quite fast, in about 25 iterations. The performance is interesting, considering the extreme nonlinearity of the problem, and the consequent mismodeling produced by linearization. The quaternions, however, do not converge at all, and the structure states seem to converge very slowly. Furthermore, a high degree of instability was observed in the solution, due to ill-conditioning of the matrix to be inverted in the computation of the Kalman gain; its generalized inverse had to be used instead. The performance deteriorated rapidly when small amounts of noise (up to 2.5 percent) were added to the measurements.

B. Experiments with Real Imagery

This experiment involves randomly selected points on the side of the tire of a car approaching the camera (Fig. 6). Seventeen images were made, with eight feature points per frame. The feature points were marked with adhesive dots to facilitate the measurement process. The car was moved approximately 3 in between each frame, corresponding to a tire rotation of about 14.8 deg. The direction of the translation was towards the camera (i.e., in the positive z -direction), with a fairly large component to the right (positive x -direction), and a small downward



Fig. 6. First and fourth frames of real image sequence.

component due to the positioning of the camera. The object image size (i.e., the size of the tire) is about 2 in at the start of the sequence, and about 3 in at the end. The total rotation was about 4 rad and the total translation about 45 in. The photographs were digitized to a resolution of 50 pixels/in. Two previously chosen reference points were located on all the images, and the distances of the feature points from them were measured on a Sun workstation. A simple geometrical transformation was used to reference all measurements to the coordinate axes in the first image. This was done to reduce errors due to small camera movements during imaging, and the positioning of the photographs during scanning. Feature point correspondences were obtained manually by inspection. The focal length of the imaging system was not known, and was assumed to be unity. This has the effect of scaling the translation and structure parameters up or down, but is not a serious problem since the latter can only be determined up to a scale factor.

As mentioned before, the actual state values are not known to us, so it is not possible to display the errors in the state estimate. Instead, the actual and the estimated trajectories of the feature points are shown in Fig. 7. The filter should ideally "lock on" to the motion of the target within the first few frames, and should track it efficiently in spite of small errors in measurement and modeling. Fig. 7 seems to confirm that the IEKF is doing a reasonably good job of tracking the moving object. However, in certain cases this might be misleading, since it is possible that two or more entirely different sets of motion parameters could give rise to similar image point trajectories. This is discussed further in the next section. Ideally, "ground truth" measurements using reliable measuring devices should be used to verify the IEKF state estimates.

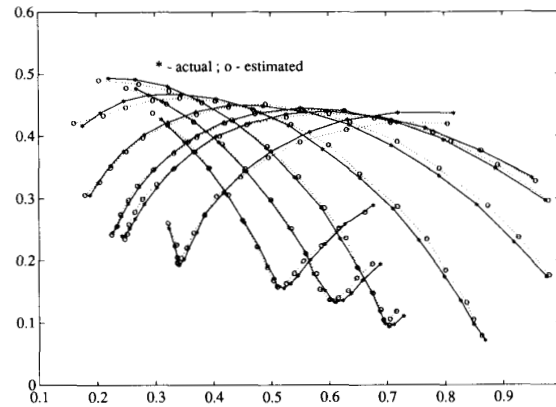


Fig. 7. Actual and estimated image point trajectories.

C. Selecting the IEKF Parameters

In order to run the IEKF, the following parameters have to be supplied in addition to the image point measurements: 1) initial estimate \hat{s} , 2) initial error covariance $P(0)$, 3) plant noise covariance matrices Q_k , and 4) measurement (observation) noise covariances matrices R_k .

The easiest way to obtain a good initial estimate is to run a batch estimation algorithm on the first few frames. Details of the batch algorithm used in this implementation are given in the Appendix. For the particular motion parameters chosen in the simulations, the batch algorithm required about 250 iterations to converge. For Cases 1 and 2, only a crude initial guess was desired, and hence the batch algorithm was forcibly terminated after about 75 iterations. For Case 3, final output of the batch algorithm after convergence was used. In all three cases, the first 10 frames were used for obtaining the initial state estimate. For the real image sequence, the initial guess was obtained by running a batch estimation algorithm on the first 14 frames for about 150 iterations.

The uniqueness of the batch solution depends mainly on the number of feature points, the number of frames, and the motion parameters. This issue is addressed in detail in [19]. As expected, the batch solution is found to be more reliable as the number of feature points and the number of frames increase; but it is also dependent on the amount of motion between frames. It is observed that multiple batch solutions are obtained only when there is some genuine ambiguity in the motion. For instance, if the frame rate is very high compared with the motion velocities, the motion is likely to be ambiguous given a fixed number of frames. This was observed, for instance, in the real image experiment when only the first 8 frames, instead of the first 14, were used to obtain the batch solution. Fig. 8 shows the image plane trajectories reconstructed from the 8-frame batch solution, which should be compared with the actual image plane trajectories in Fig. 7. It can

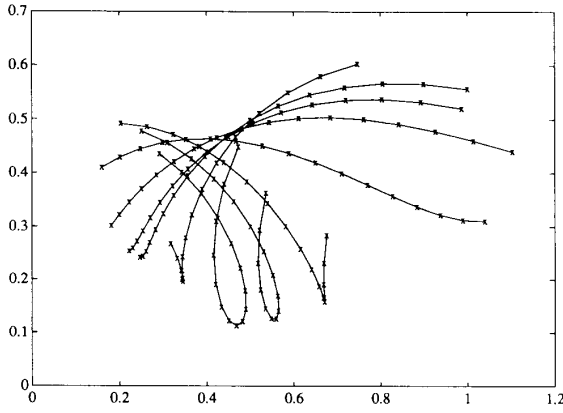


Fig. 8. Trajectories reconstructed from 8-frame batch solution.

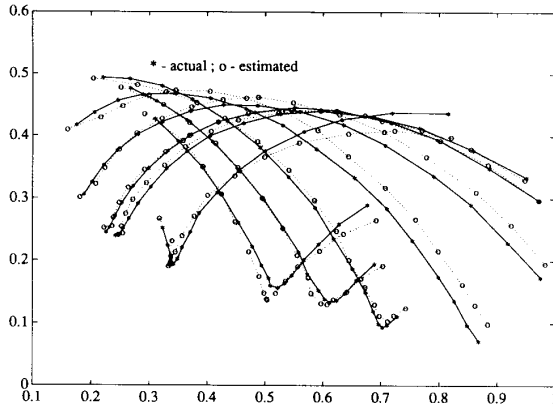


Fig. 9. True and estimated trajectories from bad initial guess.

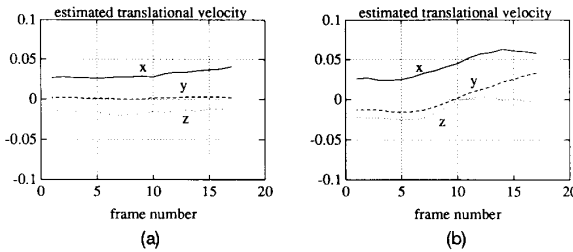


Fig. 10. Sample estimates using two different initial guesses.

be seen that the two sets of trajectories are very similar for the first 8 frames or so, after which they gradually separate. The trajectories estimated by the IEKF, using the 8-frame batch solution as the initial guess, are shown in Fig. 9. From this it appears as if the IEKF is correctly tracking the moving object. However, some state estimates seem to be diverging; this is shown in Figs. 10(a) and 10(b) where the translational velocity estimates for the 14-frame and the 8-frame cases, respectively, are shown. The estimates for the 14-frame case appear to be much more stable than those for the 8-frame case.

If a sufficiently accurate batch solution is available, approximate Cramér-Rao lower bounds (CRLBs) can be found for the error in the initial guess, i.e., the initial error covariance [30]. In the experiments described above, this was possible only for Case 3 and the real image experiment. In the remaining simulations, ad hoc values were used for $P(0)$.

The variance of the measurement noise is known for the simulated data, being a direct function of the grid resolution. (For instance, a grid resolution $\delta = 0.04$ was used for Case 2, which results in noise variance $\sigma_n^2 \approx 1.33 \times 10^{-4}$). Since this does not depend on time, we may set $R_k = R = \sigma_n^2 \times I$, assuming the measurement errors to be independent. For the real image experiment, determination of the actual measurement noise is difficult, since this involves modeling the various sources of error in the imaging system. In our research, it was assumed that the only noise in the measurements is quantization noise resulting from the scanning of the photographs. Using this assumption, the R_k for the real image experiment were chosen as in the cases involving simulated data.

There is no simple method for selecting "good" values for the plant noise matrices Q_k which play an important role in filter performance. If the Q_k chosen are inappropriate for the problem, the filter is likely to diverge, or converge to the wrong value. This is mainly due to the extremely nonlinear nature of the problem, particularly in the observation equations. The filter has therefore to be "tuned" for satisfactory convergent behavior. Convergence implies that the Kalman gain matrices generated by the IEKF should decrease in magnitude at an appropriate rate. The Kalman gain sequence of the IEKF is given by

$$K(k)_{n+1} = \hat{P}(k | k-1) H^x(k)_n^T \times [H^x(k)_n \hat{P}(k | k-1) H^x(k)_n^T + R_k]^{-1}. \quad (62)$$

The time-varying estimate error covariance P in the above equation is a function of the initial error covariance $P(0)$, the measurement noise R and the plant noise Q . Since $P(0)$ and R are fixed, the only way to control the above equation is through the plant noise Q . In our implementation, we have used the term $G_t Q_t G_t^T$ in (55) as the tuning parameter, assuming it to be a scalar multiple q times the identity matrix, for all t . A very low value for q would result in a rapid reduction of the predicted estimate error covariance and the Kalman gain, making the estimates insensitive to the measurements. A very high value would have exactly the opposite effect, causing the estimates to respond to every minute error in the measurements. Clearly, both these extremes are undesirable, since they will have an adverse effect on the filter convergence. These effects are illustrated in Fig. 11(a), where the estimation errors are shown

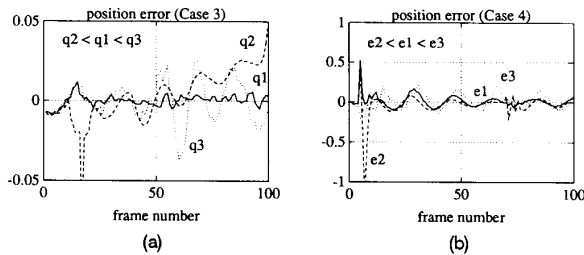


Fig. 11. Effects of varying plant noise (q) and threshold (e).

for one state (x_R) in Case 3 for three different values of q : $q_1 = 10^{-5}$, $q_2 = 10^{-11}$, and $q_3 = 0.1$. In our simulations we have chosen, by trial and error, those values of q that have resulted in good filter performance. Several other strategies are possible, involving adaptive estimation, as discussed in [28]. These issues have to be addressed in future work on recursive motion estimation.

Another effect which can be observed in certain rare cases is the ill-conditioning of the matrix inversion involved in (62). The H matrix in the equation is very sparse, and the pre and postmultiplication of P by H and H^T , respectively, results in a sparse matrix which has mostly off-diagonal elements, and is therefore likely to be ill-conditioned. The matrix inversion in (62) guaranteed to be well conditioned only if the measurement noise covariance R is sufficiently large (i.e., a large multiple of the unit matrix) and is the dominant term in the matrix summation involved in (62). This condition was not satisfied in Case 4, and the generalized inverse was therefore used. The computation of the generalized inverse of a matrix requires the specification of a threshold parameter (e), below which all singular values of the matrix are set to zero. The effects of varying e on the estimation of one state (x_R) in Case 4 are shown in Fig. 11(b). The estimation errors are plotted for three values of e , 0.001(e_1), 0.0001(e_2), and 0.01(e_3). The sudden increase in the estimation error near the 5th frame for $e = 0.0001$ is due to numerical inaccuracies in the computation of the generalized inverse, i.e., ill-conditioning. Higher values of e result in greater numerical stability, but lead to loss of information contained in the smaller singular values. The selection of the threshold could be made automatic by specifying that all singular values falling below a certain fraction of the largest one should be treated as zero. In future work, we will examine other numerically stable approaches such as the square root information filter.

VI. CONCLUSION

The purpose of this paper is to introduce a recursive method of estimating 3-D kinematics and structure of a rigid moving object from a sequence of noisy monocular images. We have described models

for the kinematics of a rigid object, and for the observation of the discrete features of the object using a single camera. The recursive estimation is done using an IEKF which is initialized by the output of a batch algorithm run on the first few frames.

The recursive estimation technique presented in this paper has numerous advantages over other methods currently in use. First, the recursive nature of the computations makes it suitable for real-time applications like object tracking, automated land vehicle guidance, robot hand-eye coordination, etc. The next major advantage is the flexibility in choosing the number of feature points, and the independence of the algorithms on the number of images in the sequence. The other advantages are robustness in the presence of noise and modeling errors, and ease of implementation. The main drawback with this scheme, as it is with most other feature-based methods, is the need for feature point correspondences; however, the predictive capabilities of the Kalman filter can be used to reduce the computation time required to match the feature points from one frame to the other, as explained in Appendix B. Another drawback is the possibility of filter divergence; this may be overcome by a suitable choice of the initialization parameters, and robust numerical techniques.

The IEKF method of recursive motion/structure estimation can be applied, with suitable modifications, when the image sequence is obtained using two cameras instead of one. The 3-D locations of the feature points can be found using stereo matching, and can be used directly instead of their projections on the image plane [27]. The performance of the IEKF improves considerably if stereo image sequences are used, since the nonlinearity of perspective projection can be eliminated from the measurement equations. Convergence is faster and more reliable, and an accurate initial guess is not required. It is also easier to estimate higher derivatives of the motion parameters. In [27], a constant precession model is used for the rotation of the object, and a constant acceleration model for the translation.

Several other extensions and enhancements to our work are possible. These include use of lines or edges as the features to be tracked (instead of points), interleaving the feature correspondence search with the recursive estimation, and using more general models for the kinematics of the object. Our current research is directed towards accomplishing some of these goals.

APPENDIX A. OBTAINING THE INITIAL ESTIMATE

In the experiments discussed in this paper, we have used the batch estimation technique developed by Broida and Chellappa, discussed in [22] to obtain the initial state estimates for the IEKF. This algorithm does not estimate the quaternions, but the latter

can be trivially initialized if the inertial and the object-centered coordinate systems are assumed to be aligned initially.

In the batch formulation, the vector \mathbf{u} of unknown parameters is

$$\mathbf{u} = \begin{pmatrix} x_R/z_R \\ y_R/z_R \\ \dot{x}/z_R \\ \dot{y}/z_R \\ \dot{z}/z_R \\ \omega_x \\ \omega_y \\ \omega_z \\ x_1/z_R \\ y_1/z_R \\ x_2/z_R \\ y_2/z_R \\ z_2/z_R \\ \vdots \\ x_M/z_R \\ y_M/z_R \\ z_M/z_R \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \\ u_{13} \\ \vdots \\ u_{3M+5} \\ u_{3M+6} \\ u_{3M+7} \end{pmatrix}. \quad (63)$$

The individual components of $s_i(\mathbf{u}, t_k)$ of (6) are (setting $t_0 = 0$),

$$s_i(\mathbf{u}, t_k) = \begin{pmatrix} x_i(\mathbf{u}, t_k) \\ y_i(\mathbf{u}, t_k) \\ z_i(\mathbf{u}, t_k) \end{pmatrix} = \begin{pmatrix} x_R/z_R + t_k \dot{x}/z_R + R_x(\mathbf{u}, i, t_k) \\ y_R/z_R + t_k \dot{y}/z_R + R_y(\mathbf{u}, i, t_k) \\ 1 + t_k \dot{z}/z_R + R_z(\mathbf{u}, i, t_k) \end{pmatrix}.$$

The terms $R_x(\mathbf{u}, i, t_k)$, etc., refer to the x -, y -, and z -components of the camera (inertial) coordinates of the i th match point, which are expressed in the rotated (object-centered) coordinate system as $(x_i/z_R, y_i/z_R, z_i/z_R)$. The rotation matrix $R(\mathbf{u}, t_k)$ of (10) is a function of $u_6, u_7, u_8(\omega_x, \omega_y, \omega_z)$, and t_k . For example, denoting the rs component of $R(\mathbf{u}, t)$ as R_{rs} ,

$$\begin{aligned} R_x(\mathbf{u}, i, t_k) &= R_{11}x_i/z_R + R_{12}y_i/z_R + R_{13}z_i/z_R \\ &= (q_1^2 - q_2^2 - q_3^2 + q_4^2)x_i/z_R \\ &\quad + 2(q_1q_2 + q_3q_4)y_i/z_R \\ &\quad + 2(q_1q_3 - q_2q_4)z_i/z_R \end{aligned}$$

where the time argument of $q(t)$ has been suppressed. The 1 in the z -component of $s_i(\mathbf{u}, t_k)$ is written as z_R/z_R , in an analogous manner to the terms $u_1 = x_R/z_R$ and $u_2 = y_R/z_R$, as discussed above. All

components of the object structure and translational kinematics are then homogeneous in $1/z_R$, which accounts for the global scale factor.

Next, the components of (8) are written as

$$\begin{aligned} X_i(t_k) &= h_X[s_i(\mathbf{u}, t_k)] + n_X(t_k) \\ &= \frac{x_i(\mathbf{u}, t_k)}{z_i(\mathbf{u}, t_k)} + n_X(t_k) \\ &= \frac{u_1 + t_k u_3 + R_x(\mathbf{u}, i, t_k)}{1 + t_k u_5 + R_z(\mathbf{u}, i, t_k)} + n_X(t_k) \end{aligned} \quad (64)$$

and

$$\begin{aligned} Y_i(t_k) &= h_Y[s_i(\mathbf{u}, t_k)] + n_Y(t_k) \\ &= \frac{y_i(\mathbf{u}, t_k)}{z_i(\mathbf{u}, t_k)} + n_Y(t_k) \\ &= \frac{u_2 + t_k u_4 + R_y(\mathbf{u}, i, t_k)}{1 + t_k u_5 + R_z(\mathbf{u}, i, t_k)} + n_Y(t_k). \end{aligned} \quad (65)$$

Then, if the noise terms $n_X(t_k)$ and $n_Y(t_k)$ are assumed to be independent, identically distributed (IID), zero mean Gaussian, then, finding the maximum likelihood (ML) estimate $\hat{\mathbf{u}}$ of \mathbf{u} can be shown to be equivalent to minimizing with respect to \mathbf{u} the sum of squared residuals:

$$G(\mathbf{u}) = \sum_{k=1}^N \sum_{i=1}^M \{ (X_i(t_k) - h_X[s_i(\mathbf{u}, t_k)])^2 + (Y_i(t_k) - h_Y[s_i(\mathbf{u}, t_k)])^2 \}. \quad (66)$$

The above minimization may be done using any standard optimization algorithm. In this research, the IMSL routine for minimization by a quasi-Newton method (UMING) was used. More details of the batch algorithm may be found in [19].

APPENDIX B. OBTAINING FEATURE POINT CORRESPONDENCES

The IEKF described in this paper assumes that the feature points have been matched over successive images (frames) in the sequence. Obtaining feature point matchings is a highly nontrivial problem, and considerable research on this issue has been reported in the literature (for instance, see [31]). This issue has not been addressed in this research, but it appears that in the present formulation the process may be considerably simplified and speeded up by using the predictive capabilities of the Kalman filter. State vectors can be predicted at future time instants, and using these, future feature-point positions in the image sequence can be predicted with known uncertainty. This information can be used to restrict the search space for future matchings.

In order to estimate the image coordinates of a match point at time t , the state estimates $\hat{\mathbf{s}}(t)$ must be computed, and then projected directly as $\mathbf{h}[\hat{\mathbf{s}}(t)]$.

Recall that the state vector is of dimension d and the measurement is of dimension $2M$. An approximate error covariance of the resulting image plane location estimates of feature point i is then immediately available as

$$C(i, t) = H_i(t) \hat{P}(t | D) H_i(t)^T \quad (67)$$

where $H_i(t) = \nabla_s \mathbf{h}[s_i(t)]$ is the $2 \times d$ linearized measurement function, each row corresponding to one of the two image plane coordinates of feature point i , evaluated at the state estimates $\hat{s}(t)$. Here, $\hat{P}(t | D)$ is the approximate state covariance matrix at time t , conditioned on all available data (e.g., past and future data, if t represents an interpolation, and past data only if t is an extrapolation). The approximate covariance matrix can be computed by integrating the differential equation (55). In correlating observed feature locations with estimated feature locations, a measure of closeness is desired between the predicted coordinates of feature point i and the measured coordinates of feature point j at time t_{k+1} , under the hypothesis that the measured and predicted feature are one and the same. For example, if a single feature is observed, and there are M predicted features, this amounts to an M -ary hypothesis test. As might be expected, the number of hypotheses grows quickly with the number of predicted and observed features, and the possibilities exist that each observed feature could be new (observed for the first time), and/or that each of the predicted features could be occluded. The so-called *residual covariance* matrix [32] is available by computing the measurement covariance matrix

$$R_j(t_{k+1}) = E[v_j(t_{k+1})v_j(t_{k+1})^T] \quad (68)$$

to give the 2×2 matrix

$$S_{ij}(t_{k+1}) = H_i(k+1) \hat{P}(t_{k+1} | t_k) H_i(k+1)^T + R_j(t_{k+1}) \quad (69)$$

making use of (55) and evaluating $H_i(k+1)$ at $\hat{s}(t_{k+1})$. The subscript i in $H_i(k+1)$ indicates the $2 \times M$ matrix obtained by taking the gradients of the scalar elements $X_{i,k+1} = h_X[s_i(t_{k+1})]$ and $Y_{i,k+1} = h_Y[s_i(t_{k+1})]$ with respect to the state vector s , evaluated at $\hat{s}(t_{k+1})$. As discussed in [32], this information can be used to help in associating (labeling) the data observed (unlabeled feature point coordinates) at time t_{k+1} with feature points being tracked, to minimize (or at least estimate) the probability of error in labeling these observed feature points.

Denote the image plane distance z_{ij} between p_i , the predicted coordinates of feature point i , and o_j , the coordinates of the observed feature point j , as $z_{ij} = o_j - p_i$. A normalized distance (χ_2^2) can be computed as

$$d_{ij}^2 = z_{ij}^T S_{ij}^{-1} z_{ij}. \quad (70)$$

If the filter is linear, and the observations are distributed normally, the distances d_{ij}^2 are seen to

be twice the exponents in the univariate probability density of each observation o_j conditioned on the identification of its mean (class label) as $E[p_i]$. Suppose that only a single feature point o_1 is observed, and the assumptions of normality hold. Then, the ML estimate of the correct class (the predicted feature point corresponding to the observed point) is given by

$$\bar{l} = \arg \min_i d_{i1}^2 \quad (71)$$

and the probability of correctly labeling o_1 can be computed from the knowledge of the state and measurement covariances, and the normalized distances of the remaining elements $\{d_{i1}, i = 1, \dots, M\}$. Next, suppose that M feature points are observed, that they truly represent noisy observations of the existing set of predicted feature points (no extra points, no missing points), and an assignment of measured points to predicted points is desired.

This assignment is computed by observing that the vector of distances $\{d_{ij}; i, j \in T\}$ (where T denotes a particular 1-to-1 assignment of observed feature points j to predicted feature points i) is multivariate Gaussian. The (unknown) mean vector of this density is the projection of the actual feature points (of which $\{p_i\}$ are estimates) into the image plane; the covariance matrix takes into account the variances of the observations, the estimates, and the process noise. The computation of the probability of error in this process when there are two observations of two features is discussed in [33].

A solution is obtained by computing the assignment T that minimizes the sum

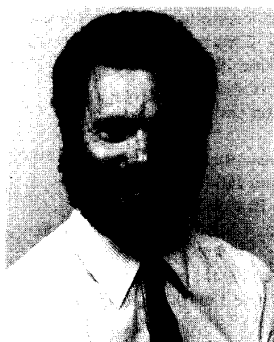
$$\bar{T} = \arg \min_T d^2 = \arg \min_T \sum_{i,j \in T} d_{ij}^2. \quad (72)$$

An optimal algorithm for this purpose, that identifies the minimum sum correlation even when the number of predicted points is not equal to the number of observed points, is discussed in [32], and is known as Munkres' algorithm, or the Hungarian algorithm. This general approach is well known in operations research problems, and allows the extension of [34] to the recursive case.

REFERENCES

- [1] Ballard, D. H., and Kimball, O. A. (1983) Rigid body motion from depth and optical flow. *Computer Vision, Graphics and Image Processing*, 22 (Apr. 1983), 95-115.
- [2] Hildreth, E. C. (1984) Computations underlying the measurement of visual motion. *Artificial Intelligence*, 23 (Aug. 1984), 309-354.
- [3] Adiv, G. (1985) Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7 (July 1985), 384-401.

- [4] Jain, R. C. (1984)
Segmentation of frame sequences obtained by a moving observer.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6 (Sept. 1984), 624-629.
- [5] Burt, P. J., Bergen, J. R., et al. (1989)
Object tracking with a moving camera.
In *Proceedings of the IEEE Workshop on Visual Motion*, Irvine, CA, Mar. 1989, 2-12.
- [6] Roach, J. W., and Aggarwal, J. K. (1980)
Determining the movement of objects from a sequence of images.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-2 (Nov. 1980), 554-562.
- [7] Tsai, R. Y., and Huang, T. S. (1981)
Estimating three-dimensional motion parameters of a rigid planar patch.
IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-29 (Dec. 1981), 1147-1152.
- [8] Tsai, R. Y., Huang, T. S., and Zhu, W. L. (1982)
Estimating three-dimensional motion parameters of a rigid planar patch, II: Singular value decomposition.
IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-30 (Aug. 1982), 525-534.
- [9] Fang, J.-Q., and Huang, T. S. (1984)
Some experiments on estimating the 3-d motion parameters of a rigid body from two consecutive image frames.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6 (Sept. 1984), 545-554.
- [10] Yasumoto, Y., and Medioni, G. (1985)
Experiments in estimation of 3-d motion parameters from a sequence of image frames.
Presented at the *IEEE Conference on Computer Vision and Pattern Recognition*, June 1985.
- [11] Aggarwal, J. K., and Mitiche, A. (1985)
Structure and motion from images: Fact and fiction.
In *Proceedings of the Third Workshop on Computer Vision: Representation and Control*, Oct. 1985, 127-128.
- [12] Meiri, A. Z. (1980)
On monocular perception of 3-d moving objects.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-2 (Nov. 1980), 582-583.
- [13] Gennery, D. B. (1982)
Tracking known 3-d objects.
In *Proceedings of National Conference on Artificial Intelligence*, August 1982, 13-17.
- [14] Shariat, H. (1986)
The motion problem: How to use more than two frames.
Technical report IRIS-202, University of Southern California, Institute for Robotics and Intelligent Systems, Los Angeles, CA, Oct. 1986.
- [15] Dickmanns, E. D., and Graefe, V. (1988)
Dynamic monocular machine vision.
Machine Vision and Applications, 1, 4 (1988).
- [16] Weng, J., Huang, T. S., and Ahuja, N. (1987)
3-d motion estimation, understanding, and prediction from noisy image sequence.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9 (May 1987), 370-389.
- [17] Jazwinski, A. H. (1970)
Stochastic Processes and Filtering Theory.
New York: Academic Press, 1970.
- [18] Broida, T. J., and Chellappa, R. (1986)
Kinematics and structure of a rigid object from a sequence of noisy images.
In *Proceedings of IEEE Workshop on Motion: Representation and Analysis*, May 1986, 95-100.
- [19] Broida, T. J. (1987)
Estimating the kinematics and structure of a moving object from a sequence of images.
Ph.D. dissertation, University of Southern California, Los Angeles, 1987.
- [20] Dickmanns, E. D. (1988)
An integrated approach to feature based dynamic vision.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1988, 820-825.
- [21] Broida, T. J., and Chellappa, R. (1986)
Estimation of object motion parameters from noisy images.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8, 1 (Jan. 1986), 90-99.
- [22] Broida, T. J., and Chellappa, R. (1986)
Kinematics of a rigid object from a sequence of noisy images: a batch approach.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (June 1986), 176-182.
- [23] Broida, T. J., and Chellappa, R.
Estimating the kinematics and structure of a moving rigid object from a sequence of noisy monocular images. (submitted for publication).
- [24] Niva, G. D. (1982)
The use of quaternions with an all-attitude IMU.
In *Proceedings of the Annual Rocky Mountain Guidance and Control Conference*, Jan. 1982, 269-283.
- [25] Friedland, B. (1978)
Analysis of strapdown navigation using quaternions.
IEEE Transactions on Aerospace and Electronic Systems, AES-14 (Sept. 1978), 764-768.
- [26] Wertz, J. R. (Ed.) (1978)
Spacecraft Attitude Determination and Control.
New York: D. Reidel Publishing Co., 1978.
- [27] Young, G. S., and Chellappa, R. (1988)
3-d motion estimation using a sequence of noisy stereo images.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, June 1988, 710-716.
- [28] Maybeck, P. S. (1982)
Stochastic Models, Estimation, and Control, Vol. 2.
New York: Academic Press, 1982.
- [29] Bar-Itzhack, I. Y., and Oshman, Y. (1985)
Attitude determination from vector observations: Quaternion estimation.
IEEE Transactions on Aerospace and Electronic Systems, AES-21 (Jan. 1985), 128-135.
- [30] Broida, T. J., and Chellappa, R. (1989)
Estimation of object motion parameters from noisy images.
Journal of the Optical Society of America A, 6, 6 (June 1989), 879-889.
- [31] Baird, H. (1984)
Model-Based Feature Matching Using Location.
Cambridge, MA: MIT Press, 1984.
- [32] Blackman, S. S. (1986)
Multiple-Target Tracking with Radar Applications.
Dedham, MA: Artech House, 1986.
- [33] Broida, T. J., and Blackman, S. S.
An expression for nearest neighbor data correlation error probability.
IEEE Transactions on Aerospace and Electronic Systems; to be published.
- [34] Sethi, I. K., and Jain, R. (1987)
Finding trajectories of feature points in a monocular image sequence.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9 (Jan. 1987), 56-73.



Ted J. Broida received the B.A. degree in fine art from Antioch College, Yellow Springs, OH, in 1977, and the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles. His doctoral research was performed in the Signal and Image Processing Institute.

He has been employed by Hughes Aircraft Company since 1979, in the Advanced Programs Division of the Radar Systems Group, where he is a Senior Staff Engineer. His work includes design, analysis, and performance prediction of both single sensor and multiple sensor fusion systems for multiple target tracking, as well as real-time sensor and processor resource allocation problems. His research interests also include applications of estimation theory and nonlinear filtering theory. He is the author or coauthor of a number of papers in these areas, co-authored a chapter in *Multiple Target Tracking with Radar Applications* with S. S. Blackman, and recently presented at a UCLA short course on Tracking with Electro-Optical (Imaging) Sensors.

Dr. Broida was a recipient of the Hughes B.S. Scholarship, and the M.S. and Ph.D. fellowships. In 1980 he was awarded the Certificate of Outstanding Academic Achievement at USC by the Metro LA section of the IEEE.



S. Chandrashekhkar (S'88) received the B.Tech. degree with distinction in electronics from the Regional Engineering College, Calicut, India in 1985, and the M.E. degree with distinction in computer science from the Indian Institute of Science, Bangalore, India in 1987. He is currently working towards a Ph.D. degree in electrical engineering at the University of Southern California, Los Angeles.

He is chiefly interested in current problems in image processing and computer vision, especially those related to the analysis of visual motion; use of recursive techniques and neural-like algorithms for computer vision problems; surface reconstruction and stereo.

Mr. Chandrashekhkar is a recipient of the Pre-Doctoral Merit Fellowship of the University of Southern California.

Rama Chellappa was born in Madras, India, and received the B.S. degree (honors) in electronics and communications engineering from the University of Madras, India, in 1975, and the M.S. degree (with distinction) in electrical communication engineering from the Indian Institute of Science in 1977, and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1978 and 1981, respectively.

During 1979–1981, he was a Faculty Research Assistant at the Computer Vision Laboratory, University of Maryland, College Park, MD. Since 1986, he has been an Associate Professor in the Electrical Engineering Systems of the University of Southern California, Los Angeles, and as of September 1988, he has been the Director of the Signal and Image Institute at the University of Southern California, Los Angeles. His current research interest are in Signal and Image Processing, Computer Vision, and Pattern Recognition.

Dr. Chellappa is a member of the Tau Beta Pi and Eta Kappa Nu. He coedited two volumes of selected papers on image analysis and processing, published in Fall 1985. He served as Associate Editor for *IEEE Transactions on Acoustics, Speech and Signal Processing* and is a co-editor of *Computer Vision, Graphics and Image Processing: Graphic Models and Image Processing*. He was a recipient of a National Scholarship from the Government of India during 1969–1975. He received the 1975 Jawaharlal Nehru Memorial Award from the Department of Education, Government of India, the 1985 Presidential Young Investigator Award, and the 1985 IBM Faculty Development Award. He also received the 1990 School of Engineering Award for Excellence in Teaching at the University of Southern California. He was the General Chairman of the 1989 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Workshop on Artificial Intelligence for Computer Vision and also Program Co-Chairman of the NSF sponsored Workshop on Markov Random Fields for Image Processing, Analysis and Computer Vision.

