USC-SIPI REPORT #170

A Two-Step Approach to Passive Navigation Using a Monocular Image Sequence

by

S. Chandrashekhar and R. Chellappa

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
Powell Hall of Engineering
University Park/MC-0272
Los Angeles, CA 90089 U.S.A.

Partially supported by the Office of Naval Research under the grant N00014-89-5-1598.

A Two-Step Approach to Passive Navigation Using a Monocular Image Sequence*

S. Chandrashekhar R. Chellappa

Signal and Image Processing Institute
Department of Electrical Engineering–Systems
University of Southern California
University Park, MC-0272
Los Angeles, CA 90089

Abstract

This paper discusses the design and application of estimation techniques to the problem of obtaining the kinematics of a moving vehicle and the structure of the external environment, based on a sequence of images taken by a camera attached rigidly to the vehicle. "Structure" refers to the 3-D locations of selected points of interest (landmarks) in the environment, expressed in an inertial, or "world" coordinate system. The (noisy) image plane trajectories of these feature points are assumed to be available, and the 3-D positions of a small number (typically 3 or 4) of them are assumed to be known. Using simple models for the motion of the vehicle it is possible to utilize effectively the information contained in image sequences of arbitrary length. Batch and recursive estimation techniques are developed to estimate the motion and structure parameters. In the first step, the batch estimation method is applied to obtain an initial estimate and its approximate error covariance, which is then used in the second step to initialize the recursive filter for tracking the parameters of interest. Experimental results on synthetic data and on one of the UMASS ALV image sequences are given.

*Partially supported by the Office of Naval Research under the grant N00014-89-5-1598

1 Introduction

Several new trends have emerged in the field of visual motion analysis in recent years, such as the use of motion models [1, 2, 3], long sequences[2, 4, 5], and estimation theoretic methods [6, 7, 8, 9, 10, 11]. The use of motion models simplifies motion analysis by reducing the number of parameters to be estimated. Sensitivity to measurement noise and feature point occlusions can be greatly reduced by incorporating the information contained in long image sequences . Finally batch and recursive estimation techniques can be used together for real-time motion analysis and tracking applications. These approaches may be combined to yield a powerful paradigm for visual motion analysis, which is simple, robust, flexible and efficient.

In this paper, we develop this paradigm for the *passive navigation* problem, in which the objective is to aid the visual navigation of a vehicle in an environment containing a small number of landmarks, some known and some unknown. The vehicle is assumed to be equipped with a camera which obtains images of the scene at regular intervals, generating an image sequence. The various parameters of interest in the navigation of the vehicle are to be estimated based on a set of feature points identified and matched over the image sequence. The kinematic parameters of interest are the position of the camera[1], its velocity, and higher-order translational parameters; its attitude (i.e. orientation), angular velocity, and higher-order rotational parameters. The structure parameters involved are the locations of the unknown landmarks in the scene, represented as point features. In this work, all these motion and structure parameters are represented in 3-D in an inertial "world" coordinate system, which is external to the moving vehicle.

Our basic approach has been to develop models for the motion of the camera, the time-evolution of this motion, and the observation of point features in the environment, and to formulate the problem as one of parameter or state estimation. Based on N frames of data, with noisy image coordinates of M features in each frame, we form estimates of the unknown parameters in the assumed models. The estimation techniques which could be used are of two basic types — batch and recursive [12, 13]. In a batch procedure [2], all the information available about the motion of the feature points is used together in a single-stage estimation of the parameters of interest. This approach is robust and simple, and reasonably fast if the number of parameters is small, but suffers from the drawback that the estimates will be available only at the end of the sequence, after measurements from all the images in the sequence become available. Furthermore, it requires large amounts of memory to store these

[1] The camera is assumed to be rigidly fixed to the vehicle. Hence the words "camera" and "vehicle" are used interchangeably in the rest of the paper.

measurements. The recursive approach [14] , on the other hand, is much faster, and requires much less storage, since at any time instant only the current values of the estimates are stored. However, the speed of convergence of the recursive approach depends on the accuracy of the initial guess provided. A good compromise would be to use the batch approach over the first few image frames, and use the resulting batch estimate to initialize the recursive estimator. Then, for the remainder of the sequence, the recursive estimator can be used to "track" the motion and structure parameters.

In the batch formulation, the parameters to be estimated are placed in a parameter vector. The batch estimate is obtained using the principle of nonlinear least-squares, minimizing the discrepancy between the observed image point locations and the ones obtained using the parameter vector and the models of motion and imaging. The batch estimate and its approximate covariance computed using Cramér-Rao lower bounds (CRLBs) are used to initialize the recursive estimator. In the recursive formulation, the parameters to be estimated are treated as states, and a state-space representation is obtained. 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 the measurement model precludes a simple Kalman filtering approach, and hence an approximate nonlinear filtering approach is used. An Extended Kalman Filter (EKF) [15] is designed for the problem, and its performance on real and synthetic data is studied.

The approach developed in this paper has several advantages. Both kinematic and structure parameters are estimated based entirely on the image sequence, without requiring additional sensors such as those required in [11, 16, 17]. Furthermore, parameters such as the (absolute) orientation and position of the camera in the WCS are estimated, unlike most existing methods. Sensitivity to measurement noise is reduced by incorporating the information contained in long image sequences. There is no restriction on the number of feature points or on the number of image frames. Occlusion and disappearance of feature points are handled easily by software, without the need for reprogramming. No initial guess of any parameter is needed, although any prior information can be conveniently and effectively utilized. The method has the ability to cope with fairly large modelling errors, and none of the assumptions made in developing the approach (such as smoothness of motion) seems to be critical to its success in a real application. The only major precondition is the requirement of feature point correspondences. This issue can be addressed using methods such as those presented in [16, 18, 19, 20], but it has not been implemented in our work so far.

2 Models

The fundamental model of this paper is that the motion of the camera 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.

As mentioned earlier, all the parameters of interest, including the kinematics of the camera and the structure of environmental landmarks, are represented in a world coordinate system (WCS), the origin of which is not observed, in general. The WCS is a stationary coordinate system external to the moving vehicle. A camera-centered coordinate system (CCS) is also defined. The translational kinematics of the camera are defined to be the position and motion of the origin of the CCS with respect to the WCS. Camera rotational kinematics are defined to be the camera's angular position and motion with respect to the WCS. The feature points in the scene are assumed to be constant in the WCS, i.e the scene is rigid. The camera coordinates of a feature point can be found by applying a simple transformation on its world coordinates. Using the central projection imaging model, the image of each point is then simply the ratio of its x– and y–coordinates to its z–coordinate (all in the CCS), multiplied by the appropriate focal lengths, and shifted by the coordinates of the centre of the image. These concepts are made precise in this section.

2.1 Translational Motion Model

Since the spatial coordinates of the origin of the CCS $\mathbf{p}_R(t)$ 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,

$$\mathbf{p}_R(t) = \mathbf{p}_R(t_0) + \sum_{k=1}^{n} \left. \frac{\partial^{(k)} \mathbf{p}_R(t)}{\partial t^{(k)}} \right|_{t=t_0} \frac{(t-t_0)^k}{k!} \tag{1}$$

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.

2.2 Rotational Motion Model

Quaternions, described for example in [21], can be used to propagate the transformation matrix $R(t)$ in time, with the rotation of the CCS represented by the camera's 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 $\mathbf{q(t)} = (q_1(t), q_2(t), q_3(t), q_4(t))^T$. Suppressing the time dependency,

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

or, more briefly,

$$R(t) = R[\,\mathbf{q}(\mathbf{t})\,]. \qquad (3)$$

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

$$(q_1, q_2, q_3, q_4)^T = (n_1 \sin\theta/2,\ n_2 \sin\theta/2,\ n_3 \sin\theta/2,\ \cos\theta/2)^T \qquad (4)$$

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 (in the least angle sense). It is termed a unit quaternion because $|\mathbf{q}| = 1$. The quaternion $\mathbf{q}$ propagates in time according to the differential equation [22, 21]

$$\dot{\mathbf{q}}(t) = \Omega(\omega_t)\,\mathbf{q}(t), \quad \mathbf{q}(t_0) = \mathbf{q}_0 \qquad (5)$$

where

$$\Omega(\omega_t) = \frac{1}{2} \begin{pmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{pmatrix}_t \qquad (6)$$

The reason for resorting to quaternions is that the differential equation of (5) 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 non-orthogonal axes. In order to represent higher order rotational derivatives, ω can be expanded in Taylor series as in (1), and the differential equation of (5) can be integrated numerically, with time-varying ω.

2.3 Imaging Model

A central projection imaging model is assumed, defined by

$$h : P \mapsto \Pi \qquad (7)$$

where

$$\mathbf{p} = (x, y, z)^T \in P = \{(x, y, z)^T \in R^3\} \tag{8}$$

is a spatial point location and

$$\rho = (X, Y)^T \in \Pi \subset R^2 \tag{9}$$

is an image plane point position (all expressed in the CCS). The space Π is nominally a finite rectangle, corresponding to the image plane of a camera. Then,

$$X = f_x \cdot \frac{x}{z} + X_0 + n_X, \qquad Y = f_y \cdot \frac{y}{z} + Y_0 + n_Y \tag{10}$$

map spatial coordinates to noisy image coordinates, where f_x and f_y are the focal lengths of the camera in the vertical and horizontal directions, and (X_0, Y_0) are the coordinates of the centre of the image, expressed in the CCS. The terms n_X and n_Y are the image plane noise components, assumed to be zero mean, independent and identically distributed (i.i.d). Thus, the measurement model for a single point $\mathbf{p} = (x, y, z)^T$ is

$$\rho = \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} f_x \cdot \frac{x}{z} + X_0 \\ f_y \cdot \frac{y}{z} + Y_0 \end{pmatrix} + \begin{pmatrix} n_X \\ n_Y \end{pmatrix} = h[\mathbf{p}] + \mathbf{n}. \tag{11}$$

2.4 Observation Model

Combining the above results, we obtain the following model expressing the relationship between the parameters to be estimated and the observed image locations of the feature points. Let $\mathbf{p}_i = (x_i,\, y_i,\, z_i)^T$ be the world coordinates of match point i. Let $\mathbf{p}_R(t) = (x_R(t),\, y_R(t),\, z_R(t))^T$ be the world coordinates of the origin of the moving camera-centred reference frame (not observed directly). Let $R(t)$ be the 3 × 3 coordinate transformation matrix that aligns the world coordinate axes with the camera coordinate axes, changing with time as the camera rotates. Let $\mathbf{p}_{iC}(t)$ be the spatial, camera-centered coordinates of match point i at time t.

The complete observation model is then given by the following equations.

$$\mathbf{p}_{iC}(t) = R'(t)\,(\mathbf{p}_i - \mathbf{p}_R(t)) \tag{12}$$

At time t_k the image plane measurements of the match points are, from (11),

$$\rho_i(t_k) = h[\mathbf{p}_{iC}(t_k)] + \mathbf{n}(t_k), \tag{13}$$

which can be written as

$$\rho_i(t_k) = (X_i, Y_i)_k^T = h\,[R'(t)\,(\mathbf{p}_i - \mathbf{p}_R(t))] + \mathbf{n}(t_k), \tag{14}$$

where $i = 1, 2, \ldots, M$ and $k = 1, 2, \ldots, N$ for M match points and N image frames in the sequence.

3 Batch Formulation

The general point feature based motion analysis problem, in the context of passive navigation, may be stated as follows : Given M image point correspondences over N image frames, determine the parameters related to the motion of the camera and the structure of the feature points. There are two different approaches to this problem :

1. All motion and structure parameters are represented in the coordinate system of the camera. The objective is to estimate the camera's position and orientation, and the structure of the scene, *relative* to the camera's initial position, as well as the velocities and higher order kinematics of the camera in the CCS. For this approach it is not necessary, in general, to assume any knowledge about the structure of the environment.

2. All parameters of interest are represented in an external, stationary world coordinate system. The objective here is to estimate the *absolute* values of these parameters, i.e. their values in the WCS. In particular, the absolute position and orientation of the camera need to be determined. For this purpose it is necessary to have constraints relating the CCS to the WCS. One reasonable approach is to assume that some of the feature points in the images correspond to navigational landmarks whose 3-D locations in the WCS are known. This provides a means of relating the two coordinate systems.

In this paper, the second approach is followed. It is assumed that there are a total of M feature points in the environment, M_k of which are known, and the remaining $M_u = M - M_u$ unknown. A simple model for the camera's motion, with constant camera orientation and constant translational velocity, is assumed. This does not mean that the resulting techniques are inapplicable to more general situations; on the contrary the methods give excellent results even in the presence of significant model deviations, as the experimental results later illustrate. The following d –dimensional vector of parameters is then selected:

$$\boldsymbol{\theta} = \begin{pmatrix} \mathbf{p}_R(0) \\ \mathbf{v} \\ \mathbf{a} \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_{M_u} \end{pmatrix} \tag{15}$$

The orientation of the camera is represented by the 3-component vector a in the above expression, instead of the 4-component unit quaternion q. This is done to avoid the constraint which would be needed on the estimation process if the unit quaternion were estimated directly. The relationship between a and q is given by (4) and the following equation :

$$\mathbf{a} = \begin{pmatrix} n_1\ \theta \\ n_2\ \theta \\ n_3\ \theta \end{pmatrix} \tag{16}$$

The data on which the estimation is based are the the image point measurements of M points in N frames, denoted by

$$\rho_i(k) = h_i(\boldsymbol{\theta}, k) + \mathbf{n}_i(k)\ ;\ i = 1, \ldots, M\ ;\ k = 1, \ldots, N, \tag{17}$$

where $h_i(\boldsymbol{\theta}, k)$ is defined to be the location of of the ith feature point in the kth image in the sequence, computed using the motion and structure parameters in the vector $\boldsymbol{\theta}$, and the model in (14). The noise terms in $\mathbf{n}_i(k)$ are assumed to be zero mean, independent, and identically distributed (i.i.d.), for all points over the sequence.

The batch estimation problem may now be stated as follows : find the best estimate of $\boldsymbol{\theta}$ given the measurements $\rho_i(k)$ and the observation model (14). For our particular problem, wherein no prior information is assumed about $\boldsymbol{\theta}$, and with minimal assumptions about the measurement noise, the "best" estimate may be considered to be the one which minimizes the squared discrepancy between the measurements and the corresponding values predicted by the model, i.e.

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=0}^{k-1} \sum_{i=0}^{M-1} \|\rho_i(k) - h_i(\boldsymbol{\theta}, k)\|^2 \tag{18}$$

This least-squares minimization can be done using a standard optimization program, such as the ones used in [2, 23]. Any knowledge about the ranges or values of the parameters can be used to reduce the search space for a solution.

3.1 Computing the approximate covariance of the batch estimate

For most applications, it is useful, if not essential, to have some idea of the "correctness" of the estimated parameter set. An elegant estimation-theoretic method exists for the computation of the lower bounds of the error in estimating a set of parameters, given the conditional statistics of the observed data on which the estimates are based. Using this technique, the so-called Cramér-Rao lower bounds (CRLBs), on the covariance of the estimation error can be computed, and used as an approximation to the true error covariance[2]. Similar methods are used in [24, 25].

Forr the purpose of deriving the CRLBs, we represent the conditional p.d.f. of the measurements as multivariate Gaussian [3] , of the form:

$$p(\mathrm{z}\,/\theta) = \prod_{k=0}^{N-1} \prod_{i=0}^{M-1} \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{\|\rho_i(k)-h_i(\theta,k)\|^2}{2\sigma^2}} \tag{19}$$

where z is a vector containing all the measurements, and σ^2 is the variance of the measurement noise in each coordinate, to be obtained during calibration.

Let the estimation error covariance be

$$C_\theta \triangleq \mathcal{E}\left\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)^T\right\}$$

Define the $d \times 1$ column vector

$$\mathrm{d} \triangleq \frac{\partial \ln\, p(\mathrm{z}\,/\theta)}{\partial \theta}$$

and the $d \times d$ matrix

$$J \triangleq \mathcal{E}\left\{\mathrm{d}\mathrm{d}^T\,/\theta\right\}$$

(The matrix J is called the Fisher information matrix.) Then the basic theorem used to compute CRLBs can be stated as :

$$C_\theta \geq J^{-1} \tag{20}$$

Using the expression in (19) for the conditional p.d.f. of the data,

$$\ln\, p(\mathrm{z}\,/\theta) = K - \frac{1}{2\sigma^2} \sum_{k=0}^{N-1} \sum_{i=0}^{M-1} \|\rho_i(k) - h_i(\theta,k)\|^2$$

[2] Strictly speaking, only the approximate CRLBs will be determined by the method discussed in this section. This is because we need to know the bias on the estimation error to compute the exact CRLBs. Since it is usually not possible to obtain this information, the derivation assumes unbiased estimates.

[3] The Gaussian assumption is not required for obtaining the batch estimate.

$$\mathbf{d} = \frac{1}{\sigma^2} \sum_{k=0}^{N-1} \sum_{i=0}^{M-1} \frac{\partial(\rho_i(k) - h_i(\theta, k))^T}{\partial \theta} (\rho_i(k) - h_i(\theta, k))$$

Using the assumption that the measurement errors are independent, the Fisher information matrix can be obtained in the following form.

$$J = \frac{1}{\sigma^2} \sum_{k=0}^{N-1} \sum_{i=0}^{M-1} d_i(k)^T d_i(k)$$

where each of the $d_i(k)$ terms is of dimension $2 \times d$, given by

$$d_i(k) \triangleq \frac{\partial h_i(\theta, k)^T}{\partial \theta}.$$

The term on the right-hand side of the above expression can be simplified further, using the basic model equations. The derivation is similar to that for the H_i terms in the next section, so it will not be repeated here. The computation of the J matrix requires the true values of the parameters, which obviously will not be known in a real problem. Hence the batch estimate $\hat{\theta}$ is used as an approximation to the true parameter vector θ in the derivation of CRLBs.

4 Recursive Formulation

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 [15, 26]

$$\dot{\mathbf{s}}_t = f(\mathbf{s}_t) + G_t \mathbf{w}_t \tag{21}$$

as the plant model, and

$$\mathbf{z}(t_k) = \mathbf{h}[s(t_k)] + \mathbf{v}(t_k) \tag{22}$$

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

The following set of states is chosen for the recursive formulation. Let $\mathbf{s}(t)$ consist of the scalar elements

$$\mathbf{s}(t) = \begin{pmatrix} \mathbf{p}_R(t) \\ \mathbf{v} \\ \mathbf{q}(t) \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_{M_u} \end{pmatrix} \tag{23}$$

The time derivative of $\mathbf{s}(t)$ is

$$\dot{\mathbf{s}}(t) = \begin{pmatrix} \mathbf{v} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} \tag{24}$$

Using (23) and (24) the discrete version of the plant equation can be written as follows :

$$\mathbf{s}(k+1) = F\ \mathbf{s}(k) + G_k \mathbf{w}_k \tag{25}$$

where F is the $d \times d$ state transition matrix,given by

$$F = \begin{pmatrix} I_3 & I_3 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & I_3 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & I_3 \end{pmatrix} \tag{26}$$

and $G_k \mathbf{w}_k$ is a discretized plant noise term which is treated as a set of "tuning parameters", to be manipulated to account for errors in modelling and to keep the filter gains high enough to track the camera motion.

Using (11), the vector-valued measurement function ($2M \times 1$) of (22) is

$$\mathbf{z}(t_k) = \begin{pmatrix} \rho_1(k) \\ \rho_2(k) \\ \vdots \\ \rho_M(k) \end{pmatrix} = \begin{pmatrix} h_1(k) \\ h_2(k) \\ \vdots \\ h_M(k) \end{pmatrix} + \begin{pmatrix} n_1(k) \\ n_2(k) \\ \vdots \\ n_M(k) \end{pmatrix} = \mathbf{h}[\mathbf{s}(k)] + \mathbf{n}_k \tag{27}$$

The covariance R_k of the measurement noise $\mathbf{n}_k$ can be assumed to be σ^2 times the identity matrix, where σ^2 is the variance of the measurement noise in each coordinate. This formulation is appropriate for solution using a recursive filter such as an EKF, and in the rest of this section we design an EKF for the problem under consideration.

Following [15], the estimate $\hat{s}(k|k-1)$ denotes the predicted (extrapolated) estimate, just after a time update, while the estimate $\hat{s}(k|k)$ denotes the smoothed (filtered) estimate, just after a measurement update. The corresponding error covariances are denoted by $P(k/k-1)$ and $P(k/k)$, respectively. The iteration is initialized with

$$\hat{s}(0|0) = E\{s(0)\} = \mu_{s0} = \hat{s}(0). \tag{28}$$

and

$$P(0|0) = P_0 = E\{(s(0) - \mu_{s0})(s(0) - \mu_{s0})^T\} \tag{29}$$

The initial values of the state vector and its covariance can be obtained as described in the previous section.

The time update equation is simply

$$\hat{s}(k/k-1) = F\ \hat{s}(k-1/k-1) \tag{30}$$

The measurement update equation is

$$\hat{s}(k|k) = \hat{s}(k|k-1) + K(k)\Big[z(k) - h[\hat{s}(k|k-1)]\Big], \tag{31}$$

where the Kalman gain sequence $K(k)$ is obtained as follows:

$$K(k) = P(k|k-1)H(k)^T\Big[H(k)P(k|k-1)H(k)^T + R_k\Big]^{-1}. \tag{32}$$

and $H(k)$ is the $2M \times d$ linearized measurement function, given by

$$H(k) = \left.\frac{\partial h[s]}{\partial s}\right|_{s = \hat{s}(k|k-1)}. \tag{33}$$

In our case, with the state vector s as in (23), h[s] as in (27) we obtain

$$H(s) = \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_{M_u} \\ H_{M_u+1} \\ \vdots \\ H_M \end{pmatrix}$$

Each $2 \times d$ submatrix H_i (corresponding to the ith image point ρ_i) is defined as follows:

$$H_i = \begin{cases} \begin{pmatrix} H_{pr} & 0 & H_q & 0 & 0 & \cdots & 0 & H_p & 0 & \cdots & 0 \end{pmatrix} & \text{if } i \leq M_u \\ \begin{pmatrix} H_{pr} & 0 & H_q & 0 & \cdots & 0 \end{pmatrix} & \text{otherwise} \end{cases}$$

where H_{pr}, H_q and H_p, are the partial derivatives of ρ_i with respect to $\mathbf{p}_R$, $\mathbf{q}$ and $\mathbf{p}_i$ respectively, and $\mathbf{0}$ denotes a 2×3 zero matrix. The location of H_p corresponds to the location of the ith feature point in the state vector, if $i < M_u$. In order to obtain expressions for these terms, let us define a new notation for the rotation matrix R and the 3-D location of the feature point in the CCS. Let

$$R = \left(\begin{array}{ccc} R_x' & R_y' & R_z' \end{array} \right)$$

be the rotation matrix (as defined in Section 2.2), written in terms of the rows of its transpose, and

$$\begin{aligned}
R_{xq} &\triangleq \frac{\partial R_x}{\partial \mathbf{q}} = 2 \begin{pmatrix} q_1 & q_2 & q_3 \\ -q_2 & q_1 & -q_4 \\ -q_3 & q_4 & q_1 \\ q_4 & q_3 & -q_2 \end{pmatrix} \\
R_{yq} &\triangleq \frac{\partial R_y}{\partial \mathbf{q}} = 2 \begin{pmatrix} q_2 & -q_1 & q_4 \\ q_1 & q_2 & q_3 \\ -q_4 & -q_3 & q_2 \\ -q_3 & q_4 & q_1 \end{pmatrix} \\
R_{zq} &\triangleq \frac{\partial R_x}{\partial \mathbf{q}} = 2 \begin{pmatrix} q_3 & -q_4 & -q_1 \\ q_4 & q_3 & -q_2 \\ q_1 & q_2 & q_3 \\ q_2 & -q_1 & q_4 \end{pmatrix}
\end{aligned}$$

Let

$$\mathbf{p} = \mathbf{p}_i - \mathbf{p}_R(t)$$

and

$$x \triangleq R_x\mathbf{p} \;;\; y \triangleq R_y\mathbf{p} \;;\; z \triangleq R_z\mathbf{p}$$

Using the above notation,

$$\rho_i = \begin{pmatrix} f_x \cdot \frac{x}{z} + X_0 \\ f_y \cdot \frac{y}{z} + Y_0 \end{pmatrix}$$

The derivative terms may now be obtained directly as:

$$\begin{aligned}
H_{pr} &= \frac{1}{z^2} \begin{pmatrix} f_x[xR_z - zR_x] \\ f_y[yR_z - zR_y] \end{pmatrix} \qquad (34) \\
H_q &= \frac{1}{z^2} \begin{pmatrix} f_x[zR_{xq} - xR_{zq}] \\ f_y[zR_{yq} - yR_{zq}] \end{pmatrix} \\
H_p &= -H_{pr}
\end{aligned}$$

In our implementation, we deviate slightly from the traditional EKF 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 [27] , wherein the authors conclude that "the estimation errors are not affected by the normalization operation".

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

$$P(k|k-1) = F\, P(k-1|k-1)\, F^T + G_k Q_k G_k^T. \tag{35}$$

The smoothed covariance matrix is

$$P(k|k) = \Big[I - K(k)H(k)\Big] P(k|k-1). \tag{36}$$

5 Experimental Results

The estimation algorithms developed in the earlier sections were tested on both synthetic as well as real data. Two examples of experimental results with synthetic data, and one real image example are presented in this section. The real image sequence used is the UMASS "rocket" ALV sequence (Figure 1). The environment for this sequence contains 14 feature points[4], 13 of which are shown in Figure 2, along with the different positions of the camera during its motion. The same environment is assumed for the experiments with synthetic data, but different kinematic parameters are used in simulating the camera motion. In all the experiments, the structure of four of the 14 points is assumed to be known beforehand, and that of the remaining 10 points is estimated, along with the kinematic parameters of the camera. The batch algorithm is run on the first few frames in the sequence (7 for synthetic data and 13 for real data) and the batch estimate, together with the CRLBs as its approximate error covariance, are used to initialize the IEKF for recursive estimation.

Two minor modifications to the algorithms developed earlier are made. The first takes into account the fact that as the camera moves, some of the points in the scene disappear from the image. This information is stored in a binary array $L(i,k)$. A 1(0) in the (i,k)th position of this array indicates the presence(absence) of the ith feature point in the kth image frame in the sequence. The information in L is used in both the batch and the recursive algorithms. Another small modification to the batch method is found to be useful : instead of using all the 10 unknown points at once, only 3 are used initially. This results in a smaller parameter vector, and hence quicker and more reliable estimates of the camera position and

[4] The original sequence contains 18 features, out of which 14 were selected.

motion and the structure of the 3 unknown points selected. The structure of the remaining 7 points is obtained by a simple batch procedure which treats the previously estimated kinematic parameters as fixed. Details of these modifications are omitted for brevity.

5.1 Experiments with Synthetic Data

The simulated measurements are generated by the following steps :

1. The 3-D coordinates of the feature points in the CCS are computed for the desired number of frames using the motion model and pre-determined values of the motion and structure parameters, assuming a suitable initial starting point for the moving camera.
2. The image locations of the feature points are computed using the imaging model.
3. The image coordinates of the feature points are quantized to the desired resolution. In the experiments reported here, an image size of 512 × 512 pixels was assumed.

The results of the simulations are shown in Figures (3) to (6). In the first example, the model assumptions of constant orientation and velocity were followed exactly. The image plane trajectories of the feature points are straight lines (except for the errors due to spatial quantization), shown in Figure 3(a). The structure estimates obtained from the batch algorithm are shown in Figure 3(b). The results of the EKF are shown in Figure 4. In the second example, the camera was permitted to rotate with a small, linearly increasing angular velocity, violating the constant orientation assumption. Nonetheless, the results of the estimation algorithms are still satisfactory, as illustrated by Figures (5) and (6).

5.2 Experiments with Real Data

The first and last images of the real image sequence (consisting of 30 images) are shown in Figure 1. Details of the experimental set-up used to generate this sequence can be found in [28]. Complete ground-truth information about the motion of the camera and the structure of the selected scene landmarks is available, as well as calibration parameters of the camera. The motion of the vehicle is approximately along a straight line, and its orientation is approximately constant. However, as indicated by the image plane trajectories of feature points illustrated in Figure 7(a), a fair amount of apparently random variation is present in the camera's orientation. The challenge here is to track these variations, while preserving the simplicity of the models developed earlier. There is also some discrepancy (of upto 8 pixels) between the actual image locations of feature points, and their locations predicted by the camera calibration and the ground truth. This has to be treated as additional measurement noise. The structure estimates obtained by the batch procedure are shown in Figure 7(b) and the results of the EKF in Figure 8.

It is apparent that the motion of the camera does not obey the model developed in Section 2, since neither its translational velocity nor its attitude is constant. However, the EKF seems to be capable of handling these model deviations, as the state estimates indicate. This is essentially due to the ability of the EKF to "forget" the past, and respond to the present; an ability which can be controlled by varying the assumed plant noise covariance. Thus the EKF can be "tuned" to specific applications, depending on the extent of the model deviations expected.

6 Conclusions

Feature-based motion analysis holds promise for such applications as passive navigation and obstacle avoidance. The need for simplicity and robustness suggests a model-based, estimation-theoretic approach. This paper dealt with the development of such an approach for the passive navigation problem, using simple motion models in conjunction with batch and recursive estimation techniques. The results indicate that the methods developed here have the necessary robustness and flexibility to perform satisfactorily in a real application, wherein considerable model deviations and measurement noise can be expected. Similar methods have been applied successfully to the dual problem of estimating the kinematics and structure of a moving object observed by a stationary camera [3, 14].

Acknowledgements

The real data used in our experiments were provided to us by researchers at the University of Massachusetts. We wish to acknowledge in particular the assistance of R. Dutta and R. Manmatha in helping us to understand the details of this data set.

References

[1] J. Weng, T. S. Huang, and N. Ahuja, 3-D Motion Estimation, Understanding, and Prediction from Noisy Image Sequence, *IEEE Trans. on Patt. Anal. Mach. Intell.*, PAMI-9, pp. 370-389, May 1987.

[2] T. J. Broida and R. Chellappa, Estimating the Kinematics and Structure of a Moving Rigid Object from a Sequence of Noisy Monocular Images, *IEEE Trans. on Patt. Anal. Mach. Intell.*, 1991, Accepted for publication.

[3] G. S. Young and R. Chellappa, 3-D motion estimation from a sequence of noisy stereo images : models, estimation and uniqueness results, *IEEE Trans. on Patt. Anal. Mach. Intell.*, PAMI-12, pp. 735–759, July 1990.

[4] Y. Yasumoto and G. Medioni, Robust Estimation of Three-Dimensional Motion Parameters from a Sequence of Image Frames Using Regularization, *IEEE Trans. on Patt. Anal. Mach. Intell.*, PAMI-8, pp. 464–471, July 1986.

[5] H. Shariat and K. E. Price, Motion estimation with more than two frames, *IEEE Trans. on Patt. Anal. Mach. Intell.*, PAMI-12(5), pp. 417–434, April 1990.

[6] D. B. Gennery, Tracking Known 3-D Objects, In *Proc. National Conf. on Artificial Intell.*, pp. 13–17, August 1982.

[7] E. D. Dickmanns and V. Graefe, Dynamic Monocular Machine Vision, *Machine Vision and Applications*, 1(4), pp. 233–240, 1988.

[8] J. L. Crowley, P. Stelmaszyk, and C. Discours, Measuring Image Flow by Tracking Edge-Lines, In *Second International Conf. on Computer Vision*, pp. 658–664, December 1988.

[9] N. Ayache and O. D. Faugeras, Maintaining Representations of the Environment of a Mobile Robot, *IEEE Transactions on Robotics and Automation*, RA-5(6), pp. 804–819, December 1989.

[10] D. J. Kriegman, E. Triendl, and T. O. Binford, Stereo Vision and Navigation in Buildings for Mobile Robots, *IEEE Transactions on Robotics and Automation*, RA-5(6), pp. 792–803, December 1989.

[11] W. M. Wells, Visual estimation of 3-D line segments from motion - a mobile robot vision system, *IEEE Transactions on Robotics and Automation*, RA-5(6), pp. 820–825, December 1989.

[12] T. J. Broida, *Estimating the Kinematics and Structure of a Moving Object from a Sequence of Images*, Ph.D. Thesis, University of Southern California, 1987.

[13] S. S. Blackman, *Multiple-Target Tracking with Radar Applications*, Artech House, 1986.

[14] T. J. Broida, S. Chandrashekhar, and R. Chellappa, Recursive Estimation of 3-D Kinematics and Structure from a Noisy Monocular Image Sequence, *IEEE Transactions on Aerospace and Electronic Systems*, AES-26(4), pp. 639–656, July 1990.

[15] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, 1970.

[16] B. Roberts and B. Bhanu, Inertial navigation sensor integrated motion analysis for autonomous vehicle navigation, In *Proceedings of the Image Understanding Workshop*, pp. 364–375, DARPA, September 1990.

[17] M. J. Stephens et al., Outdoor vehicle navigation using passive 3D vision, In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 556–562, San Diego, CA, June 1989.

[18] K. E. Price and R. Reddy, Matching segments of images, *IEEE Trans. on Patt. Anal. Mach. Intell.*, PAMI-1, pp. 110–116, January 1979.

[19] S. T. Barnard and W. B. Thompson, Disparity analysis of images, *IEEE Trans. on Patt. Anal. Mach. Intell.*, PAMI-2(4), pp. 333–340, July 1980.

[20] M.K. Leung, Y.C. Liu, and T.S. Huang, Experimental results of 3D motion estimation using images of outdoor scenes, In *Proceedings of the Image Understanding Workshop*, pp. 428–432, DARPA, September 1990.

[21] J. R. Wertz, editor, *Spacecraft Attitude Determination and Control*, D. Reidel Publishing Co., 1978.

[22] B. Friedland, Analysis of Strapdown Navigation Using Quaternions, *IEEE Trans. on Aerospace and Elect. Systems*, AES-14, pp. 764–768, September 1978.

[23] R. V. Raja Kumar et al., A non-linear optimization algorithm for the estimation of structure and motion parameters, In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 136–143, San Diego, CA, June 1989.

[24] T. J. Broida and R. Chellappa, Estimation of Object Motion Parameters from Noisy Images, *Journal of the Optical Society of America A*, 6(6), pp. 879–889, June 1989.

[25] J. Weng, T. S. Huang, and N. Ahuja, Motion estimation from images: image matching, parameter estimation and intrinsic stability, In *Proc. IEEE Workshop on Visual Motion*, pp. 359–366, Irvine, California, March 1989.

[26] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, volume 2, Academic Press, 1982.

[27] I.Y. Bar-Itzhack and Y. Oshman, Attitude Determination from Vector Observations: Quaternion Estimation, *IEEE Transactions on Aerospace and Electronic Systems*, AES-21, pp. 128–135, January 1985.

[28] R. Dutta et al., A Data Set for Quantitative Motion Analysis, In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 159–164, San Diego, CA, June 1989.

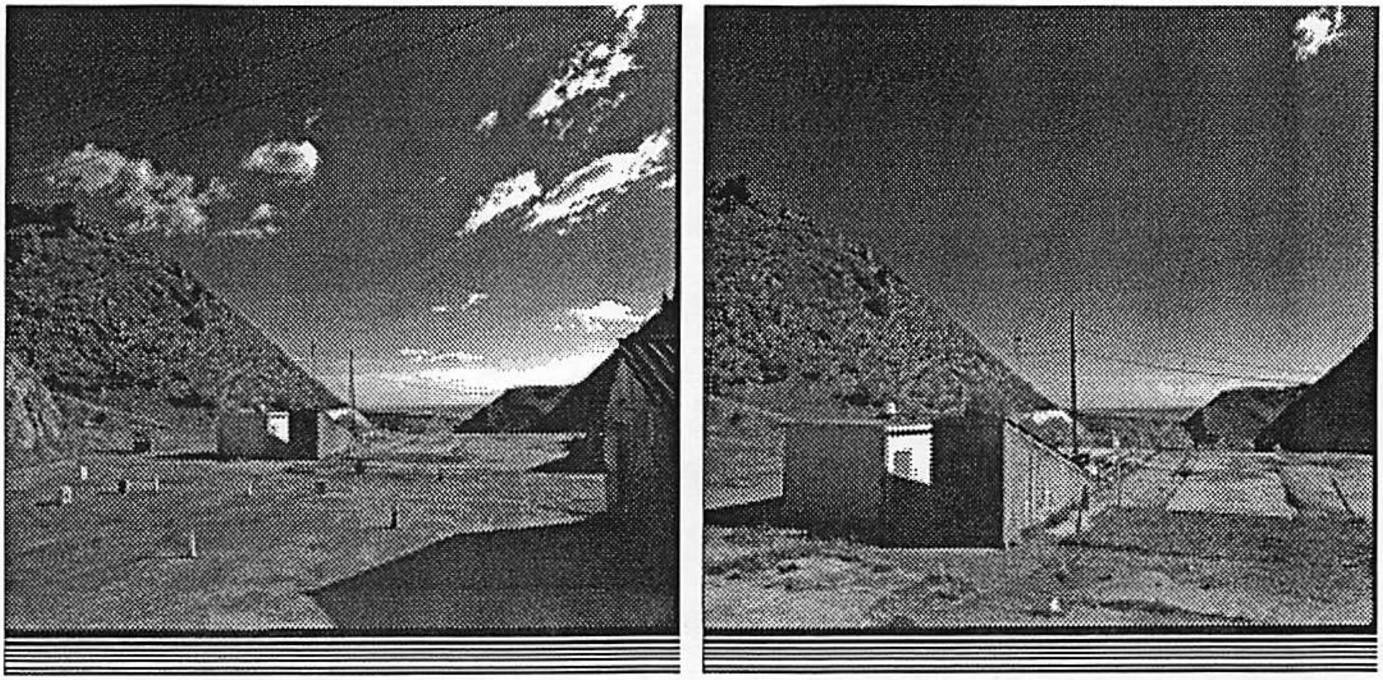

Figure 1: *First and last frames of the real image sequence.*

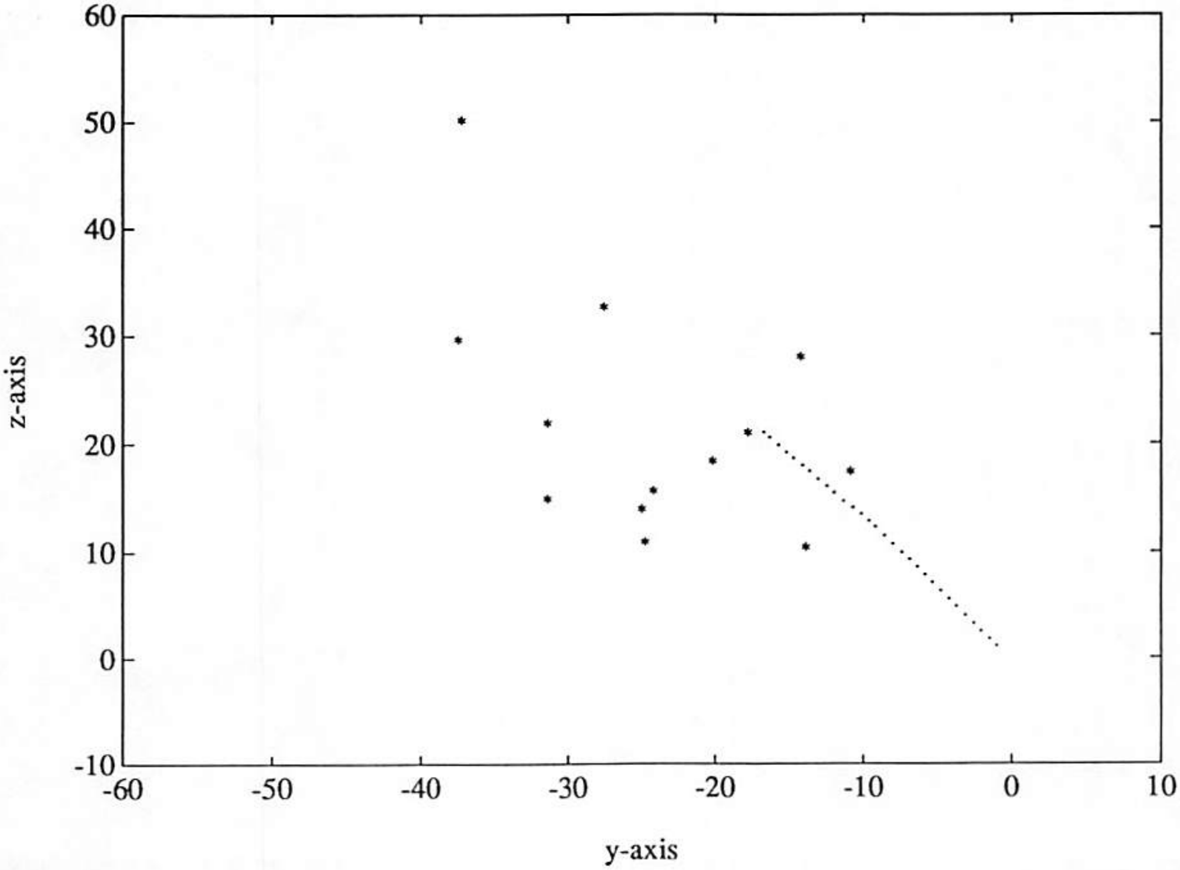

Figure 2: *Aerial view of the environment showing 13 feature points, indicated by '*', and the positions of the camera, indicated by '·'. The axes are scaled in metres.*

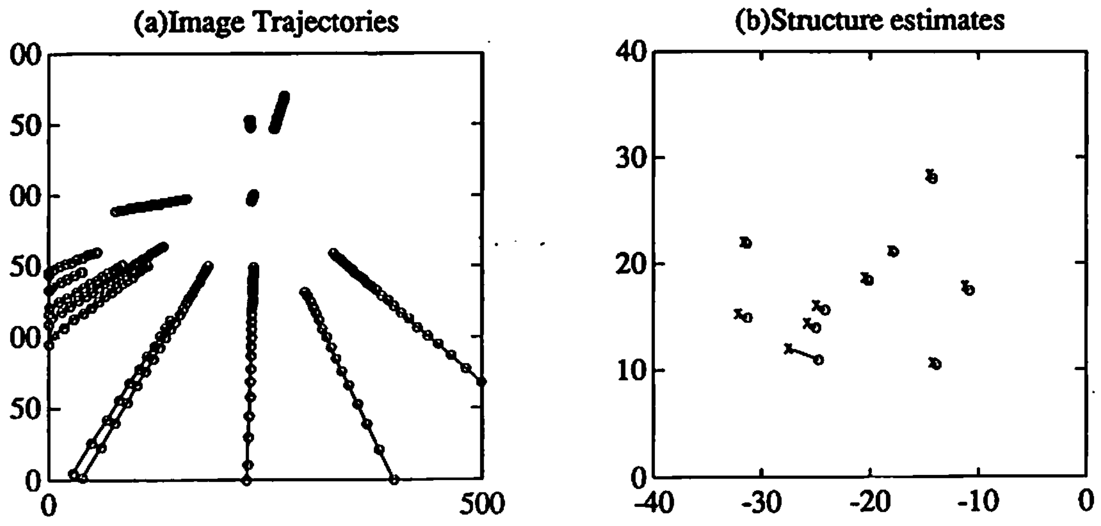

Figure 3: *Synthetic data, example 1. (a) Image plane trajectories of feature points obtained by simulation. The axes scales are in pixels. (b) View from above of the structure estimates obtained by the batch procedure, indicated by "x" and their true values, indicated by "o".*

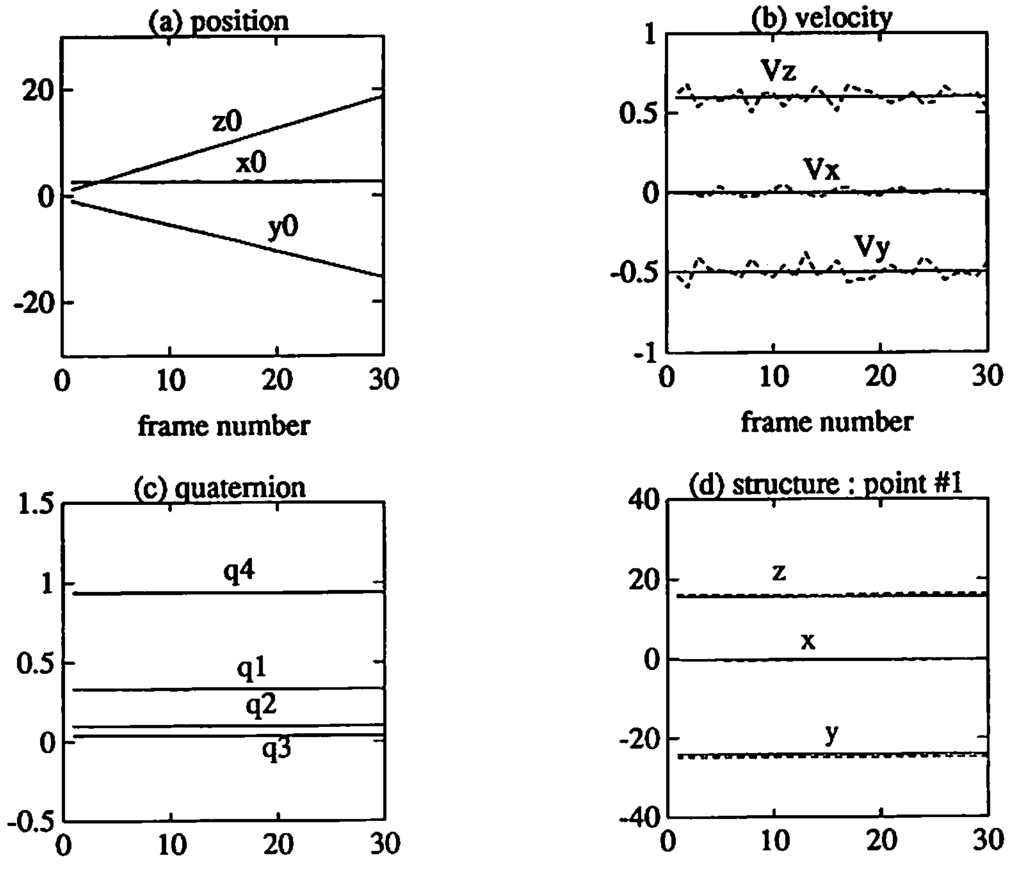

Figure 4: *Actual and EKF-estimated values of some parameters (for synthetic data, example 1). They are shown by solid and dashed lines, respectively. The scales for (a) and (d) are metres, and for (c) metres/frame. The labels on the graphs indicate the corresponding components.*

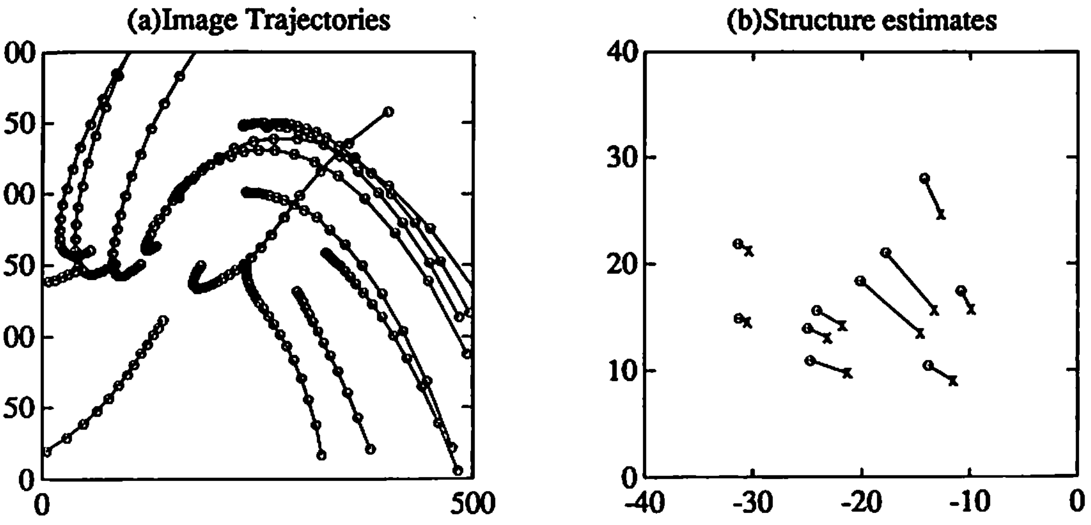

Figure 5: *Trajectories and batch structure estimates (synthetic data, example 2).*

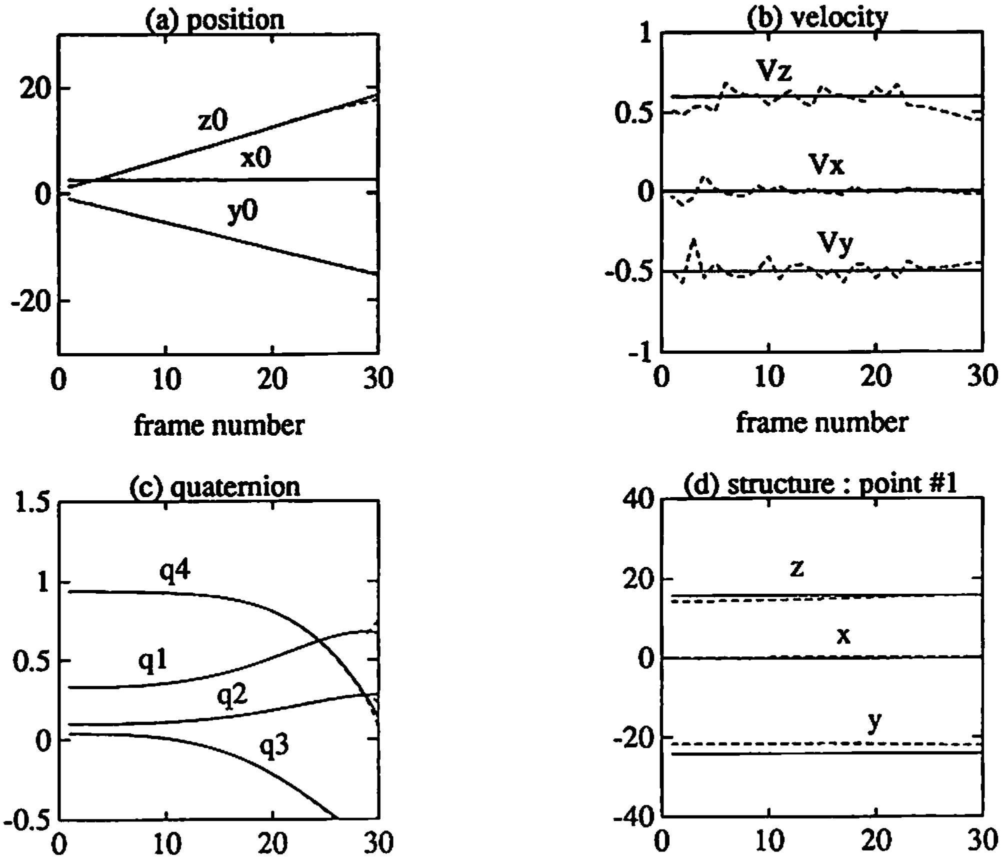

Figure 6: *Actual and estimated values of parameters (synthetic data, example 2).*

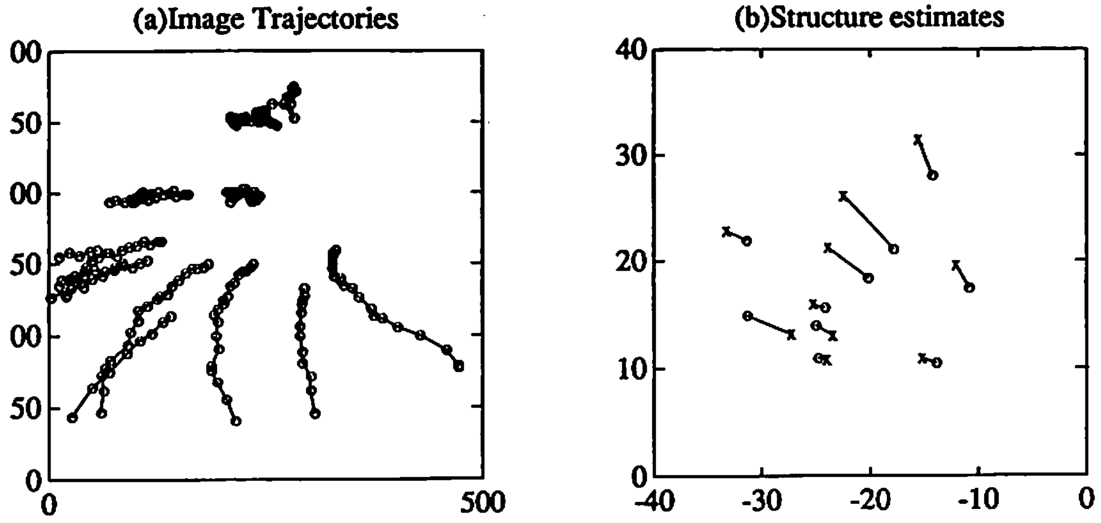

Figure 7: *Trajectories and batch structure estimates, real data.*

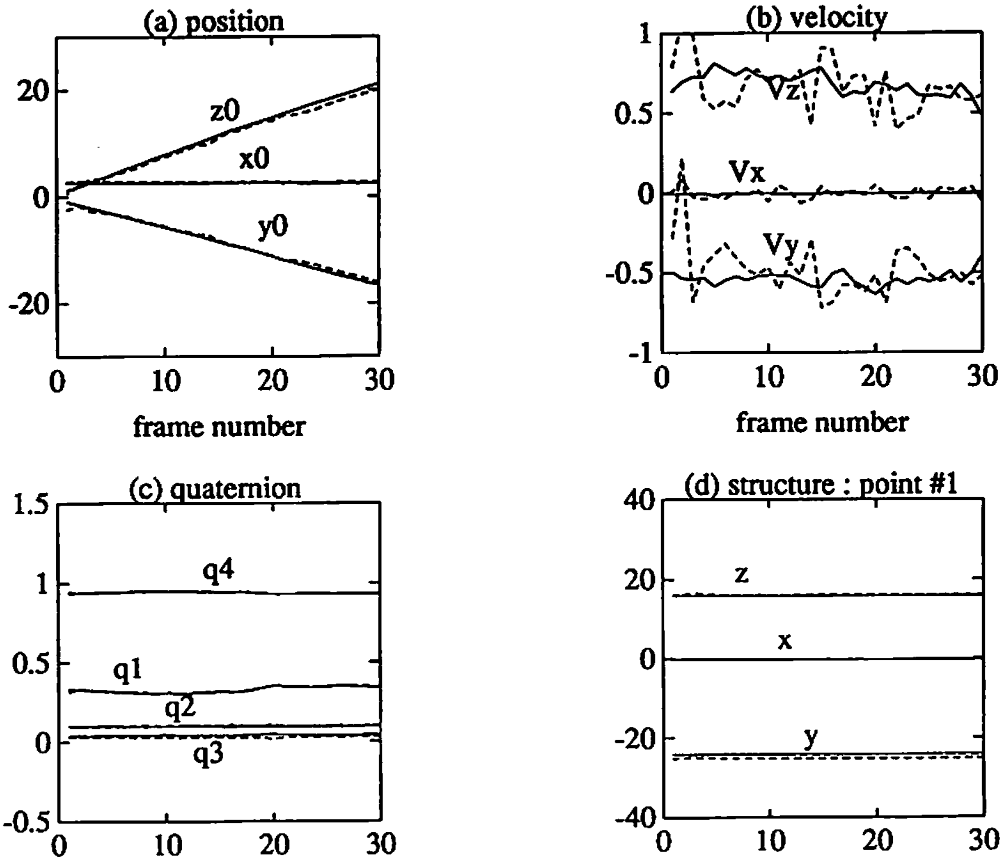

Figure 8: *Actual and estimated values of parameters, real data.*