USC-SIPI REPORT #201

Motion Analysis and Passive Navigation Using Long Image Sequences

by

Chandra Shekhar

May 1992

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
Electrical Engineering Building
University Park/MC-2564
Los Angeles, CA 90089 U.S.A.

To Dr. Y. Venkataramani, my first advisor

Acknowledgments

Behind every successful doctoral student there are several persons who contributed directly or indirectly to his research. My advisor, Rama Chellappa, has been a constant source of encouragement and constructive criticism. I would like to thank him for setting high standards and helping me to attain my full potential for creative research. I would like to thank the other members of my committee, Christoph von der Malsburg, Keith Price and Sandy Sawchuk for their advice and support. I would also like to thank Robert Kalaba, Jay Kuo, John Hauser and Gerard Medioni for valuable advice on several occasions.

The importance of a congenial and lively research environment cannot be understated. In this respect I have been very fortunate, and the company of my fellow graduate students has been an enriching experience. I wish to thank Ted Broida, Gem-Sun Young and Ze'ev Lichtenstein for several illuminating discussions on topics related to my work. I am grateful to Anand Rangarajan for helping me understand the connection between Markov Random Fields and Tibetan Buddhism. I am indebted to V. Venkateswar for his willingness to listen to my troubles, and for helping me handle many difficult situations. I am indebted to B.S.Manjunath and Qinfen Zheng for their willingness to understand my research problems, and to suggest creative (if not always successful) solutions. I wish to thank Suresh Chalasani and Manavendra Misra for helping me prepare my presentations. I am also grateful to the other students and exstudents of SIPI, in particular Yitong Zhou, Jian-Ming Wang, Tal Simchony, Dave March, Navid Haddadi, Xiao-Hong Yan, Li Li Cheng, Doug Wiley, Andrew Miller, Charlie Kuznia, Sabino Piazzolla and Zhenyu Wu for tolerating my wisecracks and in several other ways being excellent company.

One of the most exciting and rewarding periods for me was the summer I spent at INRIA, Sophia Antipolis, in France. I am indebted to Monique Thonnat, Marc Berthod, Aimé Meygret, Philippe Garnesson, Zhengyou Zhang, Jean-Francois Pouzet and Josiane Zerubia, among others, for making me feel at home, and not laughing at my atrocious French. I must mention, though, that they refused, with true Gallic adamance, to give me the recipe for their magic potion.

I wish to thank Allan Weber for valuable help on many occasions. I am grateful to him and to Toy Mayeda for maintaining the SIPI computer systems in an excellent condition.

No work of research is done in isolation; sharing of data and programs between researchers is essential. My work has been greatly facilitated by the contributions of several of my colleagues. I wish to thank B.S.Manjunath, Qinfen Zheng, Rabi Dutta, R.Manmatha, Banavar Sridhar, Haluk Derin, Teddy Kumar, Philippe Garnesson, Zhengyou Zhang, Aimé Meygret and several others for their generosity with their data and software.

I wish to thank the staff of SIPI, Linda Varilla, Mayitta Penoliar, Gloria Bullock and Delsa Tan, for helping me negotiate the sea of red tape every doctoral student has to cross before attaining his goal.

The major part of this work was funded by the University of Southern California's Pre-Doctoral Merit Fellowship and the Office of Naval Research under the grant N00014-89-5-1598. The work on obstacle avoidance was funded by the Eureka project PROMETHEUS.

Contents

List of Figures

List of Tables

Abstract

This dissertation deals with the analysis of visual motion from image sequences, with emphasis on the following three points: (a) Use of long image sequences (b) Use of recursive estimation techniques and (c) Integration of feature matching and motion estimation. The objective is to estimate motion parameters (pose, velocities, etc.) and 3-D structure parameters relating the camera(s) to the scene.

The basic idea is to extract salient points from the image sequence, and to relate their image plane trajectories to the motion and structure parameters. This is accomplished by using simple models for translational and rotational motion, and, for monocular sequences, a central projection model for imaging. The time evolution of unknown parameters is represented as a plant model, and the observation of the data as a measurement model. This state-space representation is suitable for recursive solution. A Kalman filter, or one of its variations, is used to estimate the unknown model parameters. Various methods of initialization are developed, including least-square batch algorithms, linear methods, and iterated filter-smoothers. This approach is then applied to three different situations :

1. Passive navigation : This deals with the case of a single moving camera in a stationary environment.

2. Target tracking : Here the camera is stationary, and the goal is to track the motion of a rigid object within the camera's field of view.

3. Obstacle avoidance : In this application, the situation is more general; a (stereo) camera is moving in a traffic environment containing several moving obstacles, some of them possibly nonrigid.

The algorithms developed are tested on real and synthetic data, demonstrating their robustness in the presence of measurement and modelling errors.

A new method is presented to obtain feature point correspondences from an image sequence, which fits in very neatly with the recursive estimation approach. In this method, Gabor wavelets are used to label salient image points, and point correspondence is treated as a labelled graph matching problem. The matching of feature points is interleaved with the recursive estimation of motion and structure parameters, using the predictive capabilities of the Kalman filter.

The general problem of motion analysis is highly nonlinear, and techniques such as extended Kalman filtering are not guaranteed to be stable. A method is suggested for predicting the qualitative performance of the estimation technique, based on an empirical analysis of the model at selected solution points. By examining the Hessian of the objective function corresponding to a model, properties of the model such as uniqueness of solution, conditioning, etc. are analyzed.

Chapter 1

Introduction

The analysis of visual motion from image sequences ("motion analysis") is one of the most challenging problems in the field of computer vision. Apart from its relevance to the understanding of biological vision systems, motion analysis has a number of practical applications in robotics, vehicle navigation, traffic safety, Intelligent Vehicle and Highway Systems (IVHS), aviation and space exploration, among others. It is not surprising, therefore, that research efforts in this field have increased exponentially in the past two decades, particularly in the last few years. Scores of papers are published annually, covering both the theoretical as well as the practical aspects of motion analysis, and several conferences and workshops are being held to deal exclusively with this area of computer vision.

A wide variety of methods have been developed, each with its own distinct characteristics and claims to superiority over other approaches. Methods based on the computation of temporal and spatial image gradients, also known as "optic flow" methods, have been in existence for several years. The other major category of motion analysis methods consists of the ones based on feature correspondences. In both these categories, literally hundreds of algorithms are currently available. There are also a vast number of techniques that combine motion analysis with other forms of visual processing such as stereopsis, texture analysis, etc.

In spite of this veritable explosion of research activity, a satisfactory method of processing and understanding image sequences—one that is practical, robust and versatile—has proved elusive. Although significant progress has been made in understanding the theoretical nature of the problem, these theoretical results have not, in general, been successfully translated into workable algorithms. Most existing methods suffer from several major drawbacks. Many of them work only if the input data are practically noise free; in most real applications a considerable amount of measurement noise and modelling error is unavoidable. Very few existing methods deal with the motion problem in its entirety, addressing only one of the several steps required to solve it. For instance, many methods based on feature correspondences assume that the latter are given to them by some preprocessing stage. Some are designed to work only under very restrictive conditions, these being an integral part of the assumed model. Such methods may fail in the presence of even minor modelling errors.

This research work was performed with the goal of developing a paradigm for motion analysis which is free from at least a few of these flaws. Existing techniques are combined with new methods to develop a framework for motion analysis applicable to a broad range of situations; three specific applications are discussed in detail, with experimental results on real and synthetic data.

1.1 Statement of the Problem

The goal of this research is to process image sequences to extract information that could be used for various applications such as autonomous navigation and object tracking. This is to be accomplished based on a set of *feature points* extracted and matched over the entire sequence. The parameters we are interested in estimating are the motion and structure parameters relating the camera(s) to the external environment. The specific parameters of interest will depend on the application; for instance, in analyzing an image sequence obtained by a moving camera in a stationary environment, the relevant motion parameters are those related to the translation and rotation of the camera, and

the relevant structure parameters are the 3-D locations of feature points in the scene.

1.2 Overview of the Methodology

In this research, analysis of image sequences is done by a system of several interacting algorithms, constructed using a variety of engineering tools. The salient features of this system are given below.

- **Feature extraction, description and matching:** Motion algorithms based on the computation of image gradients are very sensitive to noise, and are computationally expensive. Although feature-based methods are usually more robust and efficient, they involve additional work in the form of feature extraction and matching. Much of the earlier work in feature-based methods side-stepped these issues, and are therefore of little practical value. In this work we address these issues, and suggest ways of incorporating them into the overall motion analysis system.

  Different kinds of features can be used for motion analysis, such as points, lines, polygons, regions, corners, etc. Points are the simplest kind of features, and can be found in most scenes, natural or man-made. Lines and higher order features are richer in information, and in some cases easier to detect, but may not occur in outdoor and natural scenery. For this reason, we restrict our attention to point features. Point features may be extracted using a variety of interest operators [62, 58]. We use the method described in [47]. Gabor wavelets are used to describe, or "label" feature points, and these points are treated as nodes of a labelled graph. The problem of feature matching is then reduced to one of labelled graph matching. For one of the real image sequences, we use a method developed by Zheng and Chellappa, based on a novel approach to image registration [76].

- **Problem geometry:** The image sequence used as input to the motion analysis system may be generated in several ways. The most general situation

is that of several independently moving cameras in an environment containing multiple independently moving objects. Most motion analysis problems are a subset of this scenario. In Chapter 4, we examine the case of a single moving camera in a stationary environment. In Chapter 6 we look at the converse situation, involving a stationary camera viewing a moving object. In Chapter 7, a far more general situation is dealt with, involving stereo imagery and multiple moving objects.

- Motion models: The idea of using a motion model is to describe the motion using a fixed number of parameters, independent of the length of the image sequence. Thus the data contained in a sequence of arbitrary length can be used to estimate a fixed number of parameters. Greater smoothing can be achieved by using a large number of image frames. The validity of using motion models is based on the fact that in most real applications, the relative motion between the cameras and the scene is smooth, due to the damping and inertia present in all physical systems. Deviations from smoothness can be modelled as system noise. Although a purist would eschew the use of models on the grounds that they impose an artificial smoothness on the estimation process, the rewards of so doing far outweigh the price paid. Furthermore, in a recursive framework, the estimates *adapt* to deviations from the model, and hence the penalties of assuming a model of low dimensionality are negligible.

- Imaging models: Motion models describe the trajectories of feature points in space. These trajectories are usually not observed directly, unless there are two or more cameras. If there is only one camera, we observe the projections of these trajectories onto the image plane. This is done by perspective (or central) projection, a highly nonlinear process. A lot of work on monocular motion analysis is directed towards untangling and simplifying the highly nonlinear equations relating 3-D motion and structure parameters to the image plane locations of features.

- Batch and recursive estimation: Given a 3-D motion model, and the locations of features in the image sequence (i.e. the image plane trajectories

of features), we would like to estimate the unknown model parameters. The estimation techniques which could be used are of two basic types—batch and recursive. In a batch procedure, all the information available about the motion (i.e. all the 3-D measurements) is used together in a single-stage estimation of the motion and structure parameters. This approach is robust and simple, and reasonably fast for a linear problem, 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 are available. Furthermore, a large amount of memory is required to store these measurements. In the recursive approach, the parameters to be estimated are placed in a state vector, which is updated as and when each new data set (which in our case corresponds to the next image in the sequence) becomes available. Data which have already been processed are discarded. This approach is is much faster and requires much less storage than the batch approach. However, unlike batch algorithms, recursive algorithms usually need to be supplied with an initial guess before they can start processing the incoming data. 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 batch estimates thereby obtained to initialize the recursive estimator. Then, for the remainder of the sequence, the recursive estimator can be used to "track" the motion. Other methods of initialization, such as linear two-frame methods, can also be used if only a rough initial guess is required.

- Interleaving estimation and correspondence A major advantage with recursive techniques such as Kalman filtering is the ability to predict feature positions in space, and hence in the image, ahead of time. This capability can be used to reduce the search space for match points in the incoming images. We have successfully integrated the predictive capabilities of the recursive filter with the algorithm used to find the correspondence between features in successive images. This is described in Chapter 3.

1.3 Applications

Motion understanding is a vital part of most biological vision systems. Many animals rely on their ability to detect movement to find their prey and to evade predators. Frogs, for instance, have a very poor sense of static vision, but are able to catch insects by visually detecting their movement. Visual motion understanding is also useful in primate survival, although in this case it is not the primary sensory modality. The ability to perceive and interpret motion is essential to many human activities like driving, operating machinery, sports, etc. Thus from a biological point of view, visual motion understanding is a topic of great relevance in the development of intelligent robots with human-like capabilities. In the field of computer vision, however, we do not restrict ourselves to biologically relevant models, although we do use the knowledge gained by studying human motion perception. We are also interested in specific applications requiring capabilities which biological systems may not possess, or which do not require their degree of sophistication. Biological systems are inherently qualitative in the way they process information, and the information processing is directed towards specific goals like grasping an object or fleeing from an approaching predator. In certain applications, like target tracking, it may be necessary to precisely estimate physical parameters of an object's motion, and to determine its structure. In these cases, the computations have to be quantitative in nature. In this dissertation, we look at the motion problem from a signal processing standpoint, rather than a biologically motivated one.

Motion analysis is finding applications in many areas. Applications in robotics include locating, identifying and grasping objects using visual information. Another important application is visual navigation for autonomous vehicles. This dissertation examines this problem in detail, and presents a solution which incorporates the principles outlined in the previous section. Although this "passive navigation" problem is the main focus of this work, two other applications are also addressed: target tracking and obstacle avoidance.

In passive navigation, the goal is to utilize the information contained in an image sequence to determine the motion of a moving camera, and the structure of the environment around it. In target tracking, the camera is assumed to be

stationary, and a single moving object is assumed to be present within the field of view of the camera. The objective here is to determine the kinematics and structure of the object. Target tracking has applications in defence, commercial aviation, space exploration, etc. The problem of obstacle avoidance is related to the navigation problem, but is slightly different in the sense that no landmarks, lane boundaries, etc. may by assumed to be present, and the obstacles may themselves be in motion. This has applications in traffic safety.

1.4 Original Contributions

The original contributions of this research are as follows:

- Estimation theoretic methods are applied successfully to several motion analysis applications.
- New motion models are developed for two applications: passive navigation and obstacle avoidance.
- Previous work on object tracking [9] is extended to the case of 3-D motion and structure estimation.
- Gabor wavelets are used for feature correspondence.
- Motion estimation is interleaved with feature correspondence.
- A new method of analyzing model-based formulations is presented.

1.5 Organization of this Dissertation

Chapter 2 reviews some of the published work in the field of motion analysis, concentrating on long-frame, model-based and recursive approaches. A new method of obtaining feature point correspondences is presented in Chapter 3. The passive navigation problem is discussed in Chapter 4. A method of analyzing and evaluating model-based formulations is presented in Chapter 5. The target tracking problem is examined in Chapter 6. Chapter 7 discusses

the obstacle avoidance problem. Directions for future research are presented in Chapter 8.

Chapter 2

Review of the Literature

Feature-based motion estimation techniques have become very popular in the past few years, and the active research that has been going on in this area has resulted in a large body of literature. Considerable research effort also goes into the so-called optic flow methods, which, in contrast to feature-based approaches, rely on computations of a dense image flow field. Although this research is based solely on feature correspondences, it is appropriate to mention some of the key work in optic flow, such as [5, 1, 39, 74].

Much of the early work on feature-based motion analysis was focussed on the two-frame motion problem. The goal was to look at just two images from a sequence, and to determine the coordinate transformation (motion) of the camera(s) between the two time instants, and the 3-D structure of selected feature points. If a monocular pair of images is used, these quantities can be determined only upto a scale factor. Preliminary work on two-frame methods was done by Roach and Aggarwal [57] and Tsai and Huang [63], among others. Further progress was made by Yasumoto and Medioni [70], Fang and Huang [27], Aggarwal and Mitiche [2]. These methods had the advantage of being model-free, in general, but they were very sensitive to measurement errors. It was found that only a slight improvement could be obtained by increasing the number of points.

Several new trends have emerged in this field in recent years, such as the use of motion models [66, 15, 61, 30], and estimation theoretic methods

[34, 24, 19, 4, 42, 65, 38, 20]. The use of motion models simplifies motion analysis by reducing the number of parameters to be estimated and allowing the use of image sequences of arbitrary length. Sensitivity to measurement noise and feature point occlusions can be reduced by incorporating the information contained in long image sequences. Batch and recursive estimation techniques can be used together for real-time motion analysis and tracking applications. These approaches can be combined to yield a powerful paradigm for visual motion analysis, which is simple, robust, flexible and efficient. The work presented in this dissertation is based on this paradigm.

In the rest of this chapter, we shall discuss significant contributions to long-frame motion analysis, classifying the methods as applicable to passive navigation or to target tracking. Some of the methods presented below may be applicable to both situations, but will be discussed only in the context of one of them.

2.1 Passive Navigation

Possibly the most exciting application of motion analysis is to the visual navigation of autonomous vehicles. With the growing interest in Intelligent Vehicle and Highway Systems (IVHS), vision-based systems are being seriously considered for traffic safety applications. Some of the leading research efforts in this area are discussed below.

At the University of Munich, Germany

Pioneering work in the navigation of outdoor vehicles was done by Dickmanns, Graefe et al [23, 24]. Their work is based on a 4-D model of the world (with time being the fourth dimension). They use integrated spatio-temporal models, monocular image sequences and recursive estimation techniques in a control framework. They have developed a microprocessor-based system for motion control, and successfully used it to navigate a motor vehicle on a freeway, under certain controlled conditions such as the presence of distinct road boundaries. Their spatio-temporal approach can also be tailored to other applications such as the balancing of an inverted pendulum. Their work is definitely one of

the more successful applications of recursive techniques to practical motion analysis problems. In their published work, however, they do not analyze in detail the dynamic performance of the algorithms they use.

At Honeywell Systems and Research Center, U.S.A

Roberts, Bhanu and others have recently developed a state-of-the-art motion analysis system [58] which incorporates inertial measurements into the motion estimation framework. They select feature points in the image sequence using a combination of Laplacian and Hessian operators, and match them using a variety of metrics. These matches are combined with inertial measurements of linear and angular velocities to produce sparse range measurements, which are interpolated to obtain a complete range map. This is used to navigate the vehicle. They demonstrate their system on indoor and outdoor imagery.

At Plessey Research and Technology, UK

The "DROID" vision system developed by Stephens, Harris, Charnley et al. [62] is designed to test the limits of performance that can be attained using a purely passive system. It uses feature-based structure from motion principles to from an approximate depth map of the environment of the moving vehicle. Rough estimates of the camera's ego-motion which are obtained by inertial sensors are refined using 3-D point matches obtained using stereo and temporal matching. Sparse range measurements obtained by binocular and motion stereo are interpolated by Delaunay triangulation. The navigation of the vehicle is now guided by the resulting depth map. The researchers are currently working on motion segmentation and path planning.

At INRIA, France

Faugeras, Deriche, Ayache, Zhang et al. have developed a vision system for the INRIA mobile robot, based on trinocular stereo and line correspondences [3, 4, 75]. They are primarily interested in indoor environments, where a large number of line features may be assumed to be present. Trinocular stereo is shown to yield more robust 3-D measurements than the traditional binocular stereo. They use the Extended Kalman Filter as the main mechanism to optimally combine various noisy measurements of egomotion and 3-D structure

to form and maintain a cumulative representation of the environment of the mobile robot.

At the University of Massachusetts, Amherst: Kumar and Hanson [43] use structure from motion principles to estimate the pose of a camera, and to incrementally refine it using a sequence of images. They have developed robust techniques capable of handling gross errors in the data. The pose estimation is based on a set of recognized landmarks (3-D lines) appearing in the image under perspective projection. They have used their algorithm on both indoor and outdoor scenes. In [43] they analyze the sensitivity of the pose estimates to accurate estimation of camera parameters. Dutta, Manmatha et al. have created a database of real image sequences with complete pose and structure ground truth [25]. The sequences were obtained from a camera attached to a land vehicle moving on different kinds of surfaces such as metalled road, unpaved road, etc. Different kinds of environments are also considered, ranging from those with several man-made objects to purely natural scenes. They have discussed various issues relating to the extraction of motion and structure parameters from approximate translational motion—the type of motion most commonly encountered in autonomous land vehicle (ALV) applications [48].

2.2 Target Tracking

Target tracking has been an area of very active research for several decades. Until about ten years ago, most of the research focussed on radar-based tracking, with defence applications [7]. With the advances made in image processing in recent years, visual tracking is receiving a lot of attention. In this section, we shall discuss the work of Gennery [34], Shariat [60], Weng, Huang and Ahuja [66], Broida [9], Young [71], Franzen [30] and Heel [38].

Gennery was one of the first researchers to work in this area. His experiments on tracking known 3-D objects are reported in [34]. Gennery used a Kalman-like recursive filter, line features and incorporated feature prediction into his system.

Shariat investigated the use of more than two frames for motion analysis, using the assumption that the motion of the object is approximately constant. He estimates the translation vector, axis of rotation and amount of rotation, given the correspondences of 1 point in 5 frames, 2 points in 4 frames, etc. He develops a different algorithm for each case. He uses the rigidity constraint explicitly to reduce the motion estimation problem to one of solving a set of polynomial equations, which he does using an iterative approach. Convergence is accelerated using an "initial guess generator."

Broida [9] has used batch and recursive techniques to estimate the 3-D motion and structure of a rigid object, based on point feature correspondences form a monocular image sequence. He uses the assumption of smooth motion, like Shariat. His method is capable of handling an arbitrary number of points and frames, unlike that of Shariat. Although his experiments deal with motion of constant linear and angular velocities, his general approach is applicable to smooth motion of any degree of complexity. He uses an estimation theoretic approach to estimating the unknown model parameters from the noisy input data. He proposes a two-step approach to the problem, using a batch approach on the first few frames in the image sequence , and using a recursive approach over the remaining frames in the sequence, with the batch estimate being used as an initial guess for the recursive estimator. He uses a maximum likelihood approach for batch estimation, and an Iterated Extended Kalman Filter (IEKF) for recursive estimation. He suggests the use of approximate Cramér-Rao lower bounds to predict the performance of the estimator.

Young developed an approach similar to that of Broida, applicable to stereo data [72, 73]. The assumption of stereo data, and the consequent avoidance of the nonlinearities of perspective projection enables the use of higher-order motion models. Whereas Broida's experiments assumed constant translational and rotational velocities, Young demonstrates his algorithm on motion involving constant acceleration and constant precession. He also does not require a batch initialization step, since his recursive procedure converges rapidly from an arbitrary initial guess. He gives a constructive proof for the uniqueness

of motion parameters. He shows that for uniform sampling rate, three non-collinear feature points in five consecutive frames contain all the information necessary to uniquely determine the motion parameters.

Weng, Huang and Ahuja have proposed a Locally Constant Angular Momentum (LCAM) model for long-frame object motion, based on rigid-body dynamics. Their model constrains the motion, over short intervals of time, to be a superposition of precession and translation. It allows the instantaneous axis of rotation to change from frame to frame. They have developed a linear algorithm for the problem which can handle an arbitrary number of points and frames.

Franzen has proposed a batch method applicable to "chronogeneous" motion [30], which includes uniform acceleration and constant angular velocity rotation as special cases. He develops a closed-form method to recover structure of features undergoing known affine interframe transformations. He then uses this method to factor out the structure parameters in the case of unknown motion, reducing the problem to one of determining the motion parameters alone. He proceeds to solve this using iterative techniques.

Heel has suggested combining two-view motion estimation, based on image gradients, with recursive filtering on the depth estimates to obtain dense structure estimates of the object. Since he uses two-view motion estimation, he does not require a motion model. He estimates the motion parameters by relating the image intensity gradients directly to the 3-D motion parameters, without requiring optic flow or feature correspondences.

Chapter 3

Feature Point Matching

In this chapter, we present a scheme for obtaining the trajectories of points of interest (feature points) in a long sequence of monocular images. We combine the method of recursive filtering used for motion analysis with the technique of labelled graph matching as applied in [16, 17] for obtaining feature correspondences for pattern recognition.

The problem of trajectory estimation is formulated as a recursive tracking problem based on a plant model and a measurement model. The parameters relating to the position and motion of the points are contained in a state vector, whose time-evolution is represented by the plant model. In this chapter, a linear model is assumed for the motion of the feature points in the image plane, each point being treated individually. The measurement vector contains the observed feature point positions on the image. The observation model represents the relationship between the state vector and the observations. Together, the two models constitute a state space representation of the problem, suitable for solution with a linear Kalman filter. In later chapters, 3-D models will be used in place of the simple 2-D model used here.

Feature point correspondence is posed as a labelled graph matching problem, with the feature points treated as nodes of a labelled graph. This technique has been applied in [16, 17] for face recognition, with encouraging results. The problem of trajectory analysis is somewhat similar to the object recognition

problem in the sense that both usually require a correspondence between distinct features in two or more images, or between a stored pattern and a test pattern. In both cases, labelled graph matching provides the required invariance to limited amounts of distortion, unlike correlation-based methods which are known to be sensitive to distortion.

The performance of the labelled graph matching procedure depends on the manner in which feature points are selected and labelled. The images in the sequence have to be suitably processed to extract features and to obtain a rich description (labelling) of the features so that the probabililty of errors in matching is reduced. A feature point can be labelled in several ways, based on the intensity distribution in a neighbourhood around it. Image gray levels cannot be used directly, since they do not usually remain constant over significant time intervals. A better approach is to use wavelet-type oriented feature detectors, which are less sensitive to illumination changes, and have several desirable properties such as variable resolution and optimal localization in the spatial and frequency domains [21, 56]. In this work Gabor wavelets (sometimes called Morlet functions [37]) are used to label feature points.

The matching of feature points is interleaved with the recursive estimation of trajectories. Current information about the trajectories is used to predict the future positions of feature points, thereby reducing the search time for finding match points. Feature points are not assumed to have already been extracted in all the images in the sequence; instead, selected feature points in the first image in the sequence are "tracked" over successive images in the sequence by labelled graph matching between consecutive image frames. Thus feature point extraction in all images but the first is done automatically. Results on synthetic and real image sequences are presented.

3.1 Formulation for Recursive Solution

The idea here is to formulate the estimation of a point's trajectory as a recursive tracking problem, based on a plant model and a measurement model. In this chapter, no 3-D model is assumed for the motion; we simply assume that the

image plane trajectories of feature points are smooth. Since no 3-D model is available, each point is treated separately. In later chapters, we will develop specific 3-D models for different situations, and use these models instead of the model developed in this section.

The motion of a point $\mathbf{p} = (x, y)^T$ in the image is modelled by the following equation:

$$\mathbf{p}(t) = \mathbf{p}(0) + \dot{\mathbf{p}}(0)\, t + \ddot{\mathbf{p}}(0)\, t^2/2! + \cdots + \mathbf{p}^{(n)}(0)\, t^n/n!, \tag{3.1}$$

where where n is small compared to the number of frames in the sequence. (In other words, it is assumed that derivatives of $\mathbf{p}(t)$ higher than the nth are negligible for all t.) Typically, one would select $n = 1$ or $n = 2$.

The quantities to be estimated i.e. the position of the point and the derivatives thereof are contained in a state vector $\mathbf{s}$, defined by

$$\mathbf{s} \triangleq \begin{bmatrix} \mathbf{p}(t) \\ \dot{\mathbf{p}}(t) \\ \ddot{\mathbf{p}}(t) \\ \vdots \\ \mathbf{p}^{(n)}(t) \end{bmatrix} \tag{3.2}$$

The plant model describes the time evolution of the state vector. Using (3.1), which expresses the assumption that $\mathbf{p}^{(m)}(t) = 0 \,\forall\, m > n$, it can be written as

$$\dot{\mathbf{s}}(t) = \mathcal{F}\mathbf{s}(t) + \mathbf{w}(t) \tag{3.3}$$

where $\mathbf{w}$ is a noise term included to take into account modelling errors, and the matrix $\mathcal{F}$ is of the form

$$\mathcal{F} = \begin{bmatrix} \mathbf{0} & I_2 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \ddots & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & I_2 \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \end{bmatrix} \tag{3.4}$$

where I_2 is the 2×2 identity matrix. Equation (3.3) has to be discretized, in order that it can describe the evolution of the state vector from one sampling instant to another. The discrete version of the plant model can be found by integration of (3.3) over the sampling interval (i.e. interframe period), and the result is

$$\mathbf{s}(k) = F\,\mathbf{s}(k-1) + \mathbf{w}_k \tag{3.5}$$

where the matrix F is given by

$$F = \begin{bmatrix} I_2 & t\,I_2 & \frac{t^2}{2!}I_2 & \cdots & \cdots & \frac{t^n}{n!}I_2 \\ 0 & I_2 & t\,I_2 & \frac{t^2}{2!}I_2 & \cdots & \frac{t^{n-1}}{(n-1)!}I_2 \\ \vdots & \vdots & \ddots & & & \vdots \\ 0 & 0 & \cdots & \cdots & I_2 & t\,I_2 \\ 0 & 0 & \cdots & \cdots & \cdots & I_2 \end{bmatrix} \tag{3.6}$$

The input data to the estimator is in the form of image point positions, one for each sampling instant. The measurement model, which shows the relationship between the state vector $\mathbf{s}$ and the observation (measurement) vector $\mathbf{z}$ is given by

$$\mathbf{z}(k) = H\,\mathbf{s}(k) + \mathbf{v}_k \tag{3.7}$$

where

$$H = \left[I_2 \vdots 0 \vdots 0 \vdots \cdots \vdots 0 \right] \tag{3.8}$$

The problem formulation is now ready for recursive solution by a Kalman filter. The relevant equations are given in Appendix A.

3.2 Feature Point Labelling

In our implementation, we use a form of wavelet decomposition called Morlet transform [37] converting the image into a "resolution pyramid" by filtering with Gabor-like kernels of the form:

$$G(\mathbf{x}, \mathbf{k}) = e^{(i\mathbf{k}\cdot\mathbf{x})}\, e^{\left(-\frac{\|\mathbf{k}\|^2\|\mathbf{x}\|^2}{2\sigma^2}\right)} \tag{3.9}$$

Each kernel has two vector arguments, 2-D position ($\mathbf{x}$) and wave number ($\mathbf{k}$). The filtering consists of convolving the image $I(\mathbf{x})$ with the kernels $G(\mathbf{x}, \mathbf{k})$ i.e.

$$f(\mathbf{x}, \mathbf{k}) = \int I(\mathbf{x}')G(\mathbf{x} - \mathbf{x}', \mathbf{k})d\mathbf{x}'. \tag{3.10}$$

By varying $\mathbf{k}$ in magnitude and orientation (phase), we get a vector of labels $f(\mathbf{x}, \mathbf{k})$ at each pixel $\mathbf{x}$. The magnitude represents local spatial frequency (i.e. resolution) of the feature detector. In our experiments, we found it sufficient to use four levels of resolution, and four orientations at each level. Since the kernels consist of complex numbers, this results in a label vector of 16 complex numbers for every point in the image. This vector will be referred to as a jet.[1] Matching two points can now be done by comparing their jets, and can be accomplished with greater reliability than what would be possible by using image intensities alone.

This method also provides a means for automatic selection of points of interest. For example, they can be extracted using some kind of thresholding criterion on the magnitude of the label vector at each image point. The threshold has to be adaptively chosen from the given image, with the idea of obtaining the desired number of feature points. Another approach is described in [52]. In this chapter, feature points are chosen by inspection in the first image, and are "tracked" over the entire sequence, thereby automatically performing feature point extraction in succeeding images.

3.3 Feature Point Matching

Feature point matching between two images I_1 and I_2 is performed using the principles of labelled graph matching, which have been successfully applied in [17] for performing distortion-invariant pattern recognition. Let us assume that feature points in I_1 are available, and that we wish to find the matching points in I_2. The feature points are treated as nodes in a labelled graph, with the (vector) labels being the jets obtained by convolution with the Gabor wavelet

[1] In [16, 17], the term "jet" is used to refer to a vector containing the magnitudes of the convolved outputs, rather than the complex-valued outputs themselves.

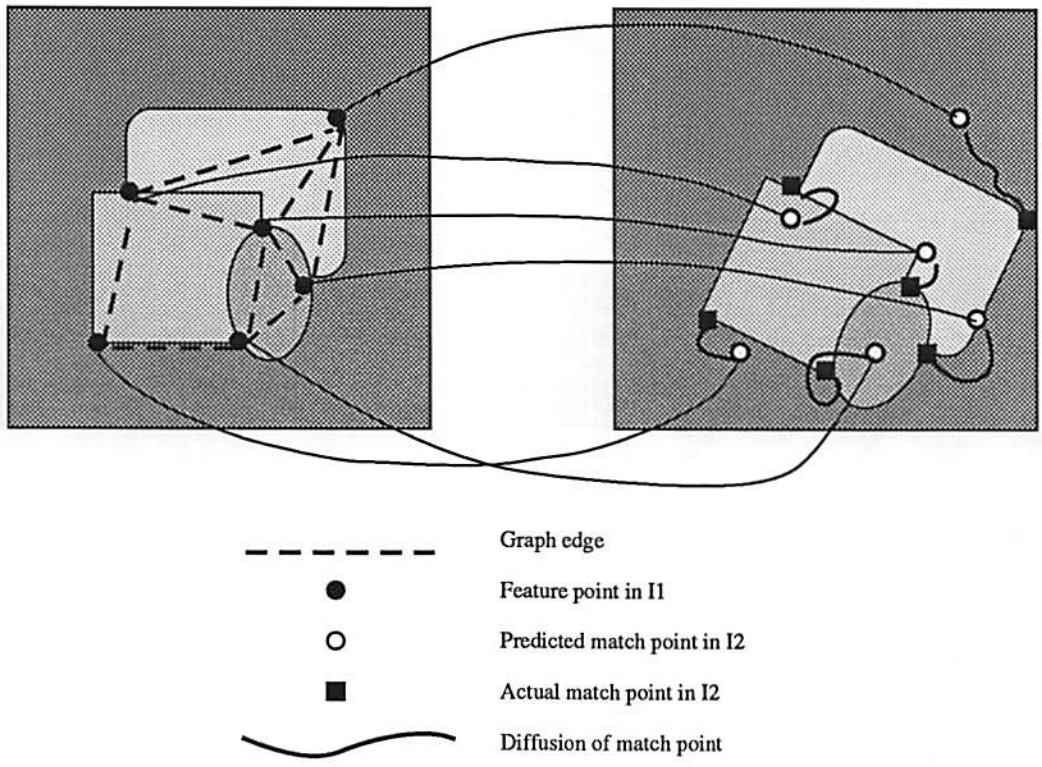

Figure 3.1: Labelled graph matching applied to motion correspondence

kernels. Neighbouring feature points in I_1 are linked to form a topological graph. This can be done automatically, using the interpoint distance as a basis for linkage; in our work this is done by inspection. Matching then consists of dynamically assigning image points in I_2 to the given feature points in I_1. This assignment is guided by three criteria (a) the similarity of the jets of potential match points (b) the preservation of the local topology of the graphs of the feature points in the two images and (c) the nearness of the match points to their *predicted* locations. The matching is be treated as minimization of an energy function of the form

$$\sum_i [C_S(i,i') + \alpha C_T(i,i') + \beta C_D(i,i')] \tag{3.11}$$

where $C_S(i,i')$ is the " similarity cost" , $C_T(i,i')$ is the "topological cost" determined locally for the ith feature point, and $C_D(i,i')$ is the "diffusion cost." The terms α and β are weighting parameters. The prime on the second argument of the functions (i') refers to fact that it is the match point in the second image corresponding to the ith point in the first image. These costs may be computed in many different ways. Some discussion of this can be found in [16, 17]. For instance, we could use the following cost functions:

$$C_S(i,i') = \sum_i [\|jet(i) - jet(i')\|^2] \tag{3.12}$$

$$C_T(i,i') = \sum_{j \in N(i)} [d(i,j) - d(i',j')]^2 \tag{3.13}$$

where $N(i)$ is the set of neighbours of point i, and $d(.,.)$ measures the Euclidean distance between image points. Computation of the diffusion cost will be explained in the next section. If the C_S or C_D cost terms for a point are inordinately high after minimization, it is assumed that the point has been "lost" due to occlusion or other causes, and it is not tracked any further.

The discussion so far has assumed that the matching is done at all resolutions simultaneously, i.e. in a non-hierarchical manner. Indeed, this is the way in which it is done in [16, 17]. We have found that greater speed and accuracy can be achieved by a hierarchical approach, using the matches obtained

at coarser levels to guide the matches at finer levels. Here "coarse" and "fine" refer to the magnitude of the spatial frequency of the Morlet kernel, the highest frequency corresponding to the finest resolution. At coarser levels, the search is conducted in a larger neighbourhood, sampling it sparsely, and at finer levels in a smaller, densely sampled neighbourhood.

3.4 Interleaving Motion Estimation and Correspondence

The matching process described above will require that the location of match points in I_2 be known approximately to start with. If this is not the case, the search region for match points will be very large, resulting in extremely high computation time, and highly increased probability of false matches. This is where the strength of the recursive filtering approach lies; given the current state vector $\mathbf{s}_i$ corresponding to ith feature point, the position of the point can be predicted at future time instants with a *known* uncertainty (or confidence). To be precise, the position of the feature point in the incoming image can be predicted as:

$$\hat{\mathbf{p}}_i(k|k-1) = H\ F\,\hat{\mathbf{s}}_i(k-1|k-1). \tag{3.14}$$

(The notation is explained in Appendix A.) The covariance, or uncertainty, of this prediction can be shown to be:

$$C_i(k/k-1) \triangleq Cov(\hat{\mathbf{p}}_i(k|k-1)) = H(F\,P_i(k-1|k-1)\,F^T + Q_k)H^T. \tag{3.15}$$

The predicted feature point positions can then be used to initialize the matching process, and the covariance of the prediction to control the "diffusion" i.e. search of match points during the matching. In other words, the search for a match point is conducted in a region around its predicted location, the size of this region being proportional to the uncertainty of the prediction. Further, the diffusion cost term C_D in (3.11) is chosen so as to favour matches close to their predicted locations. To be precise, it is selected to be the sum of the

"Mahalanobis" distances from the predicted locations, i.e.

$$C_D(i) = \sum_i (\hat{\mathbf{p}}_i(k|k-1) - \mathbf{z}_i(k))^T C_i(k/k-1)^{-1} (\hat{\mathbf{p}}_i(k|k-1) - \mathbf{z}_i(k)) \quad (3.16)$$

The matching procedure yields the measurement $\mathbf{z}_i(k)$, which is then used to perform a "measurement update" on the state vector $\mathbf{s}_i$. This is done for all the feature points. The system is then ready to process the next image in the sequence.

3.5 Experimental Results

The method was tested on a number of image sequences, both real and synthetic, with different patterns of image motion. Results for one synthetic and three real image sequences are discusses in this section. In the motion model, a value of $n = 1$ was used, corresponding to constant image-plane velocities for the feature points. This assumption is not very restrictive, since a well-designed recursive algorithm has the ability to track the states even in the presence of model deviations. As mentioned earlier, the topological graphs are created by inspection. We have experimented with different topologies for a given set of points, and the results indicate that the algorithm is not very sensitive to the topology, as long as a reasonable number of connections (at least two for each point) are maintained. We have also tried different values of the parameters α and β in (3.11). Good performance was obtained over a fairly wide range of values of these parameters. In general, it is difficult to predict in advance the optimal values for α and β for a given image sequence. Some knowledge of the true motion is required in order to decide which of the three components of the cost function should be favoured over the other two.

The results for the synthetic sequence are shown in Fig. 3.2. The 1st, 3rd and 6th frames of the sequence are shown in Fig. 3.2(a), (b) and (c). The sequence was generated by moving a geometric pattern against a uniform background. The motion of the points is not along straight lines, but along curved trajectories. The graph topology used for matching is shown in Fig. 3.2(d).

The images are completely noise-free, and the feature points are clearly distinguishable, and can be accurately localized. It is not unreasonable to expect perfect matching under such a "best-case" scenario. This is indeed the case, as the results in Fig. 3.2(e) and (f) indicate. The trajectories determined by the method are shown superimposed on the first and last images in the sequence, to enable the observer to validate them.

The results for the real image sequences are shown in Figs. 3.3 - 3.5. The first sequence (called the "robot-arm" sequence), consisting of ten 512x512 images, was taken by a camera mounted on a PUMA robot arm. The 1st, 5th and 10th frames from this sequence are shown in Fig. 3.3(a), (b) and (c). Its motion is approximately a rotation around the optical axis of the camera. The scene is the interior of a room with several objects, and several polygonal patterns on the floor, walls and ceiling. The corners of these polygons are ideal choices for feature points, since they are precisely localizable, and can be expected to have Gabor profiles easily distinguished from those of neighbouring points. The selected feature points and the topological graph are shown in Fig. 3.3(d), and the resulting trajectory estimates are shown in Fig. 3.3(e) and (f), superimposed on the 1st and 10th frames, respectively.

The second real sequence (called the "coke-can" sequence), containing 16 images of dimensions 512x512, was obtained by a camera moving forward along its optical axis. The original sequence has 151 closely sampled image frames, from which we selected every 10th image. The 1st, 8th and 16th frames from the sampled sequence are shown in Fig. 3.4(a), (b) and (c). The scene is the top of a table with several small objects on it. The features in this case are not so well defined. For instance, the tops of the pencils have been chosen as feature points, although they are several pixels wide. This is not seem to be a serious problem, because the multi-scale nature of the Gabor wavelet representation makes it possible to treat extended features as point features, and match them within the same framework. The selected feature points and the topological graph are shown in Fig. 3.4(d), and the resulting trajectory estimates are shown in Fig. 3.4(e) and (f), superimposed on the 1st and 16th frames, respectively.

The third real sequence, known as the UMASS Rocket sequence, was obtained from a camera moving with approximately constant orientation and axial translation. In this case, a 3-D motion model (explained in the next chapter) was used. The 1st, 8th and 16th frames from the Rocket sequence are shown in Fig. 3.5(a), (b) and (c). The selected feature points and the topological graph are shown in Fig. 3.5(d), and the resulting trajectory estimates are shown in Fig. 3.5(e) and (f), superimposed on the 1st and 16th frames, respectively.

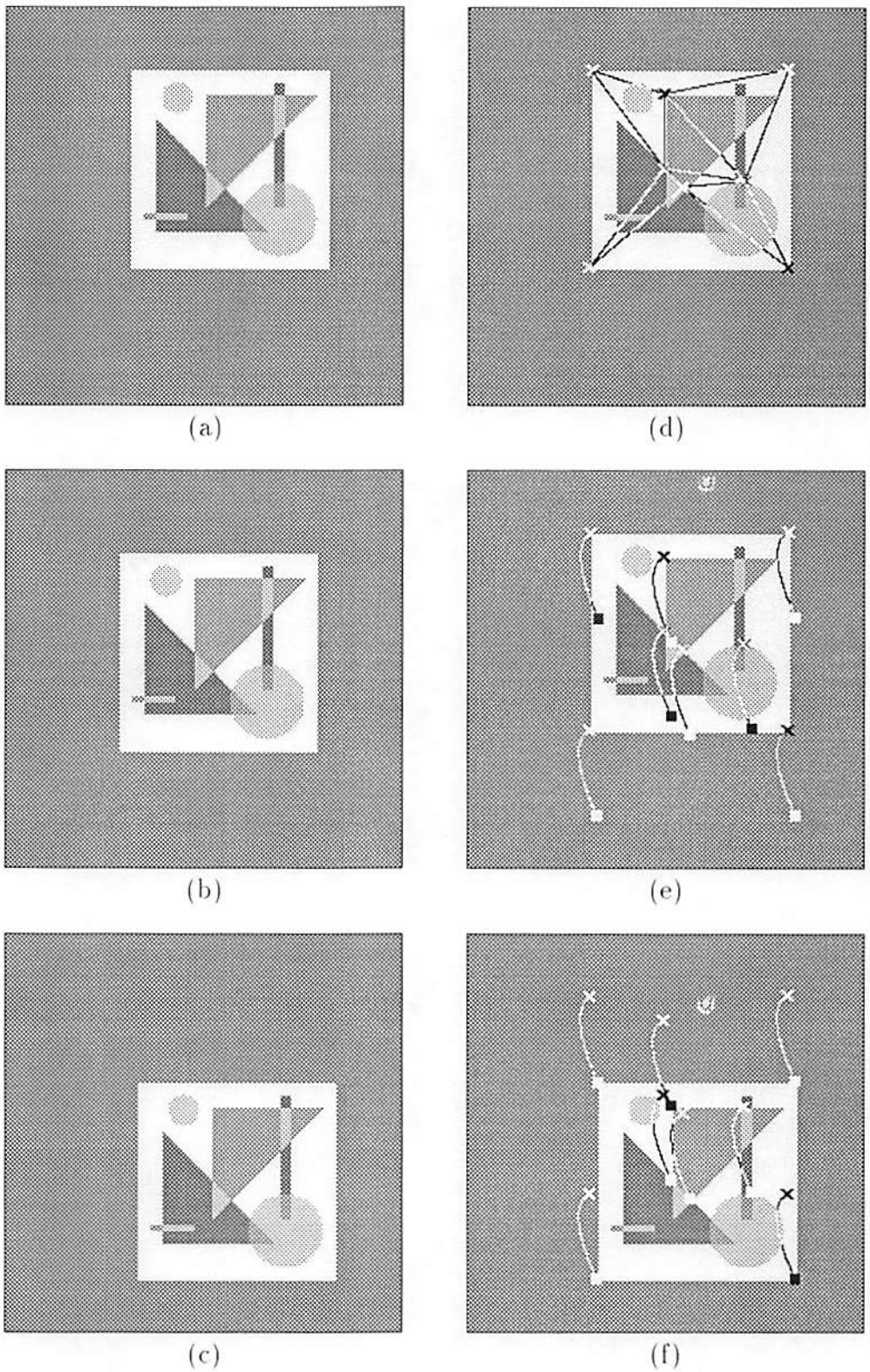

Figure 3.2: Matching results for the synthetic image sequence.

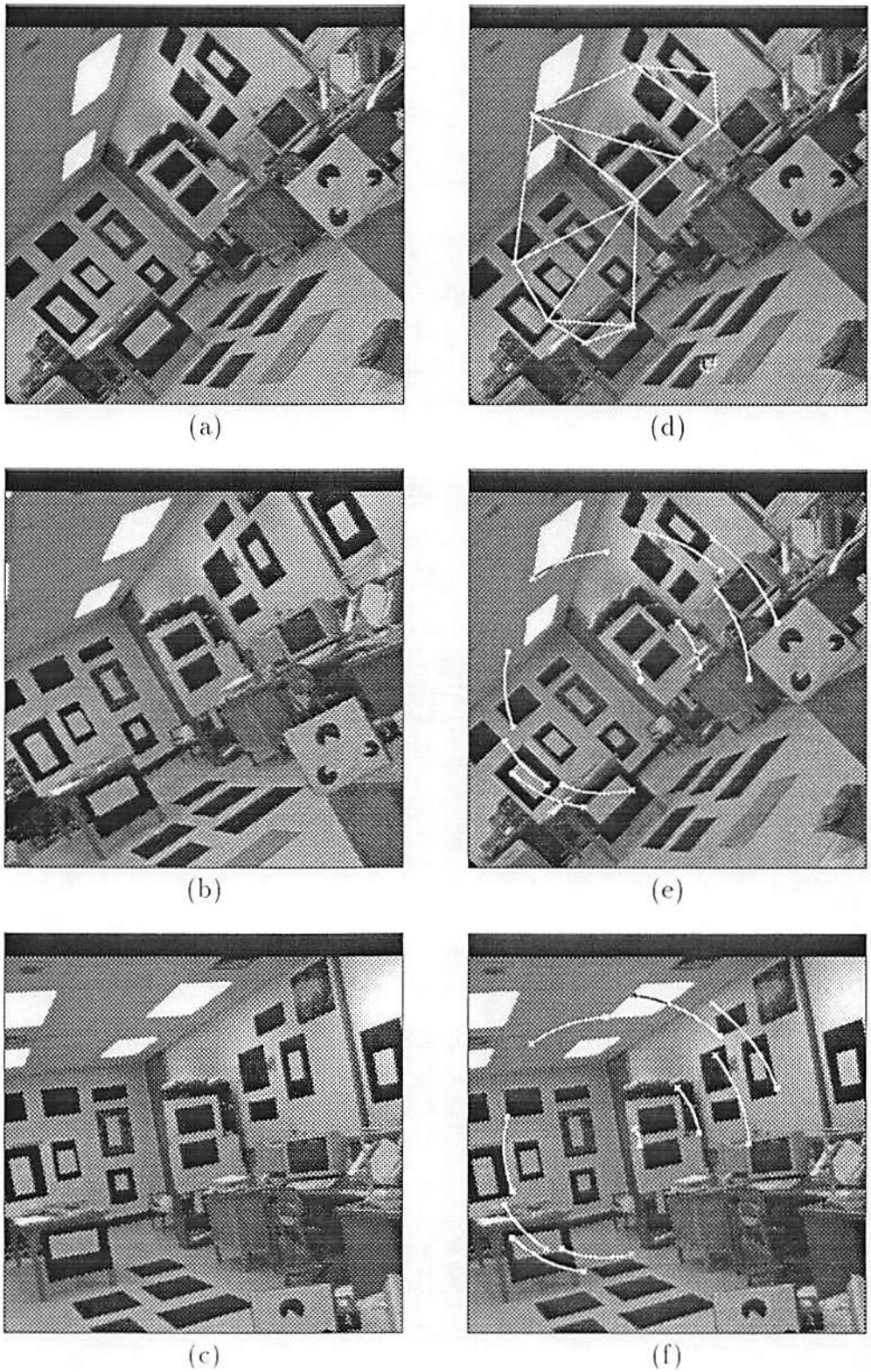

Figure 3.3: Matching results for the robot-arm sequence.

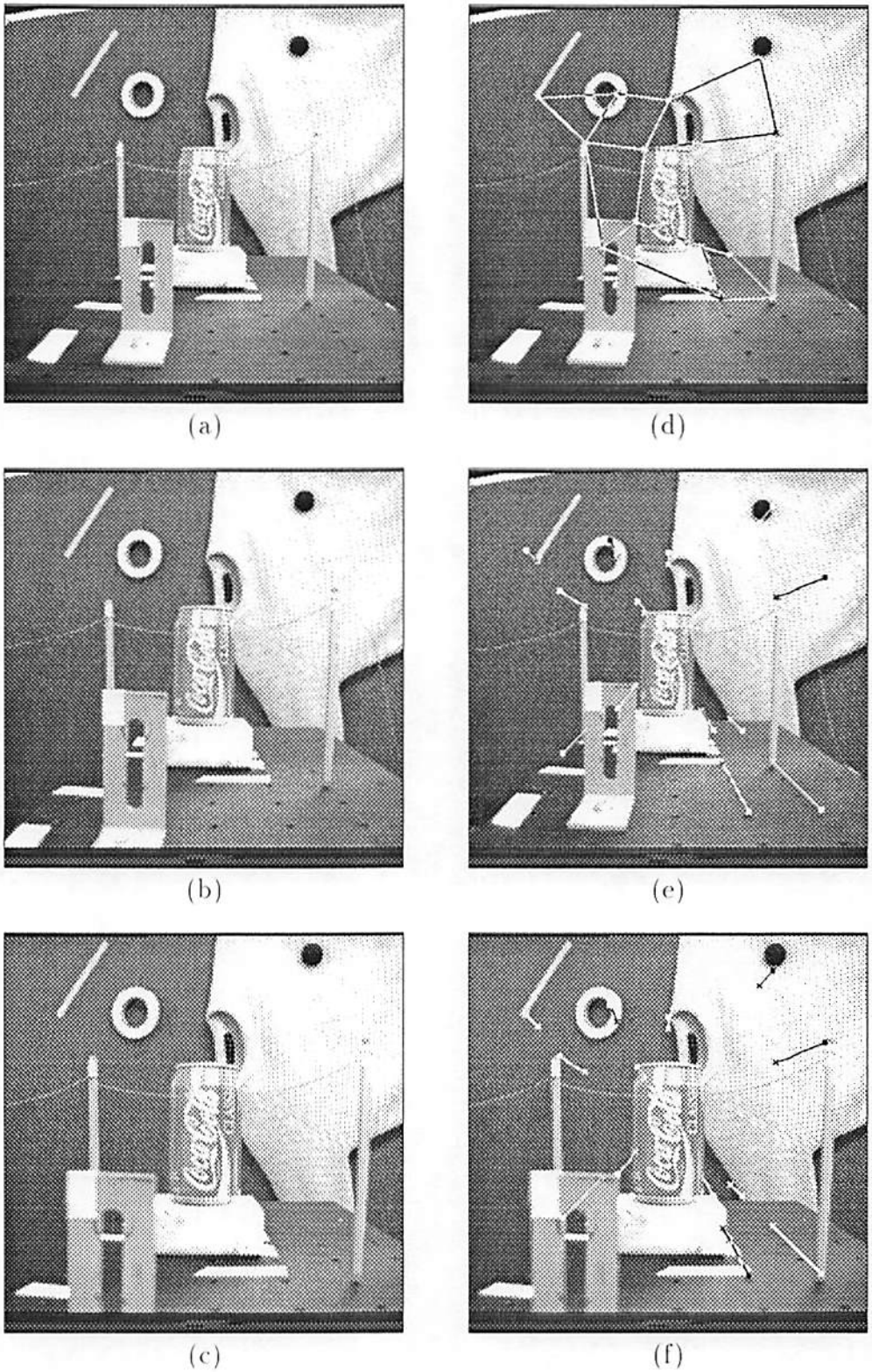

Figure 3.4: Matching results for the coke-can sequence.

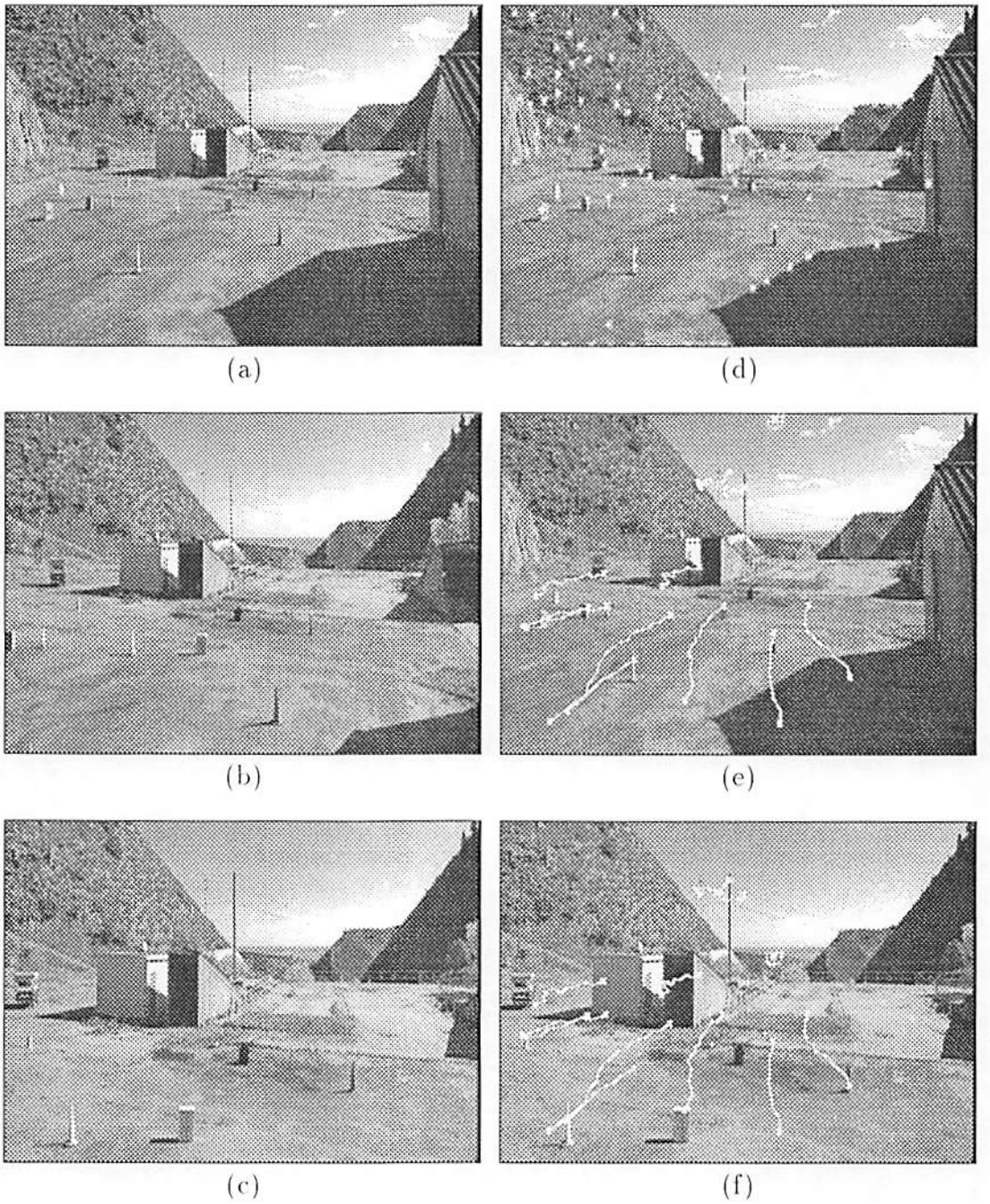

Figure 3.5: Matching results for the Rocket sequence.

Chapter 4

Passive Navigation

In this chapter, we develop the long-frame approach for the *passive navigation* problem, in which the objective is to aid the visual navigation of a vehicle (or mobile robot) in an environment containing stationary obstacles and possibly some navigational landmarks. The term "navigation" is used here in a very general sense; it also applies to such situations as the maneuvering of a robot arm amidst various objects on an assembly line. 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[1] are to be estimated based on a set of feature points identified and matched over the image sequence. The kinematic parameters of interest (motion parameters) are the position of the camera, its velocity, and higher-order translational parameters; its attitude (orientation), angular velocity, and higher-order rotational parameters. The structure parameters involved are the 3-D locations of unknown feature points in the scene.

One could take two different approaches to this problem, depending on the application, and on the kind of additional information available. The first approach represents all quantities of interest in the camera's coordinate system (CCS), and estimates them relative to the camera. The second approach estimates the "absolute" values of all parameters in a world coordinate system

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

	Approach I	Approach II
Primary mechanisms	Motion from Structure, reference navigation	Structure from Motion, dead reckoning
Representaion	World coordinates	Camera coordinates
Quantities of interest	Absolute pose, motion and structure	Structure and motion relative to camera
Additional information	Known landmarks, navigational beacons	Velocities measured using inertial navigational sensors
Motion model	Smoothness assumption is useful	Smoothness assumption is useful
Possible method of initialization	Pose estimation from the first few frames	Motion stereo over the first few frames
Application domains	Partially known or controlled environments (e.g. office, assembly line)	"Unknown" environments (e.g. highway)

Table 4.1: Comparison of different monocular approaches to visual navigation

(WCS) external to the vehicle, in which, as discussed in [40], the placing of a few easy to recognize beacons (or navigational landmarks) can considerably simplify the task of navigation. The two schemes are compared in in Table 1. In this chapter, we develop both the approaches. Our basic idea is 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 recursive 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 using an iterated extended Kalman filter (IEKF). The initial guess for the IEKF can be obtained in various ways, as described in Chapter 4.3. The matching of feature points is interleaved with the recursive state estimation. Current motion and structure estimates are used to predict the future positions of feature points, thereby reducing the search time for finding match points.

4.1 Approach I

The fundamental model of this section 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. 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 displacement and motion with respect to the WCS. The geometry of the problem is shown in Fig. 4.1.

4.1.1 Motion Model

Assuming that the motion of the origin of the CCS $\mathbf{p}_R(t)$ can be accurately modelled by a constant n^{th} derivative,

$$\mathbf{p}_R(t) = \mathbf{p}_R(t_0) + \sum_{k=1}^{n} \left. \frac{\partial^{(k)} \mathbf{p}_R(t)}{\partial t^{(k)}} \right|_{t=t_0} \frac{(t-t_0)^k}{k!} \tag{4.1}$$

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

The modelling of rotational motion is unavoidably nonlinear. The most common methods of representing rotation are by Euler angles, or by the axis of rotation $\mathbf{n} = (n_1, n_2, n_3)$, and the angle of rotation about the axis θ. Alternatively, we may write the rotation matrix $R(t)$ in terms of the unit quaternion $\mathbf{q} = (q_1, q_2, q_3, q_4)^T$ as follows:

$$R = \begin{pmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{pmatrix} \tag{4.2}$$

Since rotational displacement (also termed attitude or orientation) has only three degrees of freedom, the four components of the unit quaternion $\mathbf{q}$ are

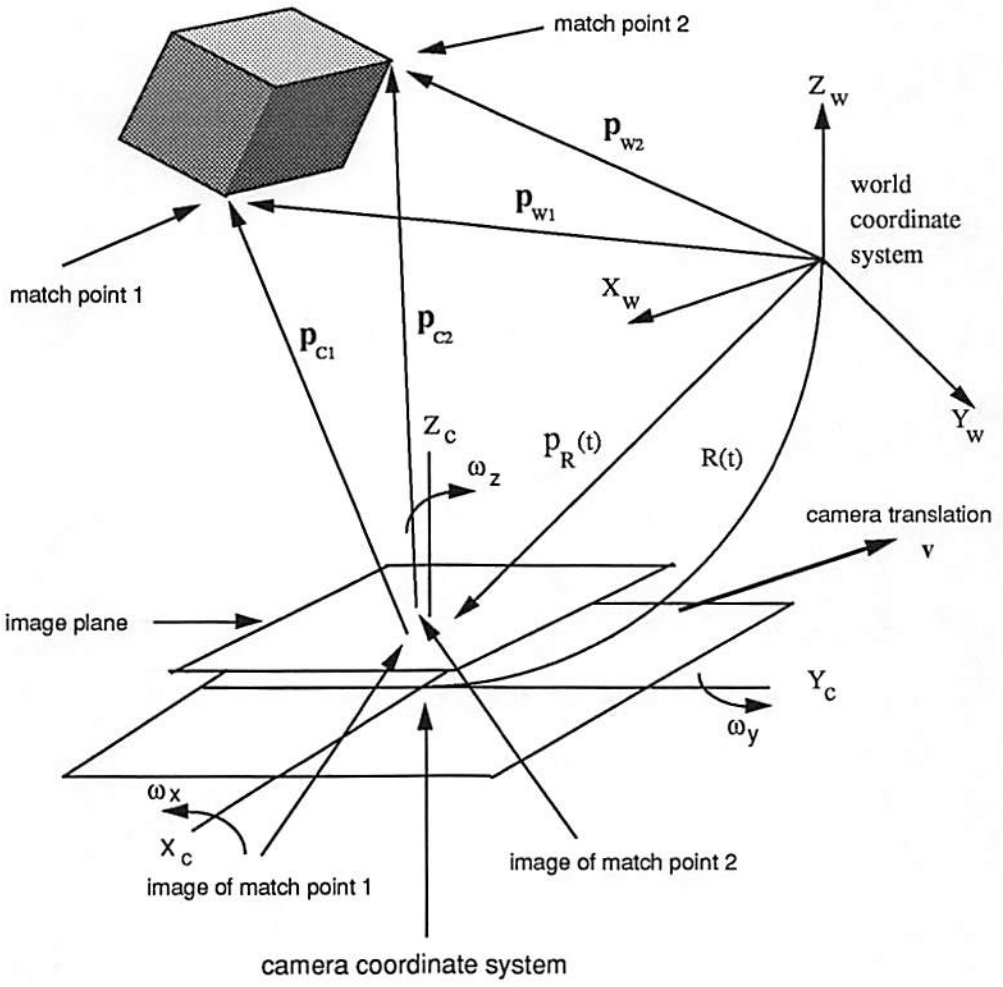

Figure 4.1: Models of motion and imaging used for passive navigation

constrained by the condition $|\mathbf{q}| = 1$. The unit quaternion is related to the $(\mathbf{n}, \theta)$ representation of the angular relation between coordinate systems by the relation

$$(q_1, q_2, q_3, q_4)^T = (n_1 \sin\theta/2,\ n_2 \sin\theta/2,\ n_3 \sin\theta/2,\ \cos\theta/2)^T \tag{4.3}$$

The quaternion $\mathbf{q}$ propagates in time according to the differential equation [31]

$$\dot{\mathbf{q}}(t) = \Omega(\omega_t)\,\mathbf{q}(t), \quad \mathbf{q}(t_0) = \mathbf{q}_0 \tag{4.4}$$

where $\Omega(\omega_t)$ is a matrix which is related to the instantaneous angular velocity $\omega_t = (\omega_x, \omega_y, \omega_z)$ as follows

$$\Omega(\omega_t) = \frac{1}{2}\begin{pmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{pmatrix}_t \tag{4.5}$$

Usually, it is reasonable to assume that ω is either zero (corresponding to constant camera orientation) or constant in time, and let the recursive filter handle minor deviations. If a higher order model is desired, one may model $\omega(t)$ using a truncated Taylor series of the form

$$\omega(t) = \omega(t_0) + \sum_{k=1}^{n} \left.\frac{\partial^{(k)}\omega(t)}{\partial t^{(k)}}\right|_{t=t_0} \frac{(t-t_0)^k}{k!} \tag{4.6}$$

The number of rotational parameters to be estimated (in addition to the four quaternions) is $3n$, where n is the order of the rotational motion. Typically one would choose $n \leq 2$.

4.1.2 Observation Model

The measurement model for a single point $\mathbf{p} = (x, y, z)^T$ is

$$\rho = \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} f_x \cdot \frac{x}{z} + X_0 \\ f_y \cdot \frac{y}{z} + Y_0 \end{pmatrix} + \begin{pmatrix} n_X \\ n_Y \end{pmatrix} = h[\mathbf{p}] + \mathbf{n}, \tag{4.7}$$

where f_x and f_y are the horizontal and vertical focal lengths, (X_0, Y_0) is the centre of the image plane, and n_x and n_y are measurement noise terms.

Combining the previous 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. 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 3-D camera-centered coordinates of match point i at time t, given by

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

At time t_k the image plane measurements of the match points are, using the central projection model,

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

where $h[.]$ is the projection operator and $\mathbf{n}$ is the measurement noise. This can be written as

$$\rho_i(t_k) = (X_i, Y_i)_k^T = h\left[R'(t)\,(\mathbf{p}_i - \mathbf{p}_R(t))\right] + \mathbf{n}(t_k), \tag{4.8}$$

where $i = 1, 2, \ldots, M$ and $k = 1, 2, \ldots, N$ for M match points (M_k of which correspond to navigational landmarks, and the remaining M_u to unknown points) and N image frames in the sequence.

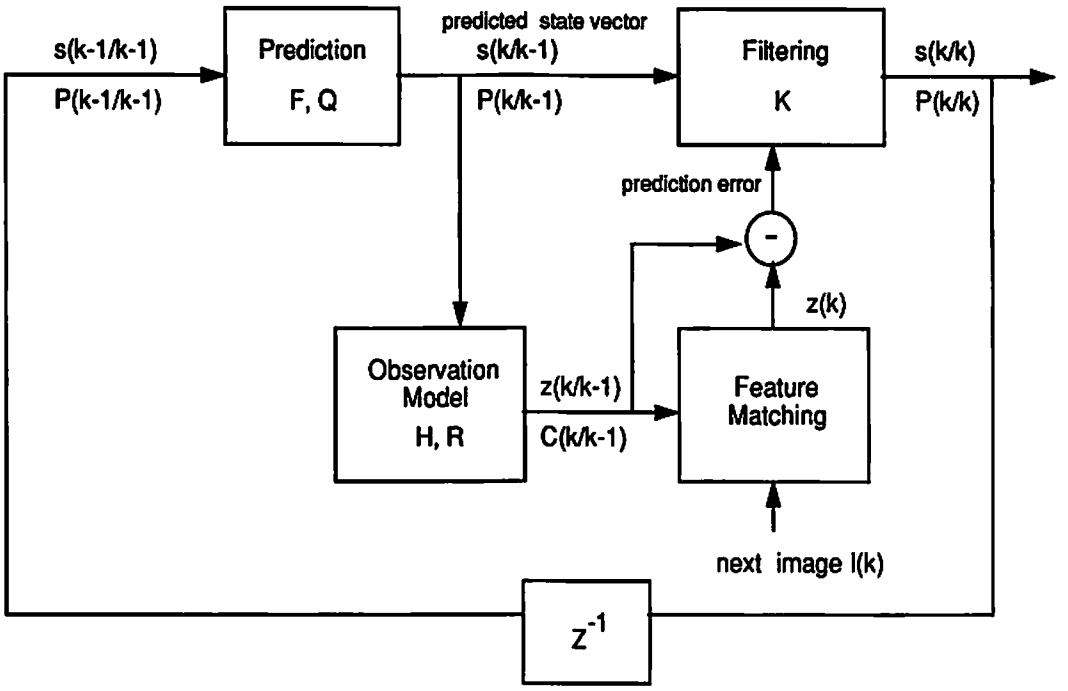

Figure 4.2: Schematic diagram of motion analysis

4.1.3 Recursive Formulation

The general problem is posed for solution by a recursive algorithm, by separating the statement of the problem into a plant model a measurement model. The parameters of interest are placed in a state vector, whose time-evolution (based on the motion model) is given by the plant equation, and whose relationship to the observed data is given by the measurement equation. This formulation is appropriate for solution using an IEKF.

The motion model presented in the previous section is fairly general, and depending on the application, one can choose specific cases of it. In all our experiments so far, we found it sufficient to use a simple motion model of

constant translational velocity v and constant camera orientation q. This does not mean that the resulting estimator is inapplicable to more general situations; on the contrary the methods give good results even in the presence of significant model deviations, as the experimental results will illustrate.

4.1.4 State Space Representation

The following d -dimensional vector of parameters is selected:

$$\mathbf{s}(t) = \begin{pmatrix} \mathbf{p}_R(t) \\ \mathbf{v} \\ \mathbf{q}(t) \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_{M_u} \end{pmatrix}_{d\times 1} \tag{4.9}$$

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

$$\dot{\mathbf{s}}(t) = \begin{pmatrix} \mathbf{v} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} \tag{4.10}$$

Using (4.9) and (4.10) the discrete version of the plant equation can be written as follows:

$$\mathbf{s}(k+1) = F\,\mathbf{s}(k) + \mathbf{w}_k \tag{4.11}$$

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

$$F = \begin{pmatrix} I_3 & I_3 & 0 & 0 & \cdots & 0 \\ 0 & I_3 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & I_3 \end{pmatrix} \tag{4.12}$$

and $\mathbf{w}_k$ is a discretized plant noise term used to compensate for errors in modelling.

Using (4.7), the vector-valued measurement function ($2M \times 1$) is

$$\mathbf{z}(t_k) = \begin{pmatrix} \rho_1(k) \\ \rho_2(k) \\ \vdots \\ \rho_M(k) \end{pmatrix} = \begin{pmatrix} h_1(k) \\ h_2(k) \\ \vdots \\ h_M(k) \end{pmatrix} + \begin{pmatrix} n_1(k) \\ n_2(k) \\ \vdots \\ n_M(k) \end{pmatrix} = \mathrm{h}[\mathrm{s}(k)] + \mathrm{n}_k \qquad (4.13)$$

The covariance R_k of the measurement noise 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.

4.1.5 IEKF Implementation

The relevant equations and notation for the IEKF can be found in [41] and in Appendix A. In order to implement one for our problem, it is necessary to determine the linearized measurement function $H(k)$, of dimensions $m \times d$, defined by

$$H(k) = \left. \frac{\partial \mathrm{h}[\mathrm{s}]}{\partial \mathrm{s}} \right|_{\mathrm{s} = \hat{\mathrm{s}}(k|k-1)} . \qquad (4.14)$$

In our case, with the state vector s as in (4.9), h[s] as in (4.13) we obtain

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

Where M_k and M_u are, respectively, the number of known and unknown features in the image sequence. Each $2 \times d$ submatrix H_i (corresponding to the

ith image point $\boldsymbol{\rho}_i$) is defined as follows:

$$H_i = \left.\frac{\partial \boldsymbol{\rho}_i(k)}{\partial \mathbf{s}}\right|_{\mathbf{s} = \hat{\mathbf{s}}(k|k-1)} . \tag{4.15}$$

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

where H_{pr}, H_q and H_p, are the partial derivatives of $\boldsymbol{\rho}_i$ 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

$$R = \begin{pmatrix} R'_x & R'_y & R'_z \end{pmatrix}$$

be the rotation matrix, written in terms of the rows of its transpose, and

$$R_{xq} \triangleq \frac{\partial R_x}{\partial \mathbf{q}} \; ; \; R_{yq} \triangleq \frac{\partial R_y}{\partial \mathbf{q}} \; ; \; R_{zq} \triangleq \frac{\partial R_z}{\partial \mathbf{q}}$$

Using (4.2), we can write

$$R_{xq} = 2\begin{pmatrix} q_1 & q_2 & q_3 \\ -q_2 & q_1 & q_4 \\ -q_3 & -q_4 & q_1 \\ q_4 & -q_3 & q_2 \end{pmatrix} \; ; \; R_{yq} = 2\begin{pmatrix} q_2 & -q_1 & -q_4 \\ q_1 & q_2 & q_3 \\ q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \end{pmatrix} \; ; \; R_{zq} = 2\begin{pmatrix} q_3 & q_4 & -q_1 \\ -q_4 & q_3 & -q_2 \\ q_1 & q_2 & q_3 \\ -q_2 & q_1 & q_4 \end{pmatrix}$$

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:

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

It is necessary to normalize the quaternion estimate immediately after the measurement update. It has been shown that this does not adversely affect the performance of the estimator [6, 10]. Numerical aspects of filter tuning, convergence, and stability are discussed in Chapter 6.

4.1.6 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 two real image examples are presented in this section.

Experiments with Synthetic Data

An aerial view of the environment for the Rocket sequence is shown in Fig. 4.3. Ten feature points (indicated by asterisks), and the approximate positions of the camera (indicated by dots), are shown. There is one more feature point, which is not shown because it is at a very large distance from the vehicle. The scales of the axes are in metres. (The environment is actually taken from the set-up used for obtaining the Rocket sequence, but different kinematic parameters are used here in simulating the camera motion.) Four points out of the 11 are treated as landmarks of known 3-D location, and the structure of the remaining seven points is estimated, along with the kinematic parameters of the camera.

The simulated measurements are generated using the following steps:

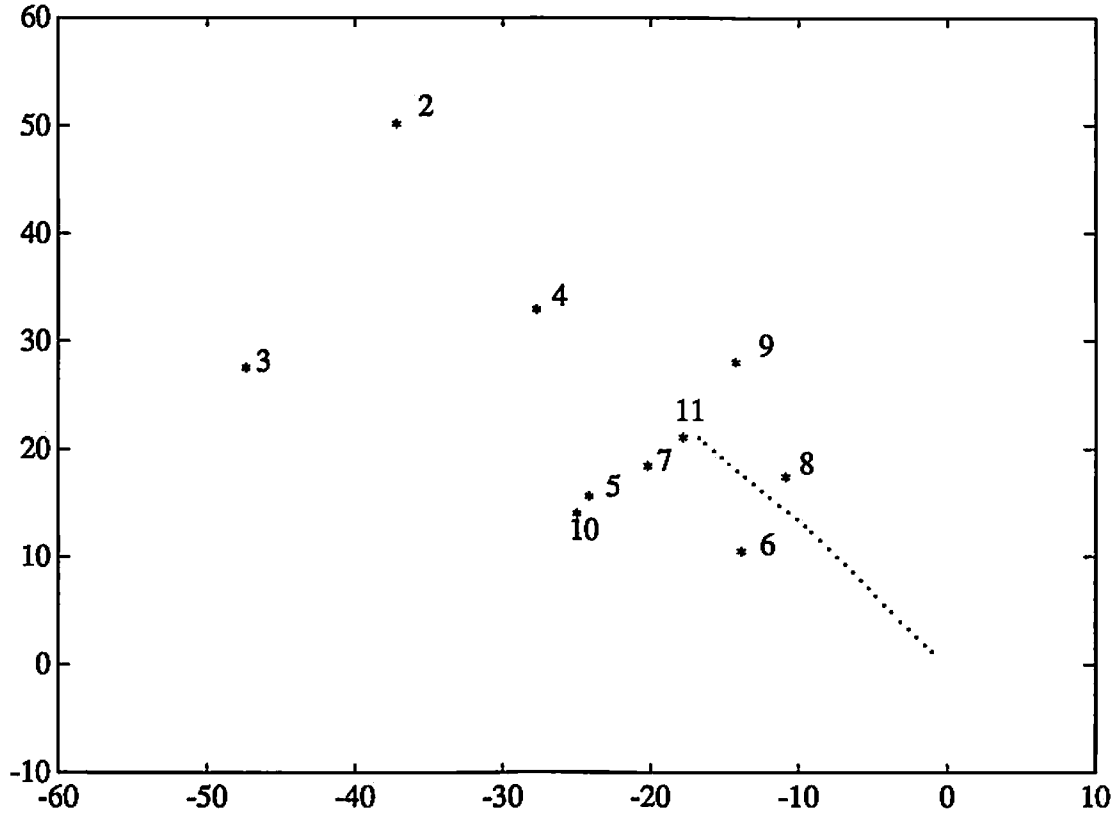

Figure 4.3: Aerial view of the environment for the Rocket sequence

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.

2. The image locations of the feature points are computed using the imaging model and camera calibration parameters.

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 measurement noise can be directly calculated from the image resolution used.

The image plane trajectories obtained by the above procedure are shown in Fig. 4.4. In the first example, the model assumptions of constant orientation and velocity are followed exactly. The image plane trajectories of the feature points are straight lines (except for the errors due to spatial quantization). In the second example, the camera is permitted to accelerate translationally, and to rotate with a small, linearly increasing angular velocity, violating the constant orientation and constant translational velocity assumptions.

The results of the forward pass of the initialization step (Section 4.3) are shown in Figures 4.20 and 4.21. The final estimation results are shown in Figures 4.5 and 4.6 as a function of the frame number. The solid lines represent the true state values, and the dashed lines their estimates. The dimensions of position and structure estimates are metres, and those of the velocities are metres/second. Three of the seven structure estimates are shown. The results for the first example are obviously better, but the results for the second example, notwithstanding its model deviations, are only slightly worse. We have obtained similar results for other values of motion and structure parameters, and with greater noise levels. These experiments demonstrate the ability of a recursive filter based on very simple models to adapt to and degrade gracefully under significant model deviations.

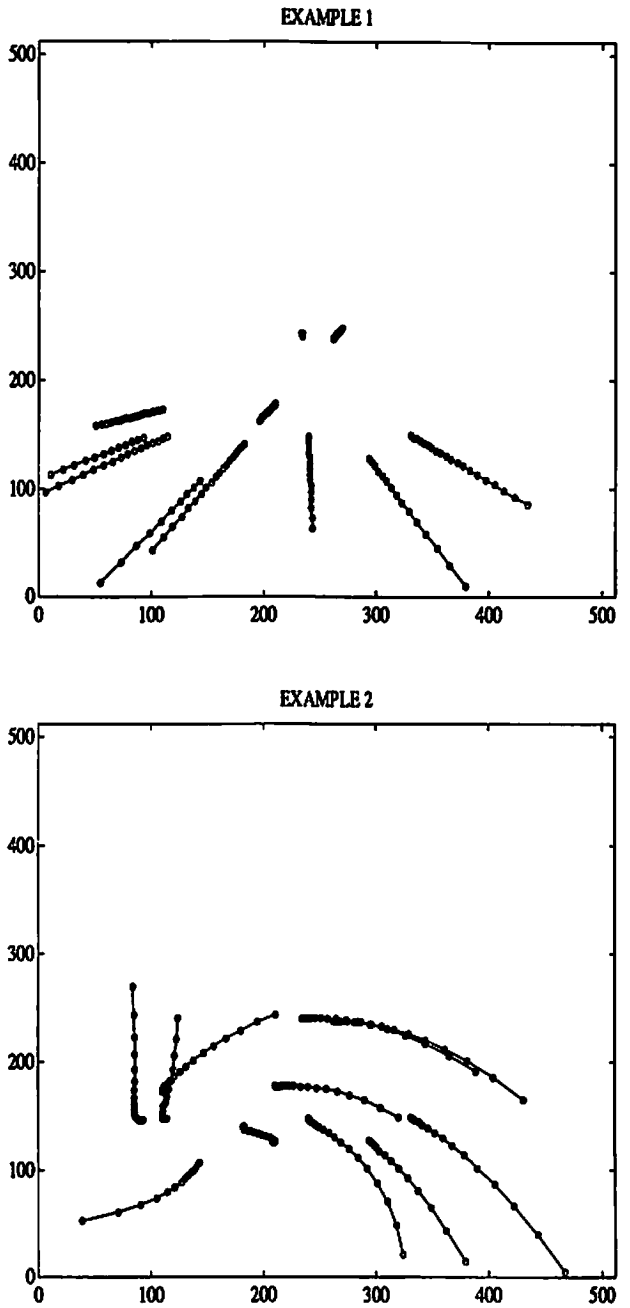

Figure 4.4: Image plane trajectories of feature points for simulated camera motion.

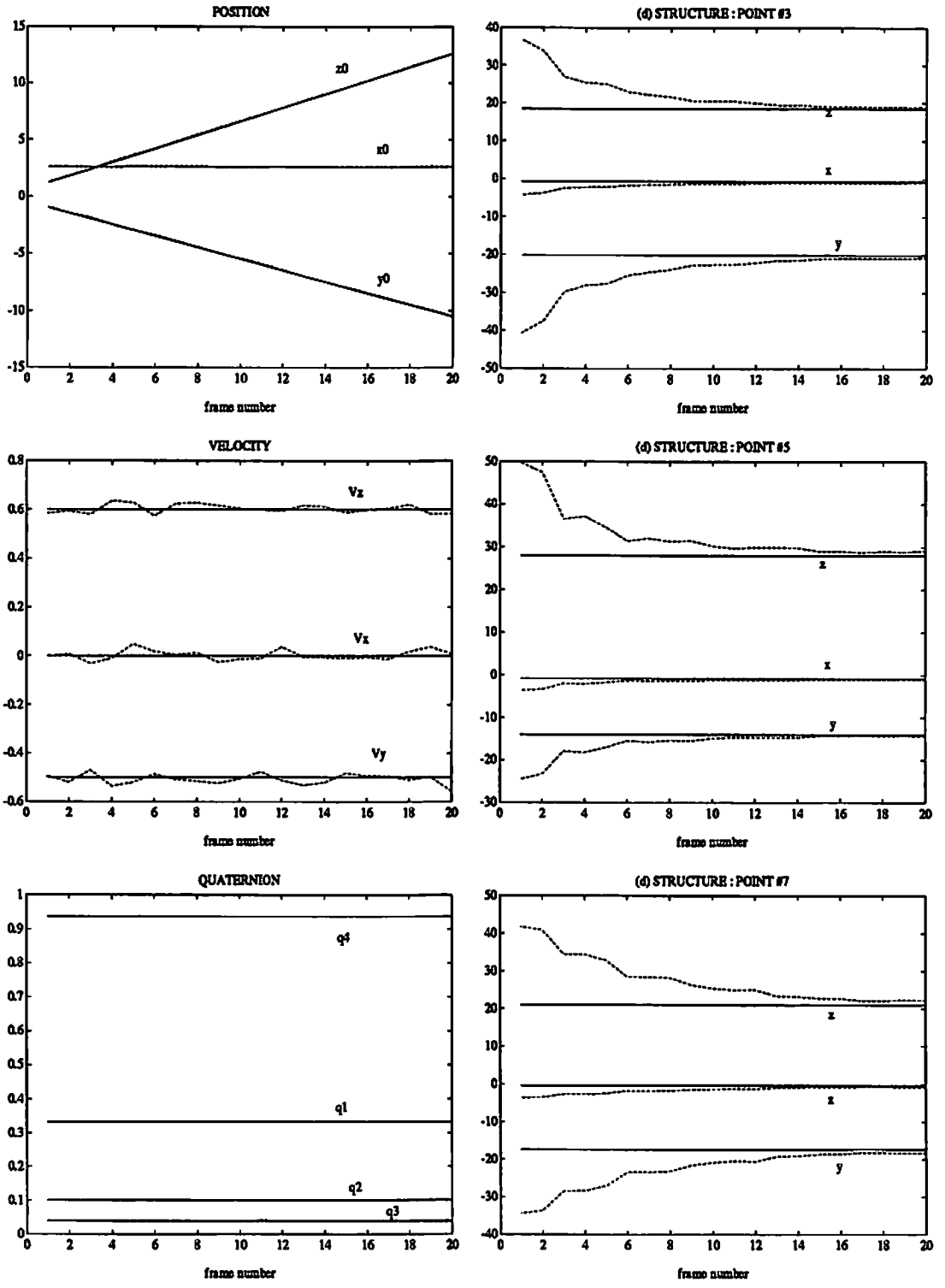

Figure 4.5: IEKF results for synthetic data: Example 1

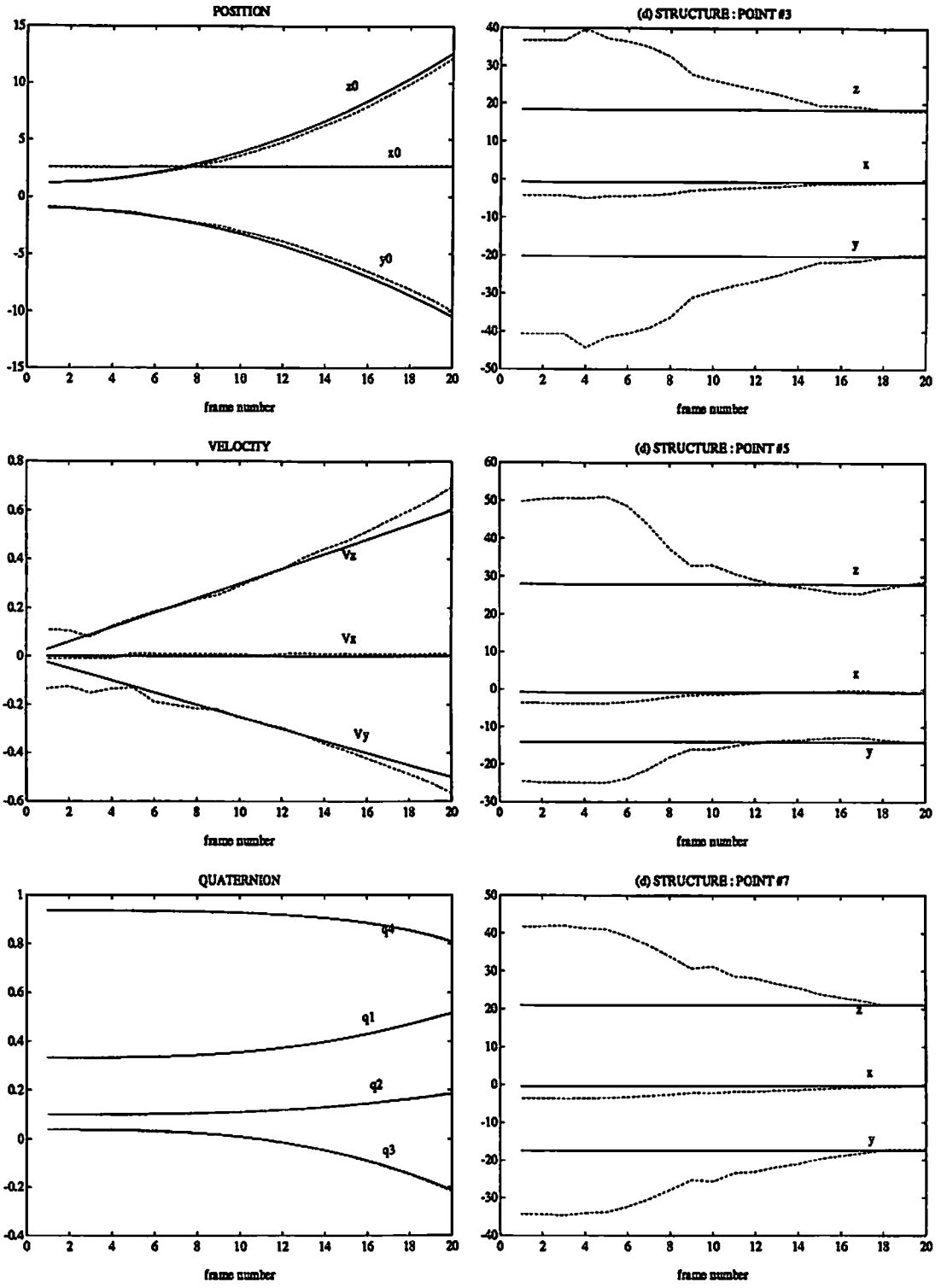

Figure 4.6: IEKF results for synthetic data: Example 2

Experiments with Real Data

The real image sequences used are the Rocket ALV sequence [25] and the "Robot" sequence [44], both created at the University of Massachusetts.

Results with the Rocket sequence:

Some images from the Rocket sequence (consisting of 16 images[2]) are shown in Fig. 4.7 and Fig. 4.8. The results of feature extraction are shown in Fig. 4.9. We use only the features which (approximately) correspond to those for which 3-D ground-truth data are available , since it is not possible to validate results obtained for other features. An aerial view of the locations of some of the selected features are shown in Fig. 4.3, along with the different positions of the camera during its motion. The results of the forward pass of the initialization step are shown in Fig. 4.22, and the final estimation results in Fig. 4.11. The estimates of the position, velocity, orientation, and locations of three of the seven unknown points are shown. There are minor errors in the estimates of position and velocity, while the orientation estimates and the final structure estimates are reasonably accurate.

The motion of the vehicle is approximately an axial translation, and its orientation is approximately constant. However, as indicated by the image plane trajectories of feature points illustrated in Figures 4.10, the motion is not very smooth since 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 several pixels) between the actual image locations of feature points, and their locations predicted by the camera calibration and the ground truth. This could be due to a variety of reasons, such as camera calibration errors and the fact that the features chosen from the images do not correspond exactly to those for which ground truth was measured. Currently we treat this as additional measurement noise, but in future work we hope to be able to find the sources of error and correct them.

[2]The original sequence has 30 images, of which we used first 16.

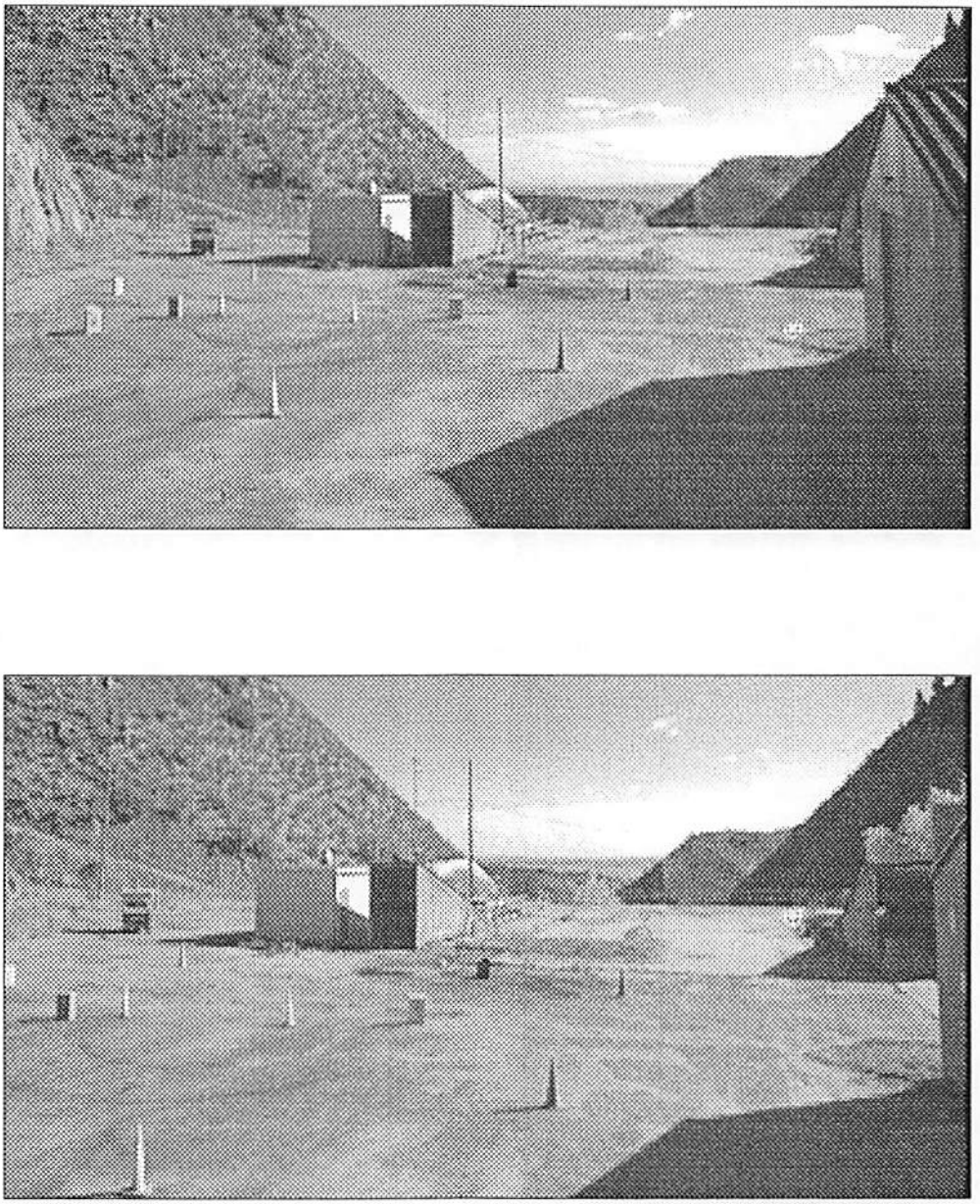

Figure 4.7: Frames 1 and 6 of the Rocket sequence.

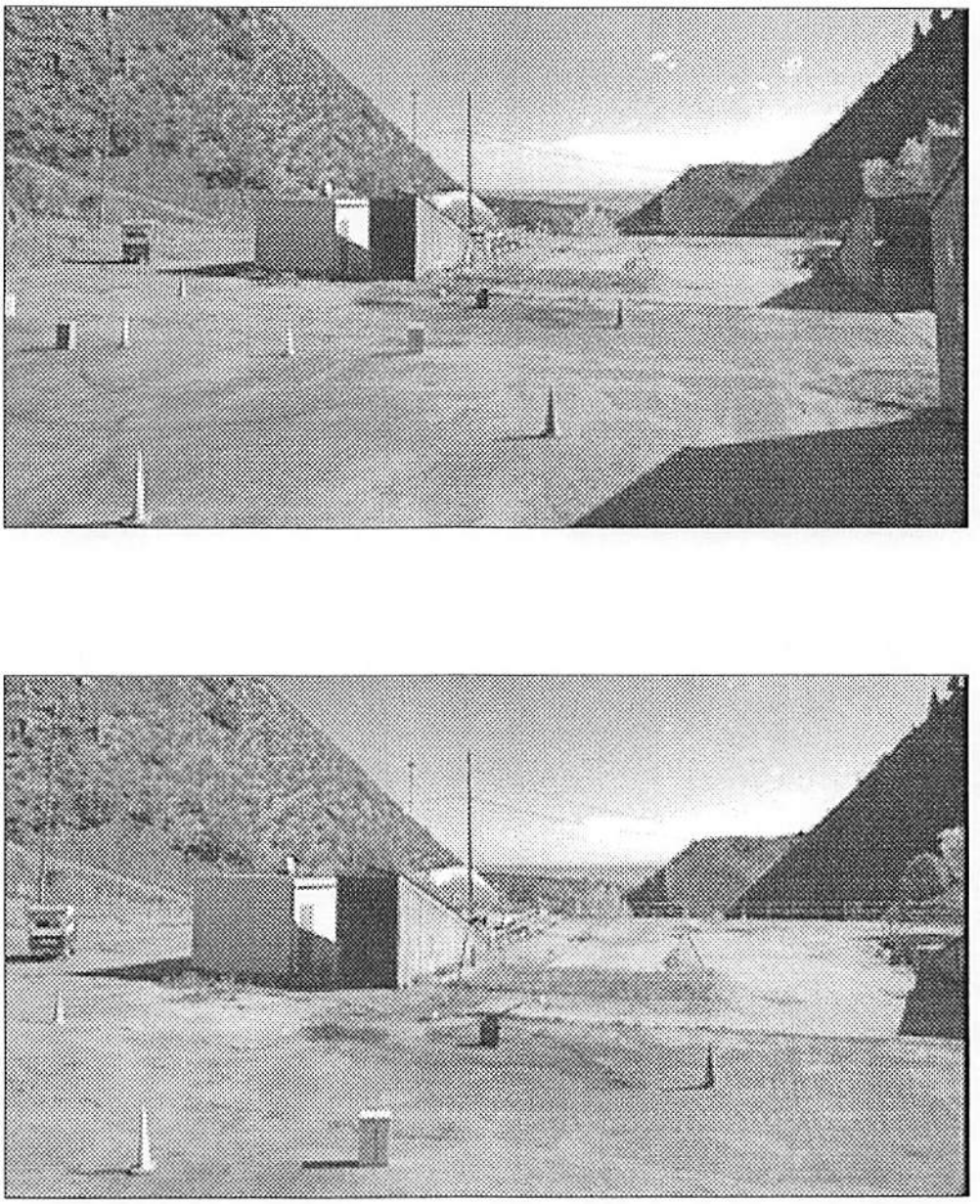

Figure 4.8: Frames 10 and 16 of the Rocket sequence.

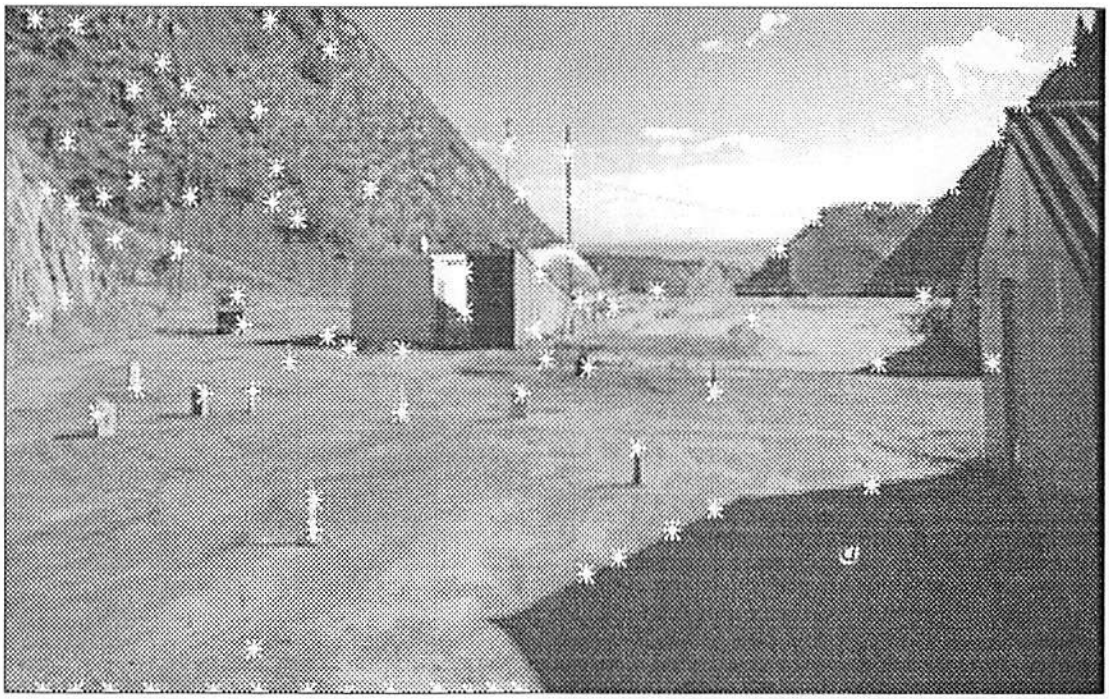

Figure 4.9: Feature points extracted from the first image of the Rocket sequence

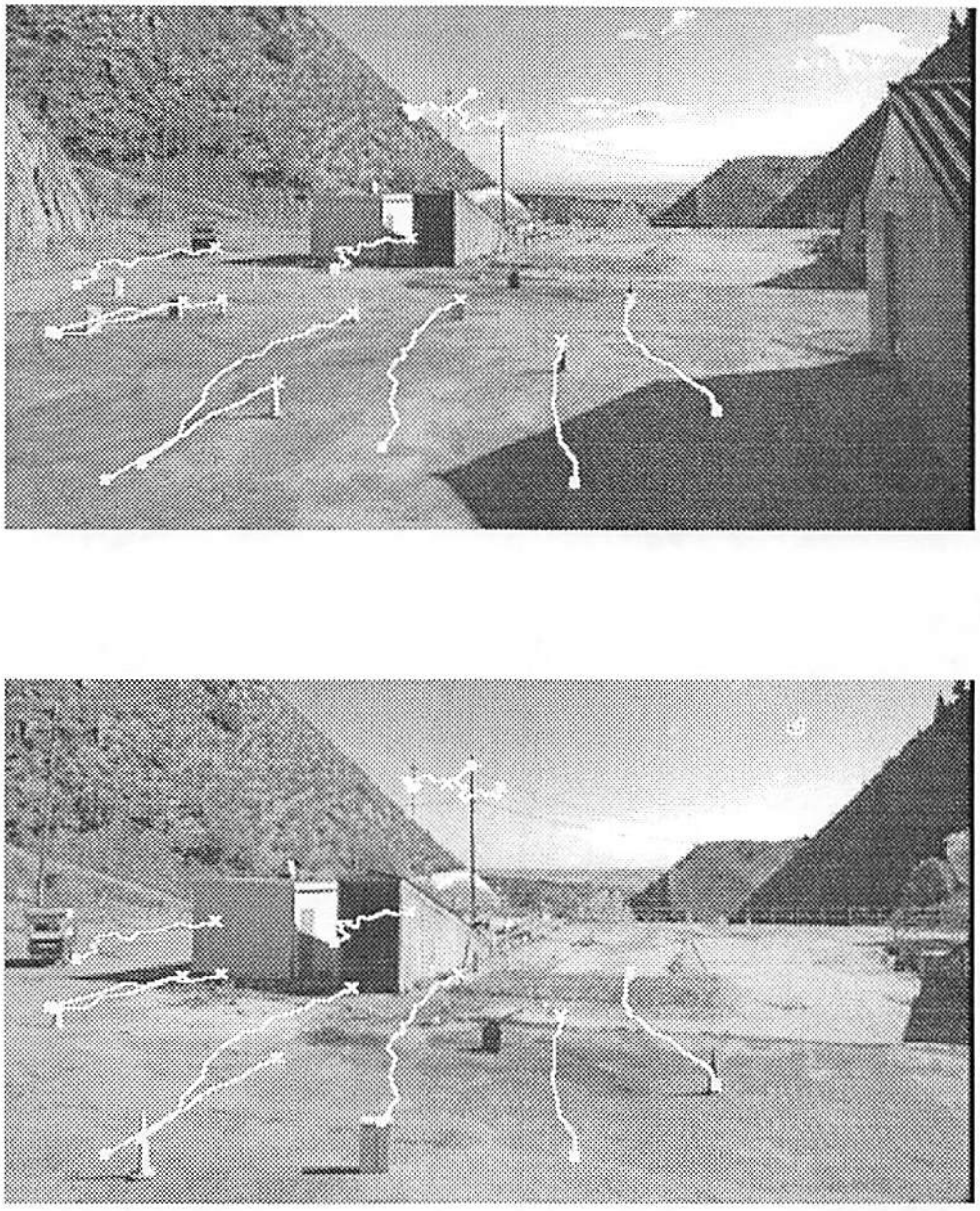

Figure 4.10: Trajectories of selected points, superimposed on the first and last image in the Rocket sequence.

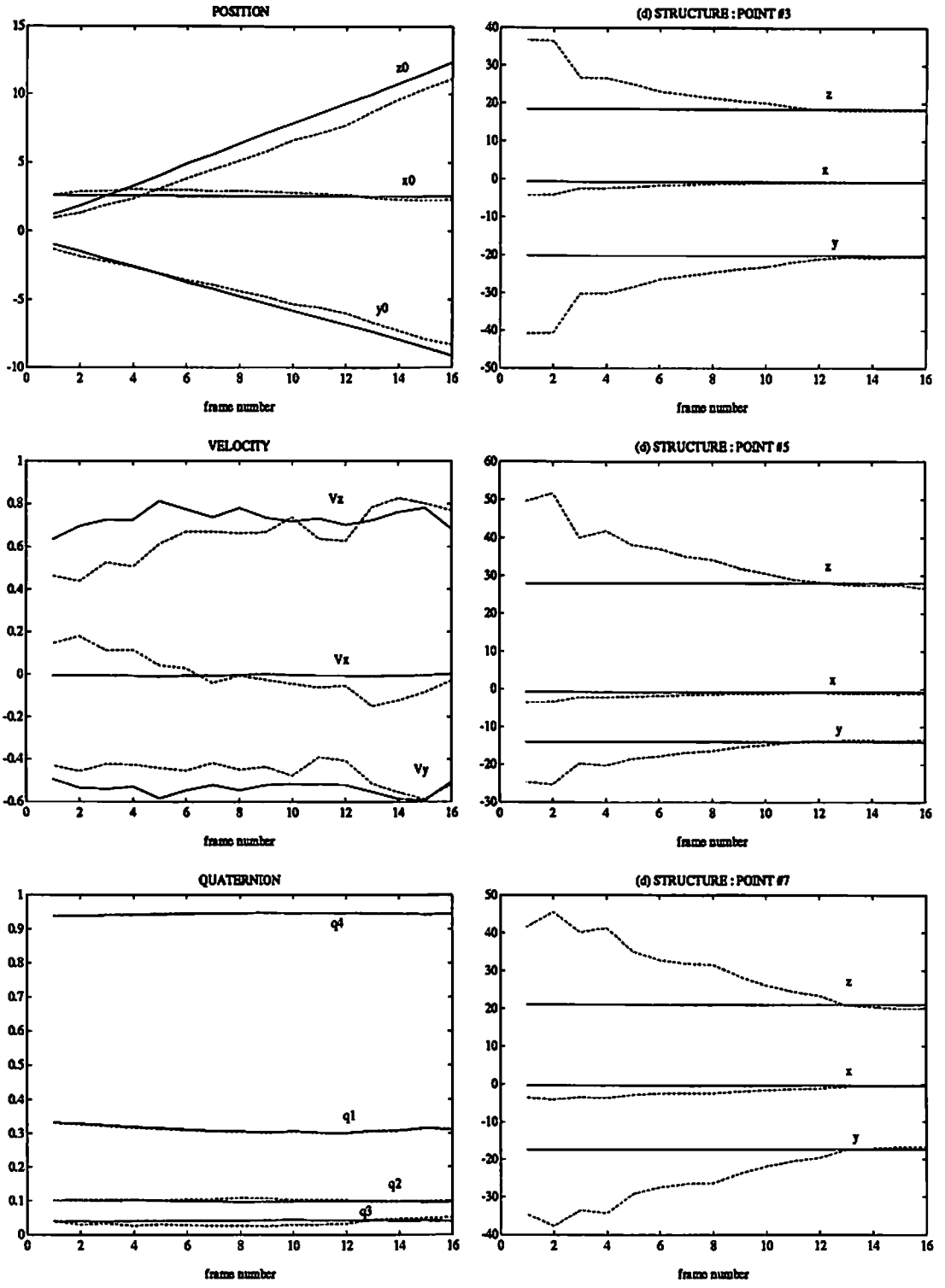

Figure 4.11: IEKF results for the Rocket sequence

Results with the Robot sequence:

The Robot sequence (Figs. 4.12–4.12), in contrast to the Rocket sequence, involves predominantly rotational motion. The images were obtained by a camera attached to a Puma2 robot arm, moving in a rough circle almost parallel to the image plane. The axial component of the motion is fairly small. The scene is the interior of an office room, containing many man-made objects and rectangular patterns on the walls and floor. The corners of these rectangles are logical choices for feature points, since they can be expected to have clearly distinguishable intensity profiles, and therefore should be easy to match. In the work done by Kumar and Hanson [44], there are 12 known points (landmarks) and 20 unknown points, distributed evenly throughout the room. For this example our feature extractors picked up very few of the points chosen by Kumar and Hanson (for which they provide ground-truth 3-D measurements), so we located the 32 points in the first image by inspection. The image trajectories of these feature points are shown in Figs. 4.14–4.15.

The Rocket image sequence, in a manner of speaking, complements the Rocket sequence, because unlike the latter, it is taken in an indoor environment, and has a substantial amount of camera rotation. One would expect the results to be poorer, because the motion is even more in violation of the assumed model than in the previous case. However, this is not really the case, as the IEKF estimates in Fig. 4.16 indicate. The estimation results are in general as good as or better than the ones for the Rocket sequence, except for the velocity estimates. The poor quality of the velocity estimates is due to the fact that the rotation of the camera, which dominates the motion, is not directly represented in the model, leading to the observation of a kind of "pseudo-translation" between successive frames. However, this does not seem to affect the other state estimates.

It is apparent in both these cases (Rocket and Robot) that the motion of the camera does not obey the model developed in Section 3, since neither its translational velocity nor its attitude is constant. However, the IEKF seems to be capable of handling these model deviations, as the state estimates indicate. This is essentially due to the ability of the IEKF to "forget" the past, and

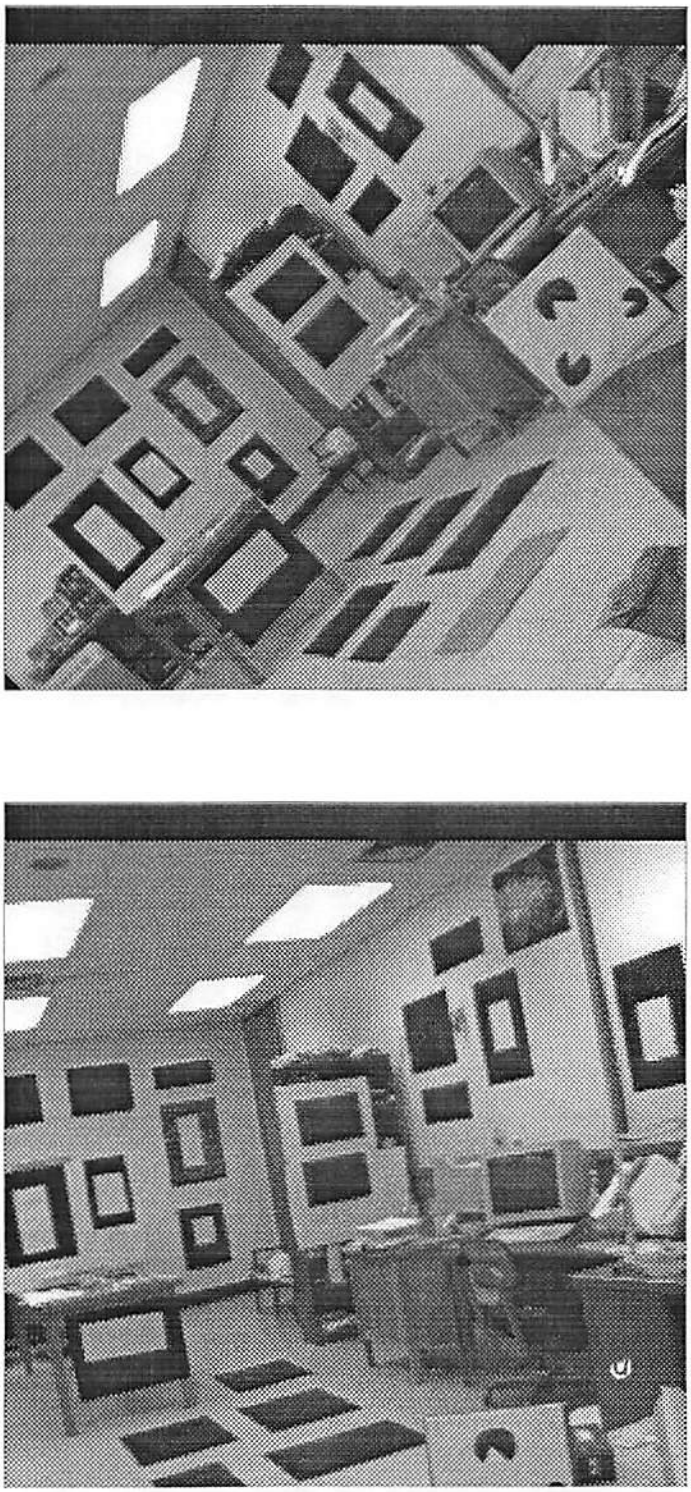

Figure 4.12: Images 1 and 10 from the Robot sequence.

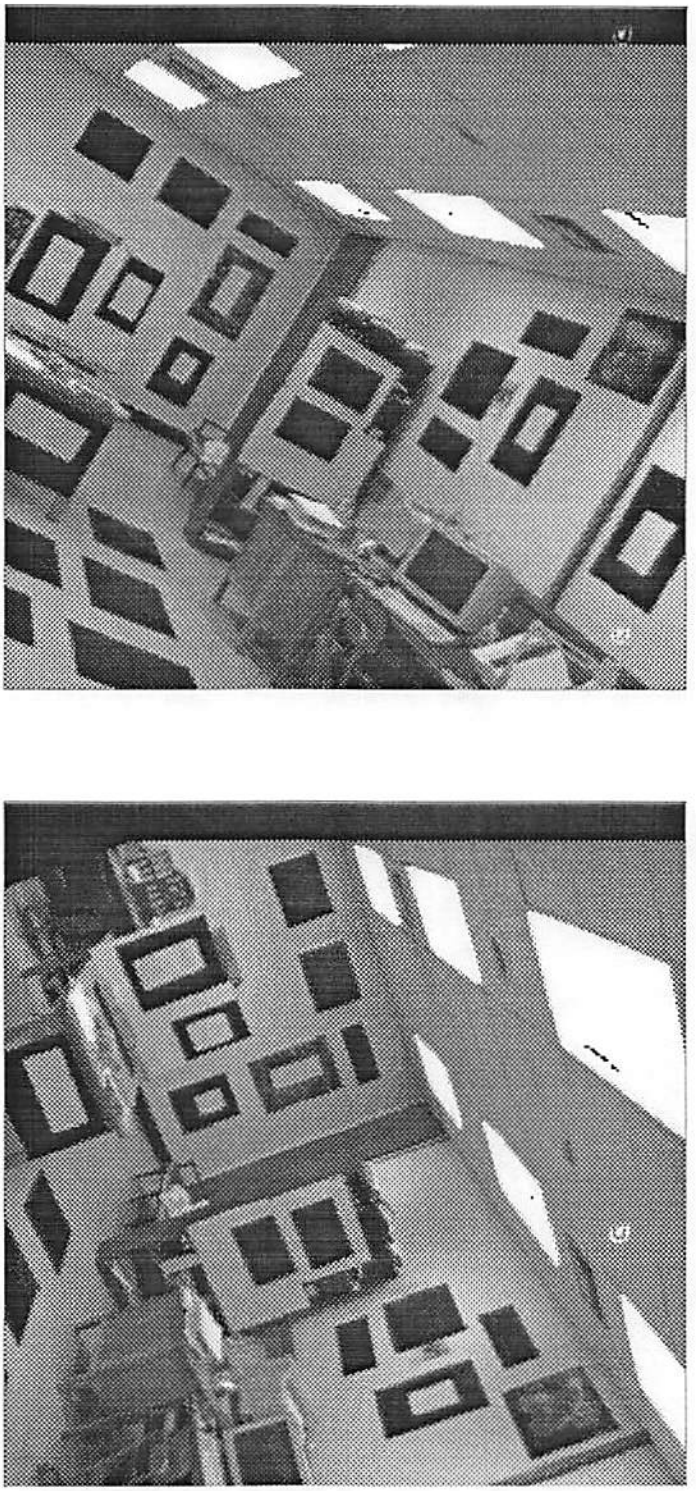

Figure 4.13: Images 20 and 30 from the Robot sequence.

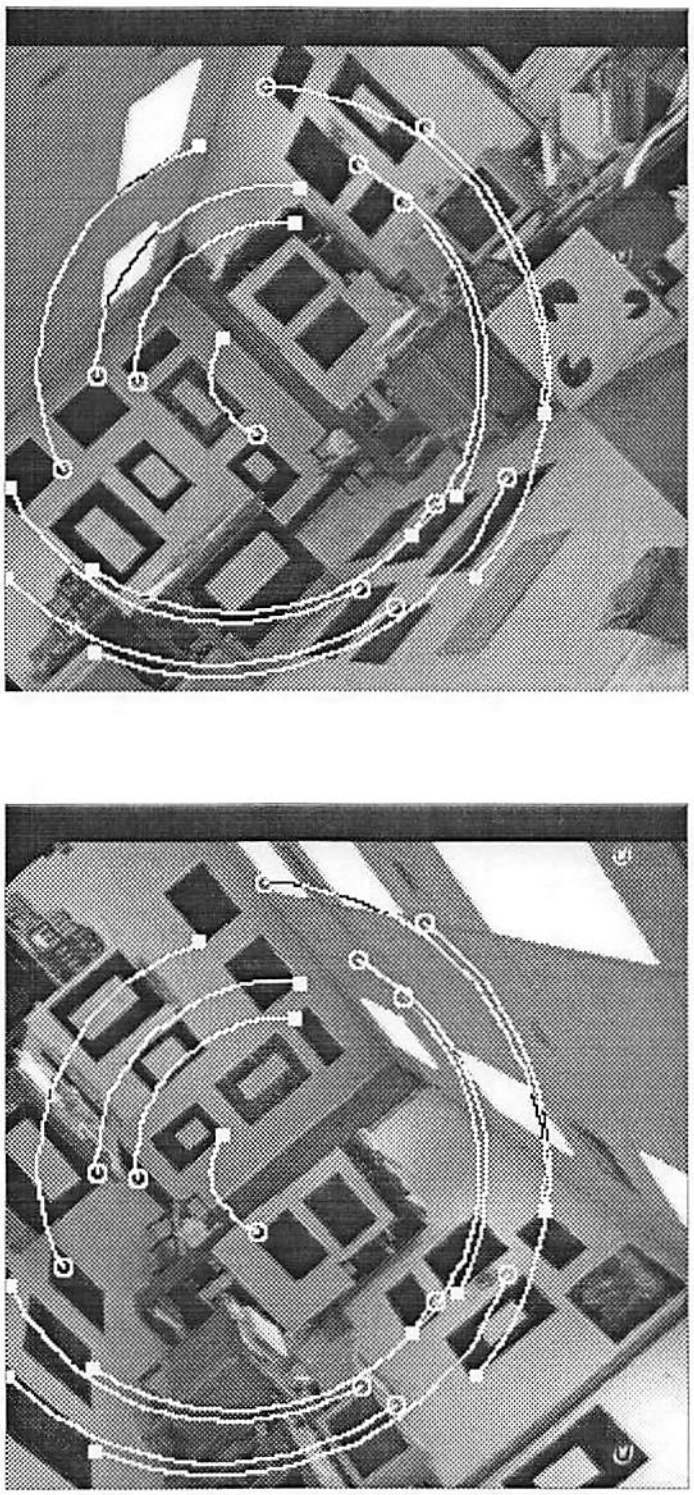

Figure 4.14: Trajectories of the known points of the Robot sequence superimposed on images 1 and 25.

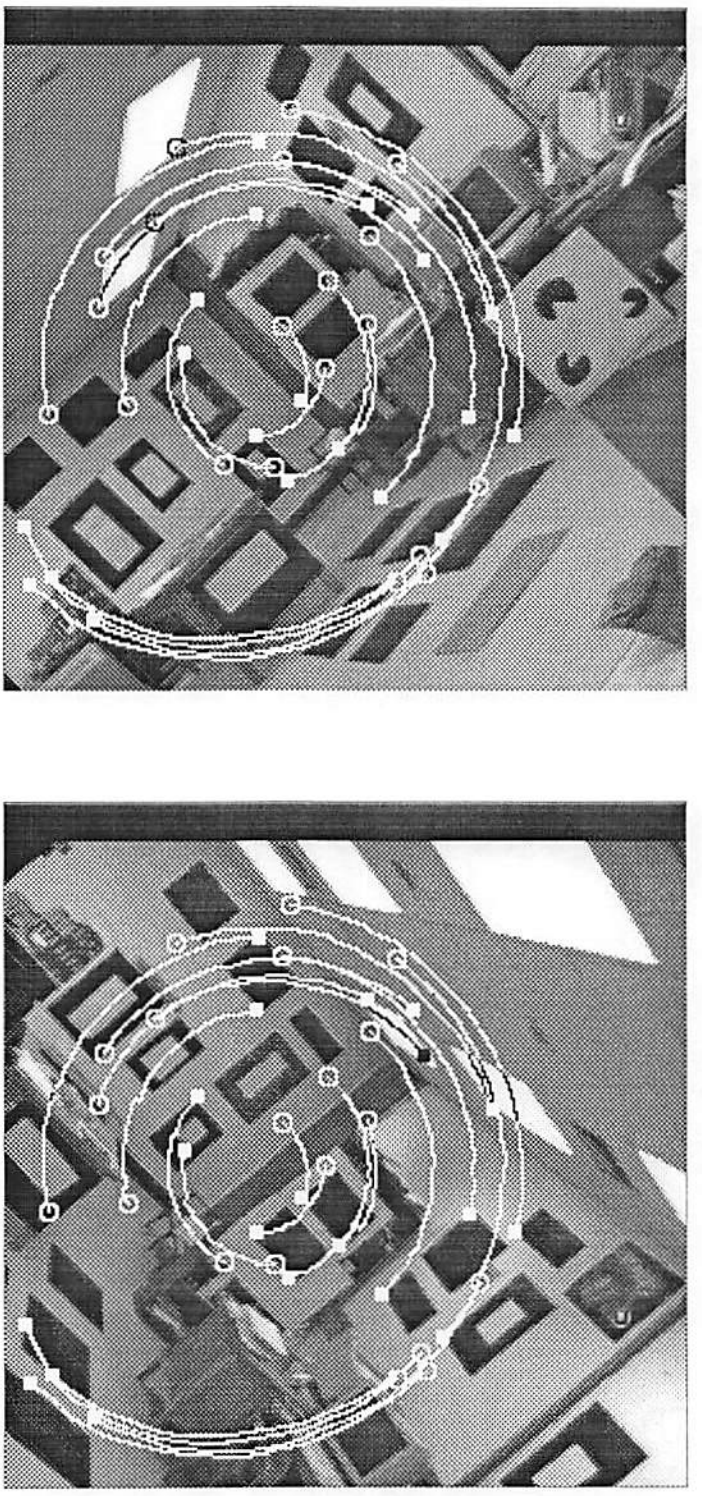

Figure 4.15: Trajectories of the unknown points of the Robot sequence superimposed on images 1 and 25.

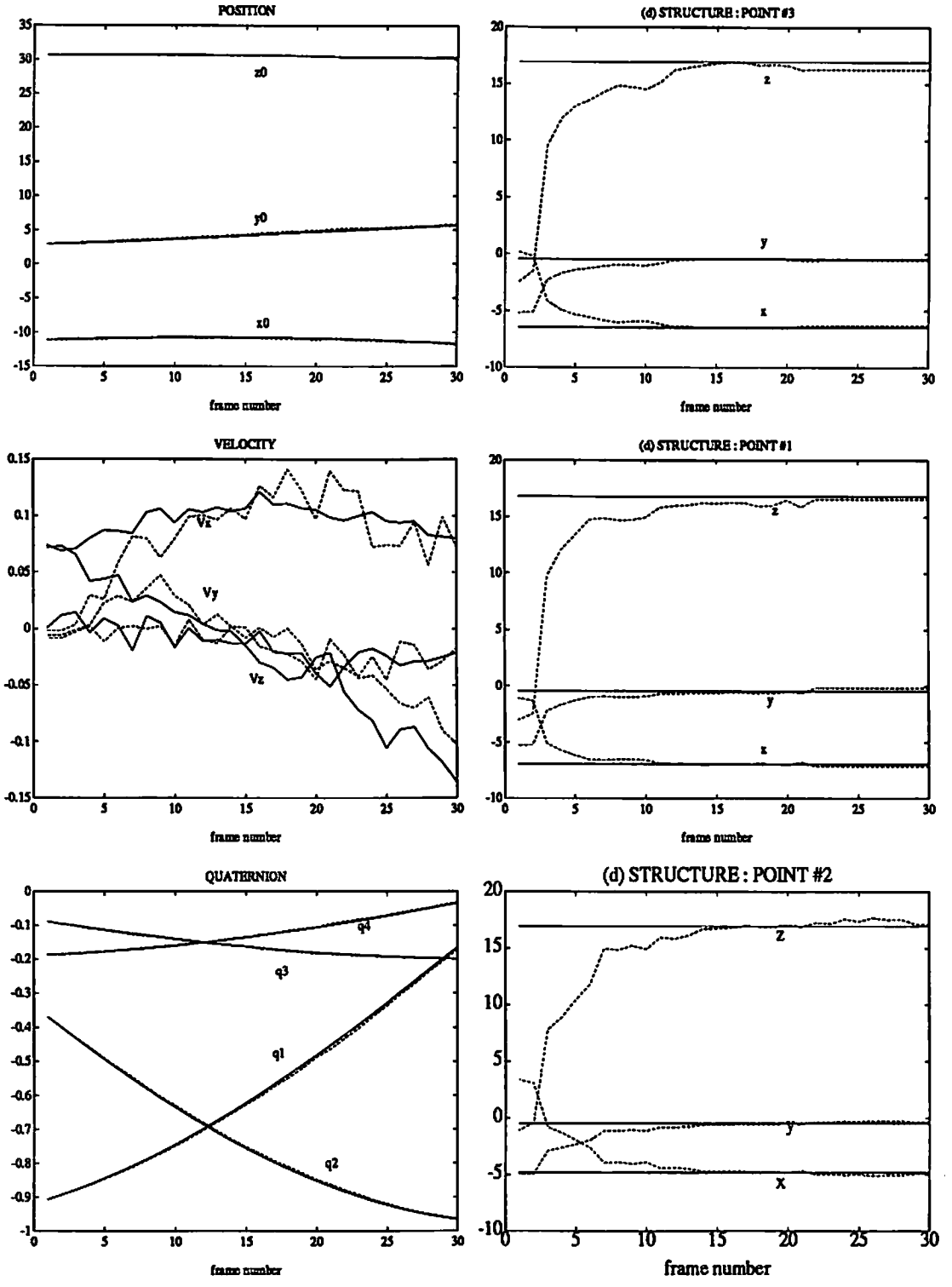

Figure 4.16: IEKF results for the Robot sequence

respond to the present; an ability which can be controlled by varying the assumed plant noise covariance. Some discussion of how this can be done is can be found in [10]. Thus the IEKF can be "tuned" to specific applications, depending on the extent of the model deviations expected.

The performance of the method depends to some extent on the number and location of the navigational landmarks. We have experimented with different configurations of the features chosen as the known landmarks. As one would expect, results improve as the number of landmarks increases, and they are more widely dispersed. In general it is advantageous to have landmarks that are fairly distant from the camera, and approximately in its path. These are less likely to disappear quickly from the field of view of the camera. This is also consistent with the idea of goal-oriented navigation, the goal being one of the distant landmarks. In the Rocket experiment reported, the four most distant feature points were treated as landmarks, and in the Robot experiment, the 12 points selected by Kumar and Hanson in [44] were used. We did not attempt the high level processing needed to identify the features as landmarks, as would be required in a real application. This is application dependent, and should be done based on knowledge of the appearance of the landmarks.

4.2 Approach II

In this section, we will address the passive navigation problem from a camera-centred standpoint—wherein the parameters of interest are represented in the coordinate system of the camera. We assume that additional information in the form of translational and rotational velocity measurements is available and we remove the previous assumption of partially known structure. This formulation is more similar than the world-centred approach to other work on passive navigation, such as [62, 58].

As mentioned earlier, measurements the dynamics of the camera are assumed to be available. These mostly take the form of (instantaneous) linear and angular velocity measurements at the sampling instants, obtained using inertial sensors. Higher order motion parameters may be derived from these;

in this research we do not attempt to do this, since the first order measurements are usually sufficient by themselves for estimation of structure. Thus we assume that the derivatives of the velocities are negligible.

4.2.1 Motion Model

The camera's dynamics consist of its instantaneous translational velocity $\mathbf{v}$ and rotational velocity $\boldsymbol{\omega}$, both referenced to the CCS. Since the rotational velocity, and hence the axis of rotation, is assumed to be constant, it follows that

$$\dot{\boldsymbol{\omega}} = 0 \tag{4.16}$$

Although the linear acceleration is implicitly assumed to be negligible, the instantaneous velocity in the CCS changes as the camera rotates—in other words, although the velocity vector points in the same direction always (to an external observer), it is referenced to a rotating coordinate system. This is expressed by

$$\dot{\mathbf{v}} = -\boldsymbol{\omega} \times \mathbf{v} \tag{4.17}$$

The imaging model is the same as that used in Approach I.

4.2.2 Recursive Formulation

The following vector of states is chosen:

$$\mathbf{s}(t) = \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega} \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_M \end{pmatrix} \tag{4.18}$$

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

$$\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t)) = \begin{pmatrix} -\boldsymbol{\omega} \times \mathbf{v} \\ \mathbf{0} \\ -\boldsymbol{\omega} \times \mathbf{p}_1 - \mathbf{v} \\ -\boldsymbol{\omega} \times \mathbf{p}_2 - \mathbf{v} \\ \vdots \\ -\boldsymbol{\omega} \times \mathbf{p}_M - \mathbf{v} \end{pmatrix} \tag{4.19}$$

Using (4.19) the discrete version of the plant equation can be written as follows:

$$\mathbf{s}(k+1) = \begin{pmatrix} R(k,k+1)\mathbf{v}(k) \\ \boldsymbol{\omega}(k) \\ R(k,k+1)(\mathbf{p}_1(k) - \mathbf{v}(k)) \\ R(k,k+1)(\mathbf{p}_2(k) - \mathbf{v}(k)) \\ \vdots \\ R(k,k+1)(\mathbf{p}_M(k) - \mathbf{v}(k)) \end{pmatrix} + \mathbf{w}_k \tag{4.20}$$

where $\mathbf{w}_k$ is a discretized plant noise term, and $R(k,k+1)$ is a 3×3 matrix representing the rotation between frames k and $k+1$. If the angular velocity $\boldsymbol{\omega}(k) = (\omega_x, \omega_y, \omega_z)^T$ is small, this matrix is approximately given by

$$R(k,k+1) = \begin{pmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{pmatrix} \tag{4.21}$$

Since the plant equations are nonlinear, it is necessary to compute the linearized plant function F, defined by

$$F = \left. \frac{\partial \mathbf{f}(\mathbf{s})}{\partial \mathbf{s}} \right|_{\mathbf{s} = \hat{\mathbf{s}}(k|k-1)} . \tag{4.22}$$

where the time dependency has not been shown. It is convenient here to define a notation for the cross product of two 3×1 vectors as follows:

$$\mathbf{x} \times \mathbf{y} = \boxed{\mathbf{x}}\mathbf{y} \tag{4.23}$$

where

$$\boxed{\mathbf{x}} \triangleq \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ x_2 & x_1 & 0 \end{pmatrix} \tag{4.24}$$

Using this notation, we can write F as follows:

$$F = \begin{pmatrix} -\boxed{\boldsymbol{\omega}} & \boxed{\mathbf{v}} & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ -I_3 & \boxed{\mathbf{p}_1} & -\boxed{\boldsymbol{\omega}} & 0 & 0 & \cdots & 0 \\ -I_3 & \boxed{\mathbf{p}_2} & 0 & -\boxed{\boldsymbol{\omega}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \ddots & & \vdots \\ -I_3 & \boxed{\mathbf{p}_M} & 0 & 0 & \cdots & 0 & -\boxed{\boldsymbol{\omega}} \end{pmatrix} \tag{4.25}$$

The measurement equation is

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{s}) + \mathbf{n}(k) = \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \\ \boldsymbol{\rho}_1 \\ \boldsymbol{\rho}_2 \\ \vdots \\ \boldsymbol{\rho}_M \end{pmatrix}_{t_k} + \mathbf{n}(k) \tag{4.26}$$

The linearized measurement function H is of the form

$$H = \begin{pmatrix} I_3 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & I_3 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \alpha_1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \alpha_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_M \end{pmatrix} \tag{4.27}$$

where the α_i terms are given by

$$\alpha_i = \frac{\partial \rho_i}{\partial \mathrm{p}_i} = \begin{pmatrix} \frac{f_x}{z_i} & 0 & f_x \frac{x}{z_i^2} \\ 0 & \frac{f_y}{z_i} & f_y \frac{y}{z_i^2} \end{pmatrix} \tag{4.28}$$

4.2.3 Experimental Results

The algorithm presented in this section was tested on synthetic and real data. The unknown parameters are the 3-D locations of feature points in the CCS. The results are shown in Figs. 4.17–4.19. In the first synthetic example, the camera velocities are larger than in the second example. The results indicate that the recursive algorithm converges faster if the camera motion is larger. This is analogous to the observation that stereo depth computations are more accurate if the baseline of the stereo cameras is bigger.

The real image sequence used is the Rocket sequence, which was described in a previous section. Measured camera kinematics are derived from the position ground truth available with the image sequence. The structure estimates are initialized using motion stereo over the first two image frames (see Appendix B). The results of recursive estimation are shown in Fig. 4.19, as a function of the frame number. The true parameter values are shown by solid lines, and their estimates by dashed lines.The dimensions are metres/second for the translational velocity and metres for the structure parameters. The sampling period is assumed to be one second.

It is evident from the first two graphs that there is considerable fluctuation in the linear and angular velocities, particularly in the latter. This is obviously a deviation from the constant velocity assumption. These fluctuations are smoothed over by the EKF, the extent of smoothing depending on the relative values chosen for the covariances of the plant and measurement noises. These covariances have to be chosen with some care; they have to reflect the various uncertainties in the problem. The measurement noise should ideally be determined by calibration, and should take into account quantization noise, sensor noise, unmodelled lens distortions, etc. The plant noise should reflect the degree of nonlinearity of the model as well as the extent of model deviations expected. Lower values for plant noise lead to greater smoothing.

Range estimation results for four of the seven points tracked are shown in Fig. 4.19. These points are at a distance of 15–40 metres from the camera at the start of the motion. Points 1 and 3 have large errors initially, and these errors are progressively reduced as the EKF uses the information from the remaining frames in the sequence. The effects of filtering are somewhat less apparent for the remaining points, because their initial estimates are themselves fairly accurate. We were unable to get comparable range results for points more than 100 metres away from the camera.

The primary mechanism in this approach, structure-from-motion, is based on image plane displacements of feature points due to camera motion (typically less than 10 pixels between frames). The structure of feature points, as in stereo triangulation, bears an inverse relationship to the magnitudes of the image displacements, resulting in structure estimates that are not very robust for small camera motion and for distant objects. Even when the camera motion is significant, points very close to the focus of expansion may not be accurately ranged. If high measurement noise is also present, the accuracy will degrade still further. However, with reliable inertial measurements, high image resolution and accurate camera calibration, range results may be comparable to those obtained using active methods.

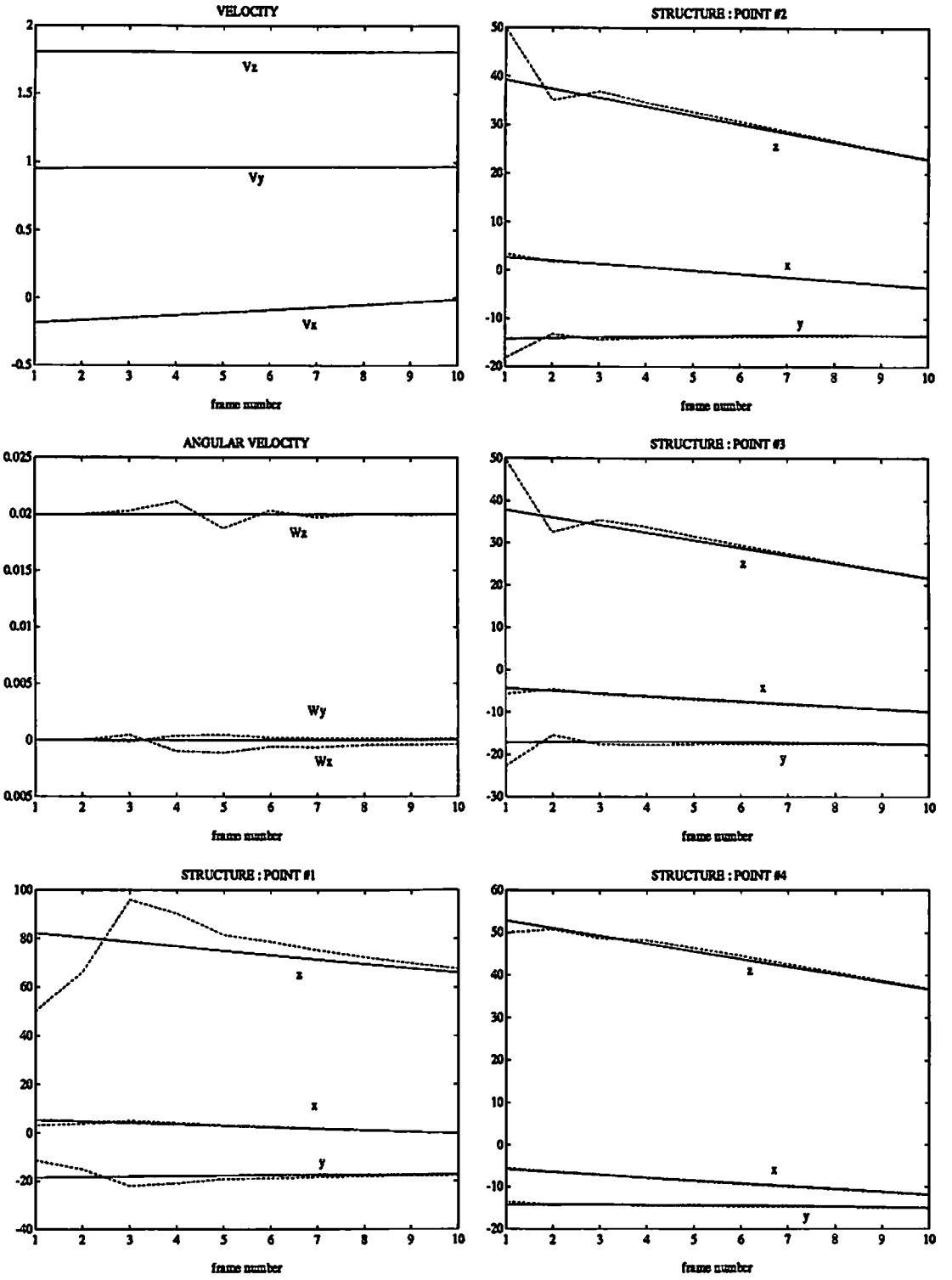

Figure 4.17: Approach II IEKF results for synthetic data: Example 1

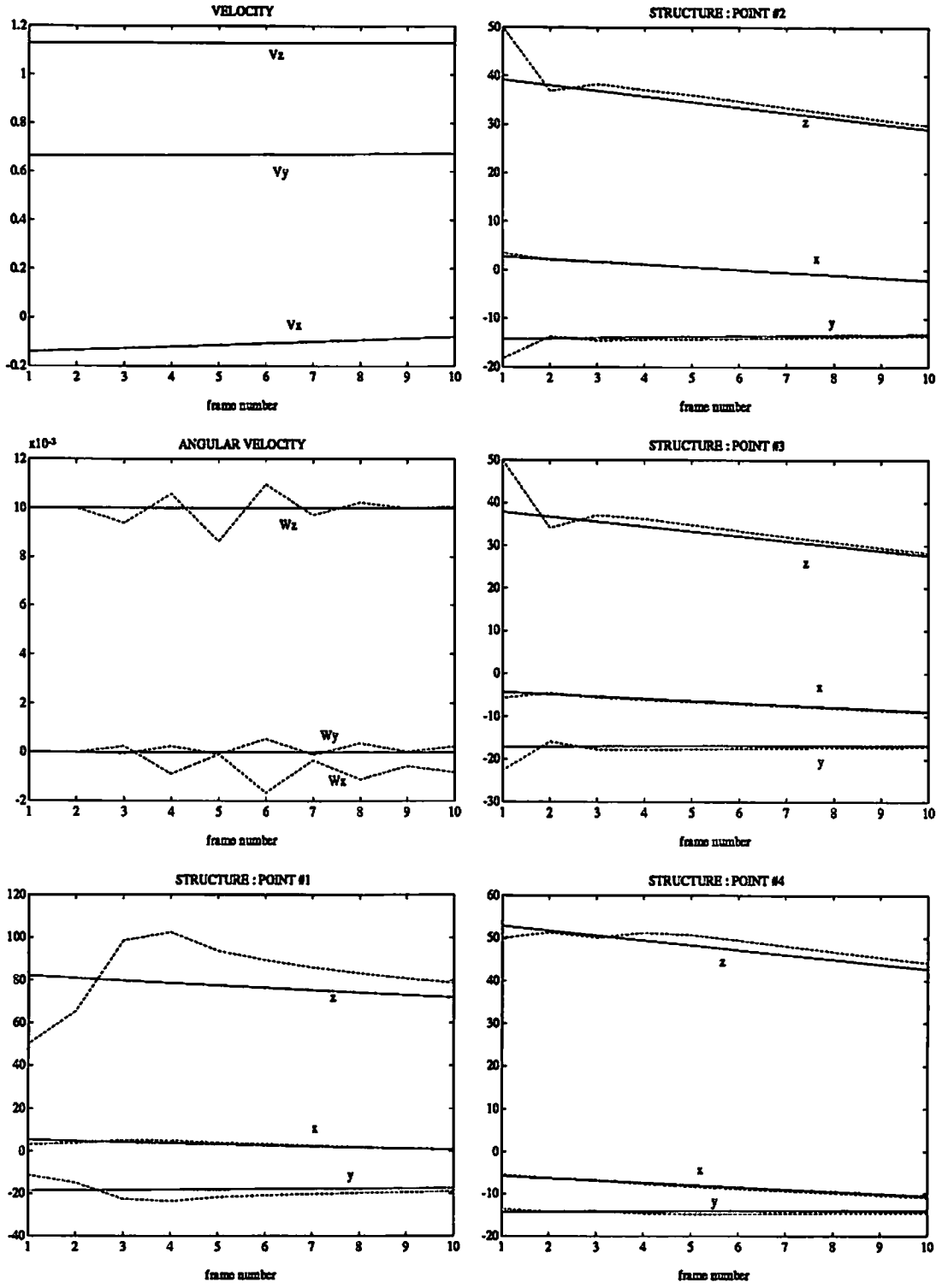

Figure 4.18: Approach II IEKF results for synthetic data: Example 2

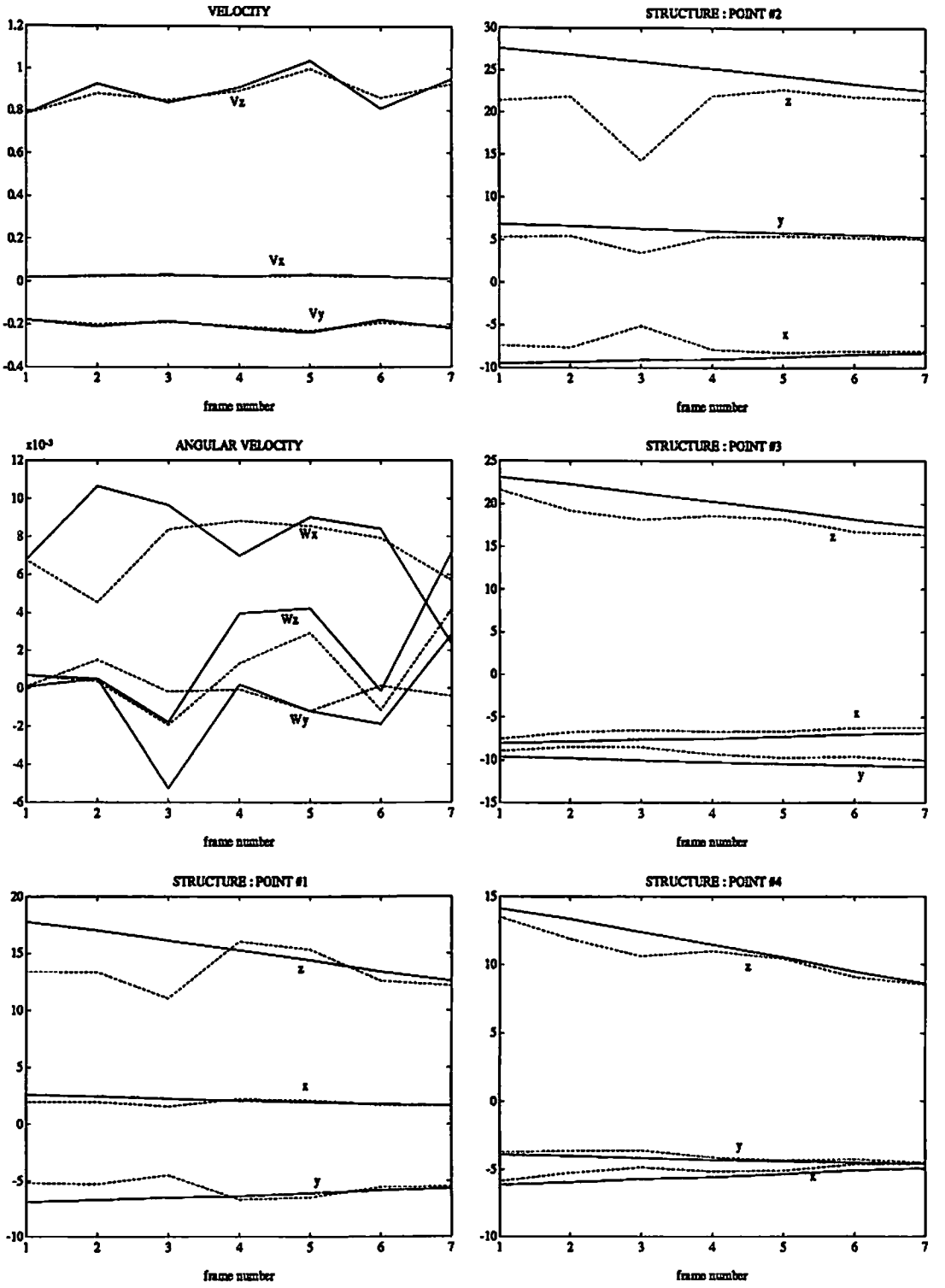

Figure 4.19: Approach II IEKF results for the Rocket sequence.

4.3 Initialization

An approximate nonlinear filter like the IEKF usually requires a reasonably good initial guess in order to converge. In our past work [10, 18], we have used a least-squares batch approach over the first few image frames to obtain an initial estimate. In this section, we examine this and other approaches to obtaining the initial guess, and compare the results obtained with each. The discussion here focuses on Approach I for passive navigation. The initial guess requirement for Approach II is not critical, and hence is not discussed here, but the results of this section may be extended to that approach as well.

4.3.1 Batch Formulation

For the approach discussed in Section 4.1, the unknown model parameters form a d -dimensional vector θ, given by

$$\theta = \begin{pmatrix} \mathbf{p}_R(0) \\ \mathbf{v} \\ \mathbf{a} \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_{M_u} \end{pmatrix} \tag{4.29}$$

The orientation of the camera is represented by the 3-component vector $\mathbf{a}$ in the above expression, instead of the 4-component unit quaternion $\mathbf{q}$. 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 $\mathbf{a}$ and $\mathbf{q}$ is given by (4.3) and the following equation:

$$\mathbf{a} = \begin{pmatrix} n_1\,\theta \\ n_2\,\theta \\ n_3\,\theta \end{pmatrix} \tag{4.30}$$

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

$$\boldsymbol{\rho}_i(k) = h_i(\boldsymbol{\theta}, k) + \mathbf{n}_i(k) \; ; \; i = 1, \ldots, M \; ; \; k = 1, \ldots, N, \tag{4.31}$$

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 (4.8). 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 $\boldsymbol{\rho}_i(k)$ and the observation model (4.8). 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.

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=0}^{N-1} \sum_{i=0}^{M-1} \|\boldsymbol{\rho}_i(k) - h_i(\boldsymbol{\theta}, k)\|^2 \tag{4.32}$$

This least-squares minimization can be done using a standard optimization program, such as the ones used in [15, 45]. Any knowledge about the ranges or values of the parameters can be used to reduce the search space for a solution. Results are shown in Table 4.2.

4.3.2 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

parameter	true value	synth. ex. 1	synth ex. 2	Rocket seq.
p	2.6223	2.6493	2.7206	2.7910
	-0.9764	-0.9896	-0.9652	-2.4334
	1.2291	1.1979	1.4850	0.6845
v	0.0000	-0.0026	-0.0340	0.0076
	-0.5000	-0.5199	-0.5041	-0.3004
	0.6000	0.6195	0.5159	0.7811
a	0.6791	0.6784	0.6841	0.6414
	0.2049	0.2052	0.2074	0.2047
	0.0804	0.0790	0.0729	0.0576
$\mathbf{p}_1$	-0.3432	-0.2947	0.0935	-0.2155
	-24.0720	-24.8835	-21.7560	-25.1813
	15.6780	16.1074	14.2698	15.9375
$\mathbf{p}_2$	-0.3164	-0.3293	-0.2271	-0.4029
	-31.3890	-31.5967	-30.3622	-33.2501
	21.9370	22.1114	21.2843	22.7937
$\mathbf{p}_3$	-0.1964	-0.2357	0.3828	-0.2992
	-13.8470	-14.1233	-11.5285	-15.1524
	10.5080	10.6892	8.9826	10.9141
$\mathbf{p}_4$	-0.2264	-0.2563	0.6799	-0.8556
	-20.1580	-20.4462	-14.5313	-23.7982
	18.4360	18.6887	13.5331	21.1771
$\mathbf{p}_5$	-0.1764	-0.2338	0.1454	-0.6456
	-10.8410	-11.1329	-9.8557	-11.9839
	17.4630	17.8969	15.7233	19.5261
$\mathbf{p}_6$	-0.2164	-0.2311	0.1667	-0.5256
	-14.2320	-14.4354	-12.6393	-15.4828
	28.0400	28.3878	24.6411	31.4431
$\mathbf{p}_7$	-0.3764	-0.3356	0.3567	-0.0433
	-24.7840	-27.4434	-21.2779	-23.9073
	11.0530	12.0585	9.7499	10.7841

Table 4.2: Results of batch estimation, Approach I. Estimates of camera position, velocity and orientation, and structure estimates of the first seven unknown points are shown.

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.[3] Similar methods are used in [14, 68].

For the purpose of deriving the CRLBs, we represent the conditional p.d.f. of the measurements as multivariate Gaussian,[4] of the form:

$$p(\mathbf{z}\ /\theta) = \prod_{k=0}^{N-1} \prod_{i=0}^{M-1} \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{\|\boldsymbol{\rho}_i(k)-h_i(\boldsymbol{\theta},k)\|^2}{2\sigma^2}} \tag{4.33}$$

where $\mathbf{z}$ 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

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

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:

$$C_\theta \geq J^{-1} \tag{4.34}$$

[3]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.

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

Using the expression in (4.33) for the conditional p.d.f. of the data,

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

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

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

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

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 Section 4.1, 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 $\hat{\boldsymbol{\theta}}$ is used as an approximation to the true parameter vector $\boldsymbol{\theta}$ in the derivation of CRLBs.

4.3.3 Initialization Using an IEKF-Smoother

The disadvantage of the batch approach is that the procedure is iterative, and may require several hundred iterations to converge. Further, it performs poorly when the input data do not conform closely to the assumed model. On the other hand, the recursive approach is fast, has a fixed number of steps, and can handle model deviations better.

We found that it is possible to use an iterated extended Kalman filter-smoother (IEKF smoother) instead of the least-squares batch approach. The procedure is as follows: the motion parameters are first initialized to some trivial, but reasonable, values (for instance, all the velocities can be set to zero,

and the camera's initial position can be set to zero, and it can be assumed to aligned with the WCS). Then the IEKF is run on the first 2–8 frames of the sequence in a purely motion-from-structure mode, using the known landmarks. A fairly large number of local iterations (2–6) are used. In the next step, the IEKF is run *in reverse* back to the first frame. This combination of forward iterated filtering and backward smoothing results in a very good initial guess for the motion parameters. The structure parameters are initialized by assuming the 3-D points to lie at the intersection of the corresponding direction rays (from the origin of the CCS to the corresponding image point) with a sphere of some arbitrary radius (say 50 metres) around the camera. With these initial values, the recursive estimator can converges rapidly. Further improvement may be possible if the initial structure estimates are obtained by motion stereo over the first two or three frames.

Initial estimates obtained using this method are shown in Figs. 4.20-4.22.

4.3.4 Error Model for Structure Estimates

One of the very important factors in the performance of a recursive estimator is the suitability of the error models used. We use the term "suitability," rather than "accuracy," because experimental results show that the actual values of the error covariances are not important; what is crucial is that the *form* of the error model should capture the uncertainties in the physical models of motion and imaging. We have observed that structure estimates, in particular, are strongly affected by the choice of error model. Structure errors tend to be very unequally distributed in the three dimensions, due to the highly nonlinear nature of the imaging process; errors in range are of far greater magnitude than errors in azimuth and elevation in the CCS. If the error probabilities are represented approximately by an ellipsoid, the ellipsoid tends to be very eccentric. This has to be taken into account while initializing the recursive estimator. In fact, structure estimates converge rapidly even from very wild initial guesses, *as long as* decent values are used for the error covariance.

The first problem we address is this: given the image coordinates $\boldsymbol{\rho} = (X, Y)^T$ of a point, estimate its position $\mathbf{p}_c$ in 3-D, relative to the camera,

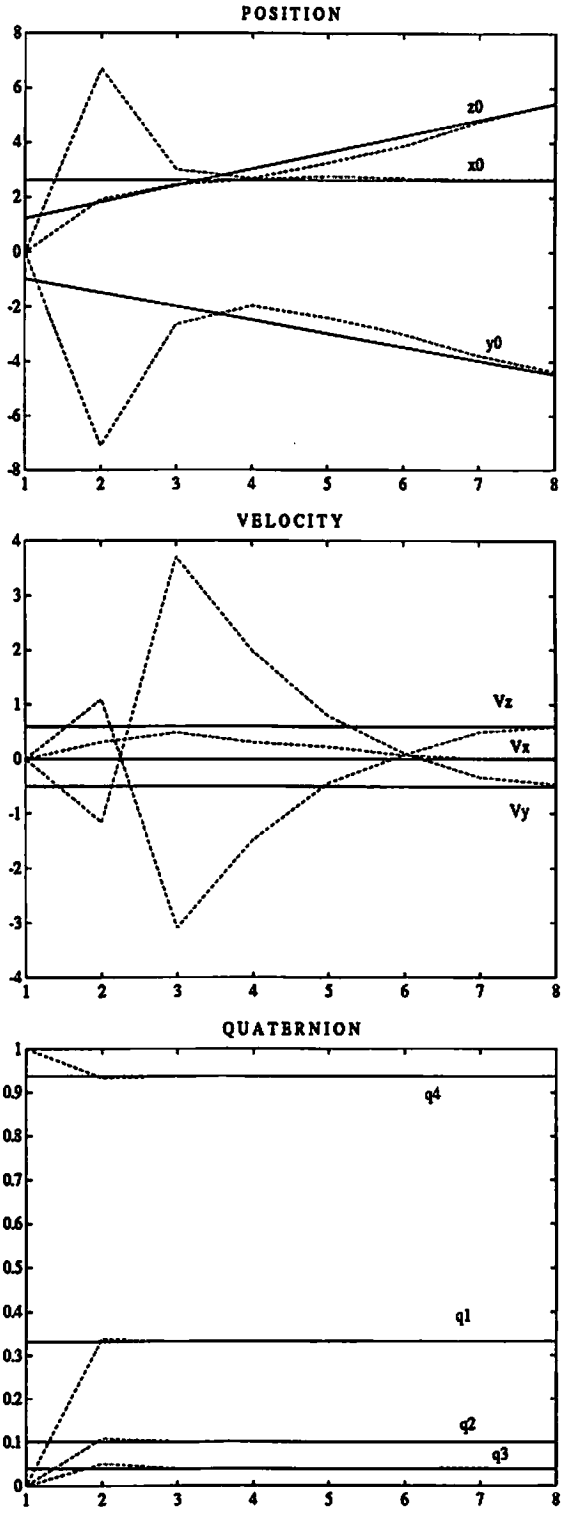

Figure 4.20: Synthetic data, example 1: initial guess using IEKF smoother

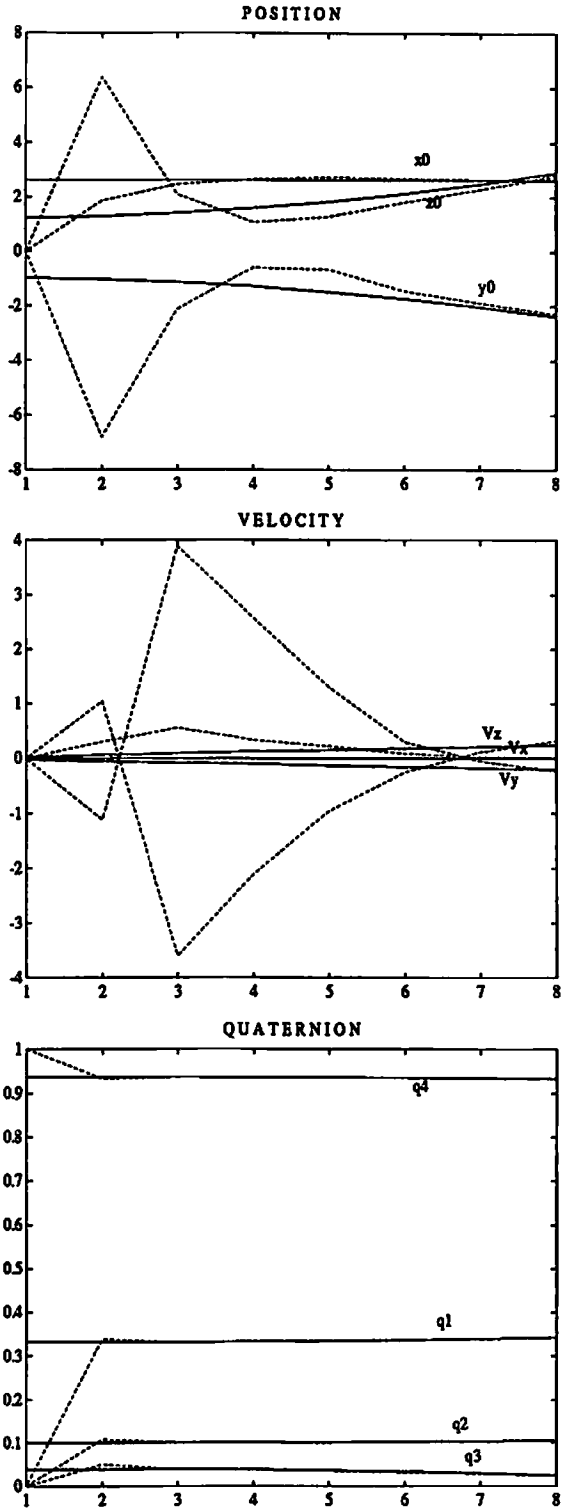

Figure 4.21: Synthetic data, example 2: initial guess using IEKF smoother

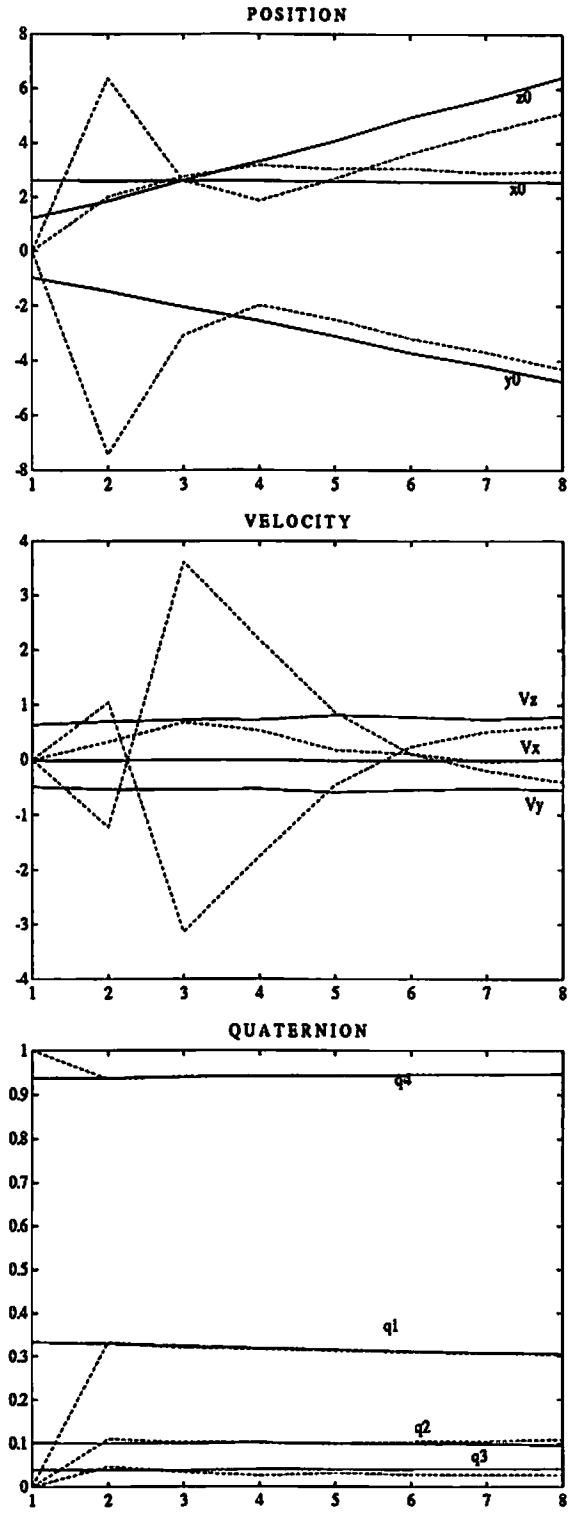

Figure 4.22: Rocket sequence: initial guess using IEKF smoother

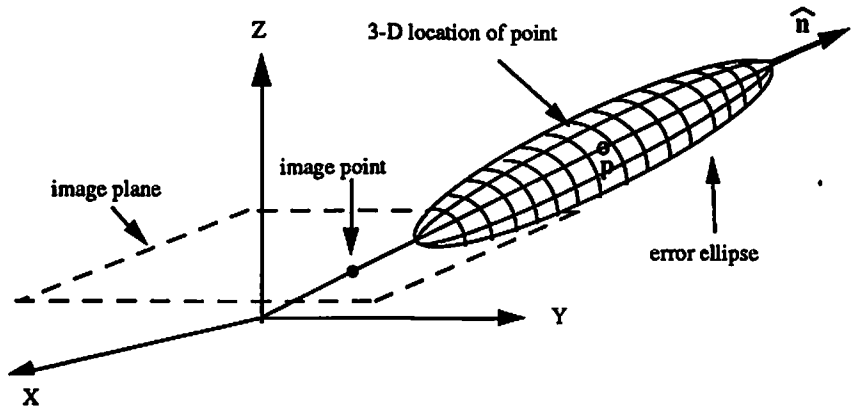

Figure 4.23: Ellipsoidal approximation of structure error

and the covariance associated with this estimate, assuming it to be at a(n) (arbitrary) distance z from the camera. The estimate of the point's 3-D location is simple; it is given by

$$\hat{\mathbf{p}}_c = z \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} zX \\ zY \\ z \end{pmatrix} \tag{4.35}$$

It's error covariance can be approximated by an eccentric ellipsoid with its major axis aligned with the direction ray connecting the origin of the CCS to the point (Fig. 4.23). This cigar-shaped ellipsoid is characterized by a radial deviation σ_r and a lateral deviation σ_l, such that

$$\sigma_r \gg \sigma_l \tag{4.36}$$

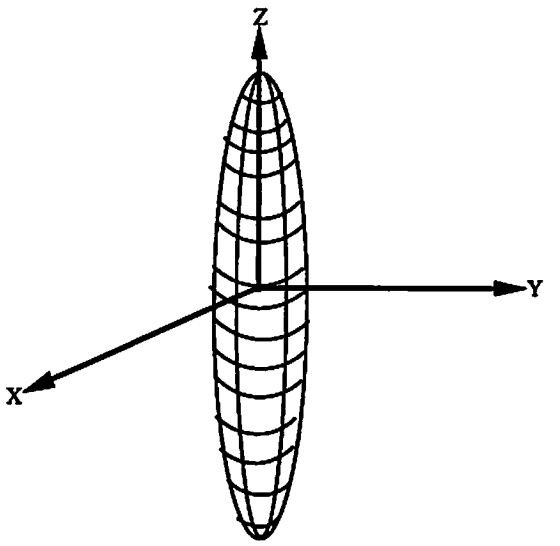

Figure 4.24: Standard form of ellipsoid

To obtain a mathematical expression for it, consider the "standard" form of an ellipsoid in (Fig. 4.24), which is defined by the equation

$$\mathbf{p}^T \underbrace{\begin{pmatrix} \sigma_l^2 & 0 & 0 \\ 0 & \sigma_l^2 & 0 \\ 0 & 0 & \sigma_r^2 \end{pmatrix}^{-1}}_{\triangleq C_1} \mathbf{p} = const \tag{4.37}$$

This ellipsoid has the same shape as the one in (Fig. 4.23), but its major axis is aligned with the z-axis, rather than with the direction ray. We need to determine the rotation which will align the unit vector in the z-direction, $\mathbf{z} = (0, 0, 1)^T$, with the unit vector along the direction ray, $\mathbf{n} = \mathbf{p}/\|\mathbf{p}\|$. Let R_0 be the corresponding rotation matrix . Then,

$$R_0 \mathbf{z} = \mathbf{n} \tag{4.38}$$

If $\mathbf{q_0}$ is the corresponding quaternion, the above equation can be written using quaternion algebra [67] as

$$\mathbf{q} \star \mathbf{z} \star \tilde{\mathbf{q}} = \mathbf{n} \tag{4.39}$$

Post-multiplying both sides by $\mathbf{q}$

$$\mathbf{q} \star \mathbf{z} = \mathbf{n} \star \mathbf{q}$$

This can be written as a standard matrix-vector product

$$([\mathbf{z}]_r - [\mathbf{n}]_l)\,\mathbf{q} = 0 \tag{4.40}$$

where

$$[\mathbf{z}]_r = \begin{pmatrix} 0 & -\mathbf{z}' \\ \mathbf{z} & -\boxed{\mathbf{z}} \end{pmatrix}$$

and

$$[\mathbf{n}]_l = \begin{pmatrix} 0 & -\mathbf{n}' \\ \mathbf{n} & \boxed{\mathbf{n}} \end{pmatrix}$$

Equation (4.40) can now be solved to obtain the quaternion of rotation $\mathbf{q}$. R_0 can then be computed. Using R_0 and the standard form of the ellipsoid in (4.37), we can now write the equation of the error ellipsoid in (Fig. 4.23) as:

$$\mathbf{p}^T \underbrace{R_0^T C_1^{-1} R_0}_{\triangleq C_c^{-1}} \mathbf{p} = const \tag{4.41}$$

If the rotation between the CCS and the WCS is given by the matrix R_{cw}, the equation of the error ellipsoid in the WCS can be written as:

$$\mathbf{p}^T \underbrace{R_{cw}^T C_c^{-1} R_{cw}}_{\triangleq C_w^{-1}} \mathbf{p} = const \tag{4.42}$$

4.4 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 chapter dealt with the development of such an approach for the passive navigation problem, using simple motion models in conjunction with 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. In the following chapters, we apply similar principles for object tracking and traffic image analysis.

Chapter 5

Model Evaluation

Model-based formulations are at the core of our approach to motion analysis. One of the criticisms levelled against such formulations is the lack of a solid theoretical basis; the performance of such formulations cannot, in general, be predicted in advance. In particular, conditions for uniqueness and of solution and robustness of parameter estimates, such as those that have been established for two-frame and other fixed-frame methods (e.g. [64, 29, 67, 26]), have not been extended to general long-frame methods, except for specific cases. Uniqueness conditions for the case of a stationary camera and moving object are presented in [15] for uniform translation, using monocular data and in [73] for constant acceleration and constant precession, using stereo data. These results are not readily generalizable to other situations. Further, uniqueness results by themselves are not sufficient to analyze a model based approach; one could even argue that uniqueness results are largely irrelevant for practical applications, since one would anyway use a highly overdetermined system in order to combat the effects of noise and modelling errors. The crucial thing is to be able to predict the accuracy with which the model parameters will be estimated from the input data. This kind of "robustness" analysis has been done for the two-frame structure-from-motion problem in [26], in an algorithm-independent way. Their results are not applicable to motion analysis using longer image sequences.

In this chapter, we outline an empirical approach to analyzing long-frame model-based formulations, which is applicable to any problem which falls into this category. The basic idea is to study the eigen decomposition of the Hessian of the batch objective function at a solution point, and to determine the shape of the objective function in the vicinity of the solution. This gives important clues about the global nature of the objective function. Similar ideas have been used in [9] to determine local uniqueness of solution, and in [14], where CRLB's are used as approximate performance bounds. In this chapter, we go much further, looking not just at the Hessian or its eigenvalues, but also at its eigenvectors, and the rows of the eigenvector matrix. The approach is developed for the passive navigation problem (Approach I), as a test case, but is applicable to other model-based formulations as well.

5.1 The Objective Function

The objective function of the batch formulation captures most of the essential features of the model-based formulation. Consider the objective function for the passive navigation problem, which minimizes the squared discrepancy between the measurements $\rho_i(k)$ and the corresponding values $h_i(\theta, k)$ predicted by the model, i.e.

$$G = \sum_{i=0}^{N-1} \sum_{i=0}^{M-1} \|\rho_i(k) - h_i(\theta, k)\|^2 \tag{5.1}$$

A (local) minimum of the objective function G should satisfy the following conditions:

$$\nabla G|_{\theta=\theta_{min}} \triangleq \left.\frac{\partial G}{\partial \theta}\right|_{\theta=\theta_{min}} = 0 \tag{5.2}$$

$$\nabla^2 G\Big|_{\theta=\theta_{min}} \triangleq \left.\frac{\partial^2 G}{\partial \theta^2}\right|_{\theta=\theta_{min}} > 0 \tag{5.3}$$

The first equation, which states that the gradient of the function should be zero at the minimum, gives the condition for the existence of a stationary point. In order for this stationary point to be local minimum of the function, it should satisfy the second condition, which states that the Hessian of the objective

function evaluated at the point should be positive definite. In other words, all the eigenvalues of the Hessian should be positive.

In most of our work, we take for granted that these conditions are satisfied. What is of great interest to us, then, is the relative magnitudes of the eigenvalues, and the eigenvectors they correspond to. This is because in the vicinity of a minimum point, the objective function can be approximated by an ellipsoid (in d dimensions) whose axes correspond to the eigenvectors of the Hessian, with lengths proportional to the inverse of the square root of the corresponding eigenvalues. This is seen more clearly by looking at a Taylor series expansion of the objective function near the solution point.

$$\begin{aligned} G(\theta) &= G(\theta_{min}) \\ &+ (\theta - \theta_{min})^T \nabla G(\theta)|_{\theta=\theta_{min}} \\ &+ (\theta - \theta_{min})^T \nabla^2 G(\theta)|_{\theta=\theta_{min}} (\theta - \theta_{min}) \\ &+ \cdots \end{aligned} \tag{5.4}$$

The first term in the above equation is constant, and can be ignored. The second term vanishes because of condition (5.2). We are therefore left with the third term and higher order terms. The latter may be ignored for $\theta \simeq \theta_{min}$. Defining the Hessian

$$H(\theta) \triangleq \nabla^2 G(\theta)|_{\theta=\theta_{min}} \tag{5.5}$$

we may thus write G approximately as

$$G(\theta)|_{\theta \simeq \theta_{min}} = \frac{1}{2}(\theta - \theta_{min})^T H(\theta)(\theta - \theta_{min}) \tag{5.6}$$

By studying the properties of the Hessian H, we can gain insight into the local structure of the objective function near a solution point, and thereby into the nature of the function itself. If the objective function does not have a good structure, it implies that the model-based formulation is flawed. What is precisely meant by a "good" structure will become clearer at a later stage of the analysis.

The gradient of the objective function, from (5.1), is obtained as

$$\nabla G = \sum_{k=0}^{N-1} \sum_{i=0}^{M-1} \frac{\partial(\rho_i(k) - h_i(\theta, k))^T}{\partial \theta} (\rho_i(k) - h_i(\theta, k)) \tag{5.7}$$

Defining the $2 \times d$ matrices $d_i(k)$

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

we can write

$$\nabla G = \sum_{k=0}^{N-1} \sum_{i=0}^{M-1} d_i(k)\theta(\rho_i(k) - h_i(\theta, k)) \tag{5.8}$$

The Hessian is now derived as follows:

$$H = \nabla(\nabla G) = \sum_{k=0}^{N-1} \sum_{i=0}^{M-1} d_i(k)\theta^T d_i(k)\theta + \sum_{k=0}^{N-1} \sum_{i=0}^{M-1} \nabla(d_i(k)\theta)(\rho_i(k) - h_i(\theta, k)) \tag{5.9}$$

Setting $\theta = \theta_{min}$ in the above equation, we get

$$H = H(\theta)|_{\theta=\theta_{min}} = \sum_{k=0}^{N-1} \sum_{i=0}^{M-1} d_i(k)^T d_i(k) \tag{5.10}$$

The other term vanishes because in the absence of noise, $(\rho_i(k) - h_i(\theta_{min}, k)) = 0$ for all i and k. The derivation of the $d_i(k)$ matrices is similar to that of the H_i terms in Chapter 4. It can be seen that the Hessian at the solution point θ_{min} is independent of the measurements.

In the rest of this chapter, we adopt a "hands-on" approach, looking at a specific instance of the model used in Chapter 4.

5.2 Case Study : Passive Navigation

Let us select the following $d-$dimensional set of parameters :

$$\theta = \begin{pmatrix} \mathrm{p}_R \\ \mathrm{v} \\ \mathrm{q} \\ \mathrm{p}_1 \\ \mathrm{p}_2 \\ \vdots \\ \mathrm{p}_{M_u} \end{pmatrix}_{d\times 1} \tag{5.11}$$

where the first three components are referred to the initial time instant t_0. We are now going to look at a specific value of θ, given in Table 5.1. This will be the solution, or minimum point, at which we will study the behaviour of the objective function.

5.2.1 The Rank of H

Local uniqueness at a solution point is determined by the rank of the Hessian at that point. If the Hessian is not of full rank, it means that the solution is locally underconstrained, and therefore not globally unique. On the other hand, if the Hessian is of full rank, it implies local but not global uniqueness of solution. Physically, rank deficiency of the Hessian corresponds to one or more directions, or degrees of freedom, along which the value of the objective function does not change.

The rank of the Hessian, determined by numerical methods, is shown in Table 5.2, as a function of the number of known feature points (M_k),the number of unknown feature points (M_u) and the number of image frames (N). The rank of the Hessian must be compared with d, the dimension of the parameter vector, since the Hessian is of size $d \times d$. The following features of the Hessian can be immediately deduced:

parameter	value
$\mathbf{p}$	2.6223
	-0.9764
	1.2291
$\mathbf{v}$	0
	-0.5000
	0.6000
$\mathbf{q}$	0.3324
	0.1003
	0.0394
	0.9370
$\mathbf{p}_1$	15.8866
	-123.2330
	144.9800
$\mathbf{p}_2$	6.5806
	-36.4954
	50.2220
$\mathbf{p}_3$	-0.5052
	-44.1819
	27.5550
$\mathbf{p}_4$	0.1266
	-30.9723
	32.3000
$\mathbf{p}_5$	-0.3432
	-24.0720
	15.6780

Table 5.1: Parameter values chosen for the case study

$M_u = 0$, $d = 10$

NM_k	0	1	2	3	4
0	0	0	0	0	0
1	0	2	4	6	6
2	0	4	8	9	9
3	0	5	8	9	9

$M_u = 1$, $d = 13$

NM_k	0	1	2	3	4
0	0	0	0	0	0
1	2	4	6	8	8
2	4	8	11	12	12
3	5	8	11	12	12

$M_u = 2$, $d = 16$

NM_k	0	1	2	3	4
0	0	0	0	0	0
1	4	6	8	10	10
2	8	11	14	15	15
3	8	11	14	15	15

Table 5.2: Rank of the Hessian for various combinations of M_k, M_u and N

1. The rank of the Hessian, relative to d, increases with the number of known points M_k and the number of frames N.

2. For a fixed number of frames, the Hessian attains its maximum rank for $M_k = 3$. Further increase in M_k does not improve the rank.

3. With two exceptions, for a fixed number of known points, the Hessian attains its maximum rank for $N = 2$. Further increase in N does not improve the rank. The two exceptions are: $M_u = 0; M_K = 1$ and $M_u = 1; M_k = 0$.

4. The Hessian never attains full rank. Its maximum rank is $d - 1$.

Most of the above observations are intuitively obvious. More known points means more constraints; more image frames means more data. This explains observation (1). It can be shown, from simple geometrical considerations, that at least three non-collinear points are required to unambiguously locate a camera. This leads to observation (2). A minimum of two frames are needed to estimate velocities (observation 3). The final observation will be explained in a later section.

5.2.2 The Eigenvalues of H

The rank of the Hessian determines local uniqueness; the range of its eigenvalues determines local "conditionedness" of the objective function. The eigenvalues of the Hessian for $M_k = 3$, $M_u = 2$ and $N = 4$ are tabulated in Table 5.3, and shown graphically in Fig. 5.1. Let them be denoted as λ_1 to λ_{16}, in increasing order. It can be seen that the first eigenvalue λ_1 is so small relative to the others that it can be treated as zero for all practical purposes. This is not surprising, because, as Table 5.2 shows, the Hessian is rank-deficient. (The reason for this will become clear when we look at the eigenvectors in the next section.)

Even after deleting λ_1, the range of eigenvalues, shown in Fig. 5.2, is very large (about 10^9). This implies that the local ellipsoidal approximation of the objective function is extremely eccentric. Minimization techniques which do

eigenvalue	value
λ_1	1.7742950e-10
λ_2	1.0444110e-02
λ_3	1.8810705e-01
λ_4	8.1348714e+00
λ_5	3.5169326e+01
λ_6	4.7979428e+01
λ_7	6.9335979e+01
λ_8	5.4268867e+02
λ_9	8.7092143e+02
λ_{10}	1.3647324e+03
λ_{11}	2.4613791e+03
λ_{12}	4.3714300e+03
λ_{13}	8.5598526e+03
λ_{14}	5.0332741e+05
λ_{15}	1.0778459e+07
λ_{16}	1.7921092e+07

Table 5.3: Eigenvalues of the Hessian

not take this fact into account may not perform very well. This is applicable to gradient-based methods such as conjugate gradients and gradient descent. Methods which use the Hessian in addition to the gradient, such as the modified Newton's approach, are likely to work better. It may also be possible to re-parametrize the problem in such a way as to reduce the range of eigenvalues.

5.2.3 The Eigenvectors of H

The somewhat hazy picture of the objective function that has emerged becomes clearer when we look at the eigenvectors of H, some of which are tabulated in Tables 5.4. Let us denote by $e_1, e_2, \ldots$ the eigenvectors corresponding to eigenvalues $\lambda_1, \lambda_2, \ldots$.

Let us first try to find an explanation for the rank deficiency of H. The eigenvector corresponding to the zero eigenvalue, e_1, is tabulated in the third column of Table 5.4. It can be seen that (except for numerical inaccuracies) all the components of e_1 are zero *except those corresponding to the quaternions.*

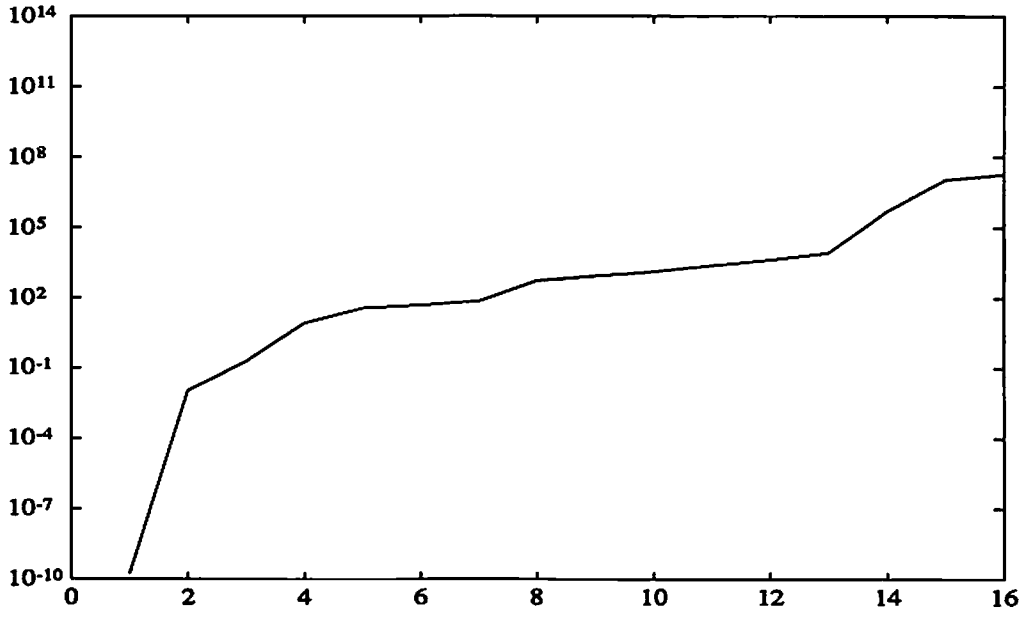

Figure 5.1: The eigenvalues for the example used for case study

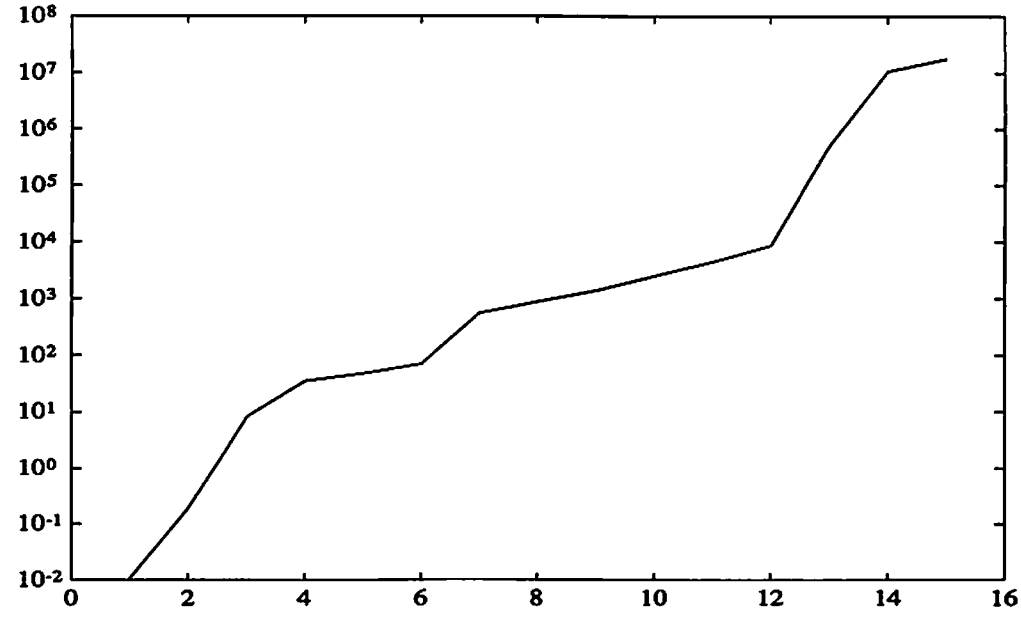

Figure 5.2: The nonzero eigenvalues for the example used for case study

parameter	e_1	e_2	e_3	e_4
p	-0.0000	-0.0001	0.0016	0.1050
	0.0000	0.0141	-0.0167	0.7384
	-0.0000	-0.0169	0.0203	-0.5491
v	0.0000	0.0000	-0.0010	-0.0532
	-0.0000	-0.0140	0.0201	-0.2862
	0.0000	0.0167	-0.0226	0.2110
q	0.3324	-0.0000	0.0000	0.0004
	0.1003	0.0000	-0.0000	-0.0005
	0.0394	-0.0000	0.0000	0.0004
	0.9370	0.0000	-0.0000	-0.0001
$\mathbf{p}_1$	-0.0000	-0.1006	-0.0480	-0.0188
	-0.0000	-0.7657	-0.3637	-0.0303
	0.0000	0.4689	0.2214	-0.0733
$\mathbf{p}_1$	-0.0000	-0.0831	0.1752	-0.0528
	-0.0000	-0.3406	0.7193	0.0285
	0.0000	0.2446	-0.5165	-0.0518

Table 5.4: The first four eigenvectors of the Hessian

This implies that e_1 lies completely in the subspace defined by the quaternion **q**. In fact, it can be shown that the nonzero subvector of e_1 is identical tp **q** upto a sign inversion. If we refer back to the introduction of quaternions in Chapter 4, we find that the four component quaternion has only three degrees of freedom, and hence has to be constrained to have unit magnitude. Since we did not enforce this condition in the objective function, we ended up with the extra degree of freedom. Thus hidden constraints in the model can be discovered by examining the eigenvectors corresponding to vanishing eigenvalues.

Eigenvectors corresponding to large eigenvalues are "good" directions, in the sense that the objective function has the desired "bowl" shape along these directions. Conversely, eigenvectors corresponding to small eigenvalues are directions along which the objective function is more or less flat, and hence hard to minimize accurately. This general idea can be developed still further by examining the *rows* of the eigenvector matrix E, defined by

$$E = [e_2\ e_3\ \cdots\ e_d] \tag{5.12}$$

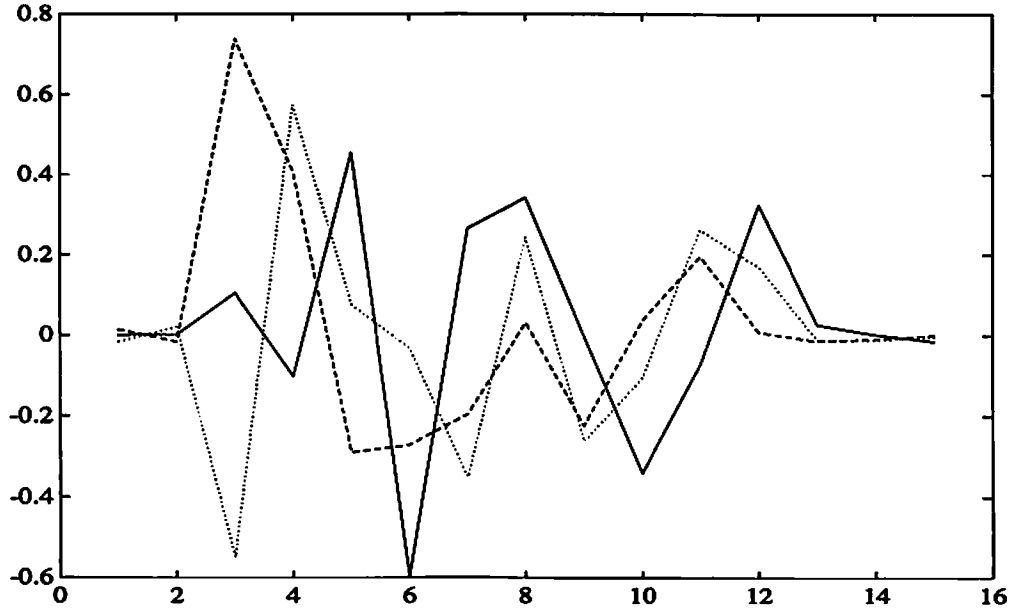

Figure 5.3: Rows corresponding to the position parameters

(We ignore the first eigenvector.) These rows can be divided into different groups, based on the components of θ they correspond to. These groups of rows are shown graphically in Figs. 5.3 to 5.7. Analogous to "good' and "bad" eigenvectors, we can also classify the unknown model parameters in θ as good or bad, depending on their contributions to good and bad eigenvectors. For instance, quaternions are "good" parameters, in the sense that they contribute to eigenvectors on the higher end of the scale, as illustrated by Fig. 5.5. This is equivalent to saying that in the subspace corresponding to the quaternions, the objective function has a nice bowl shape, and hence can be minimized with high accuracy. The structure parameters, on the other hand, contribute mainly to the bad eigenvectors (Figs. 5.6 and 5.7), indicating that it will be difficult to estimate the structure parameters accurately from the input data. These remarks may be verified by looking at the experimental results in Chapter 4.

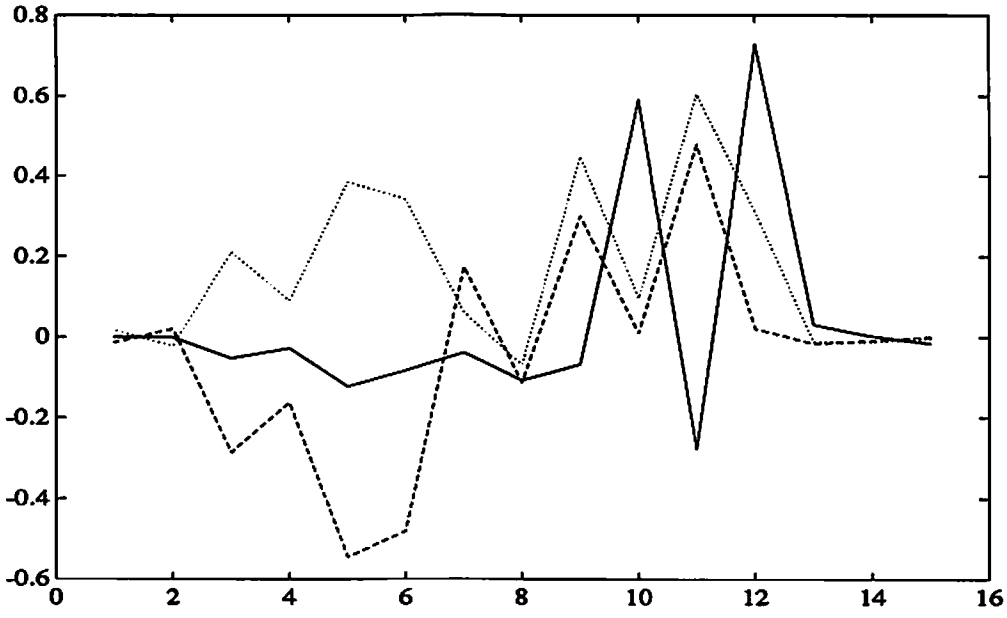

Figure 5.4: Rows corresponding to the velocities

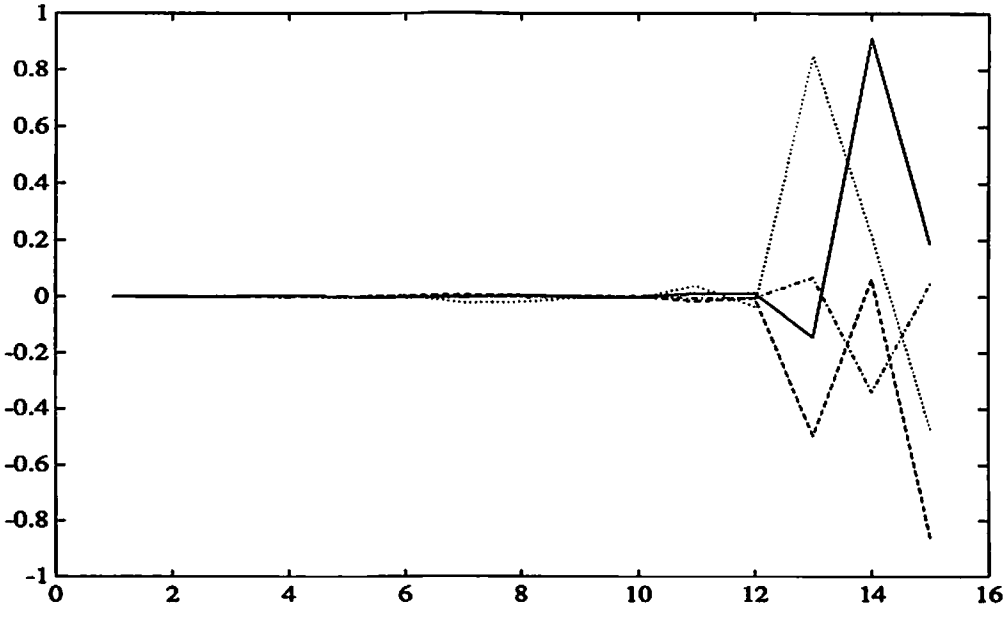

Figure 5.5: Rows corresponding to the quaternions

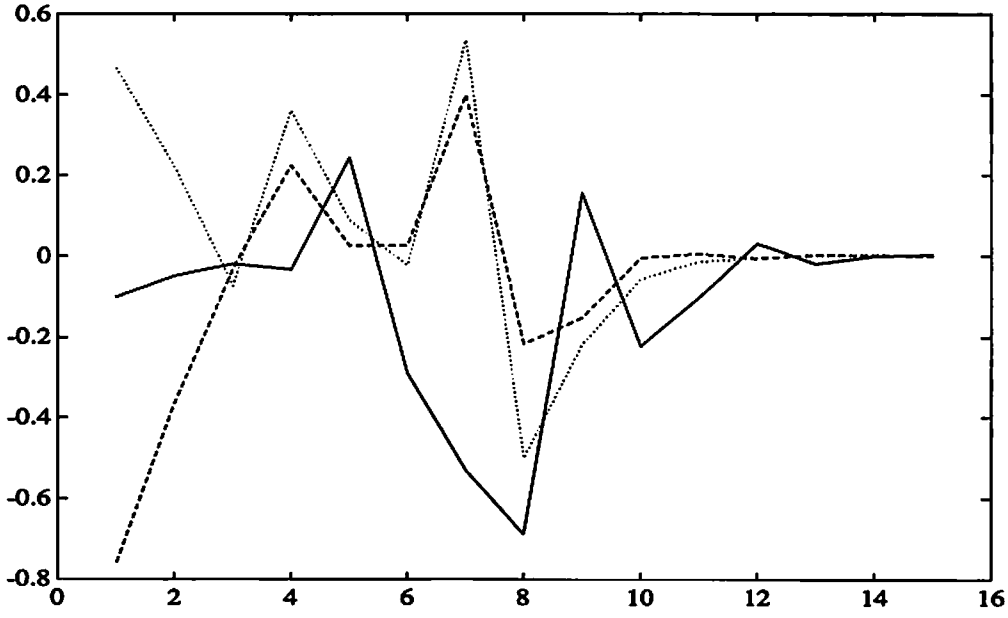

Figure 5.6: Rows corresponding to the structure parameters (Point no.1)

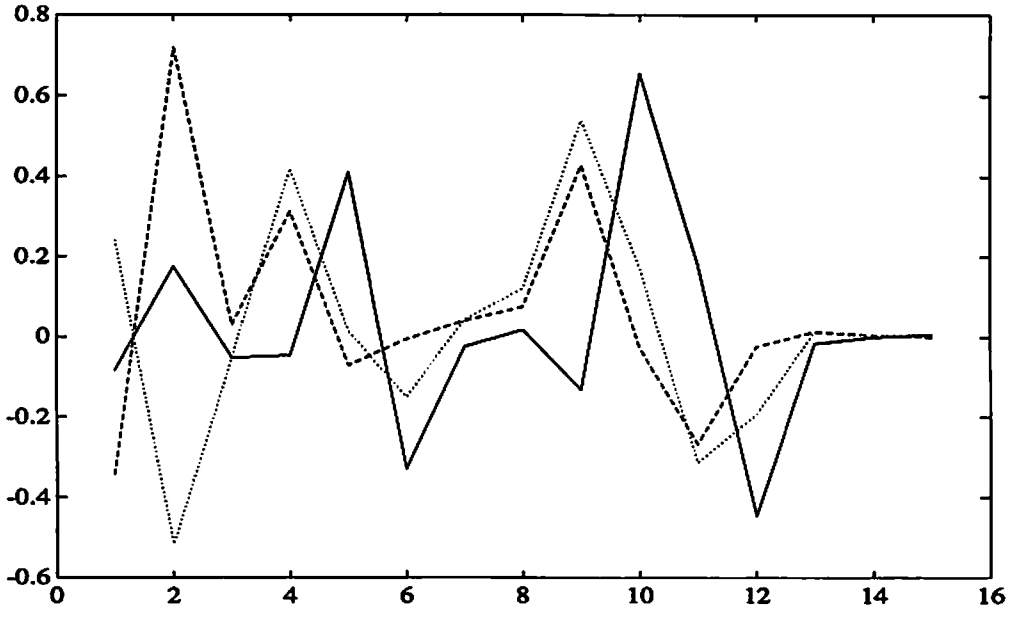

Figure 5.7: Rows corresponding to the structure parameters (Point no.2)

Chapter 6

Target Tracking

The models developed in the previous chapters were for a moving camera in a stationary environment. Though this is the most common application of motion analysis, the complementary problem of a fixed camera looking at a moving object is also of great interest. In this chapter, we present a long frame approach to the determination of the kinematics and structure of a rigid moving object based on a monocular sequence of images obtained by a stationary camera.[1]

Fig. 6.1 illustrates the basic models for motion, structure, and the observation of the object. 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.

[1]The work discussed in this chapter is based on the model proposed, but not implemented, by Broida in [9]. In this chapter, details of the implementation of this model are presented, the performance of the recursive algorithm is analyzed. This work was done before the matching techniques of Chapter 3 were developed, and hence feature point correspondence was done manually.

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 $\sim$10% 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.

Previous work by Broida and Chellappa in this area is discussed in [11, 12, 13, 15]. In [11] 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 favourable 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 [12]. A recursive solution was applied to simulated imagery involving pure translation and unknown structure. In [13], rotational motion was included, and a batch approach was applied to synthetic data: the experiments involved known object structure. In [15], 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.

In the present chapter, 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. Details of the models used are given in [9, 10].

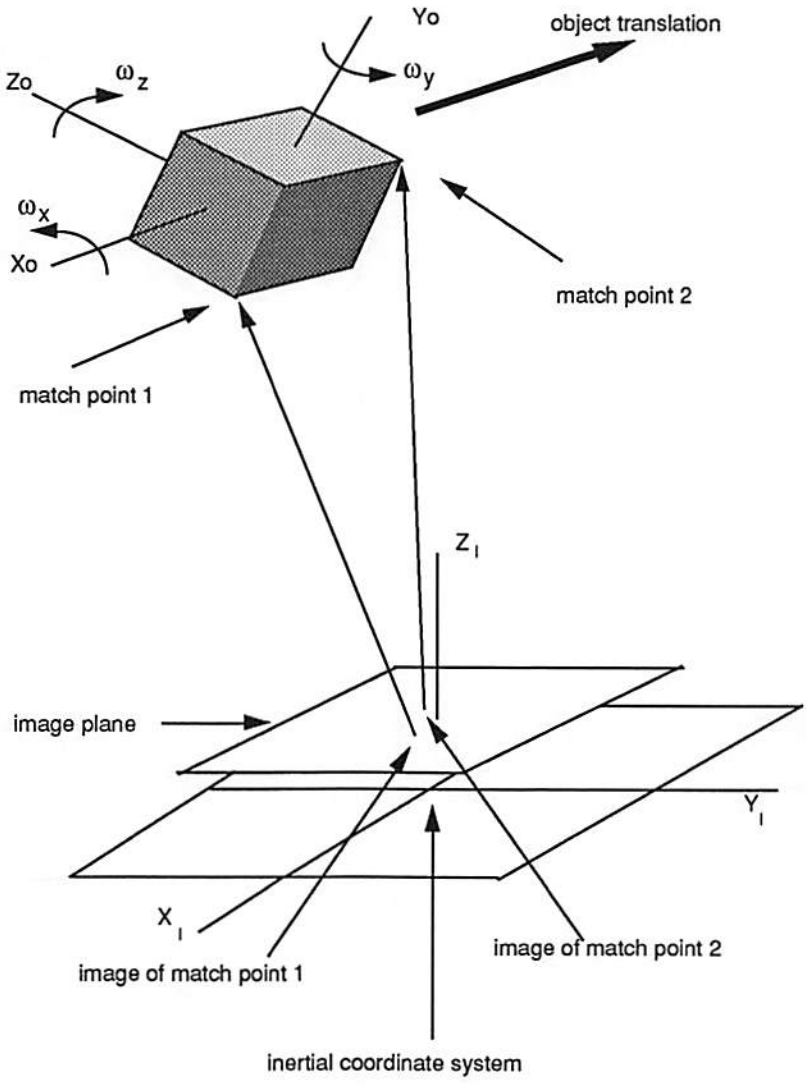

Figure 6.1: Models of motion and imaging used for object tracking

Here we summarize the recursive formulation, and discuss experimental results and implementational aspects.

6.1 Formulation for Recursive Solution

In this chapter, we deal 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.

$$\mathbf{s}(t) = \begin{pmatrix} x_R(t)/z_R(t) \\ y_R(t)/z_R(t) \\ \dot{x}/z_R(t) \\ \dot{y}/z_R(t) \\ \dot{z}/z_R(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \\ \omega_x \\ \omega_y \\ \omega_z \\ x_1/z_R(t) \\ y_1/z_R(t) \\ z_1/z_R(t) \\ \vdots \\ x_M/z_R(t) \\ y_M/z_R(t) \\ z_M/z_R(t) \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \\ s_{10} \\ s_{11} \\ s_{12} \\ s_{13} \\ s_{14} \\ s_{15} \\ \vdots \\ s_{3M+10} \\ s_{3M+11} \\ s_{3M+12} \end{pmatrix}_t \tag{6.1}$$

The origin of the object-centred coordinate system is expressed as

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

and the angular displacement between the object and camera coordinate systems are represented by the quaternions $(q_1(t), q_2(t), q_3(t), q_4(t))$. The scalar terms x_i, y_i, z_i are the coordinates of the i^{th} feature point in the object-centered system. The translational velocities are $(\dot{x}, \dot{y}, \dot{z})$ and the angular velocities are $(\omega_x, \omega_y, \omega_z)$. The The time derivative of $\mathbf{s}(t)$ is

$$\dot{\mathbf{s}}(t) = \begin{pmatrix} \dot{x}/z_R(t) - [x_R(t)/z_R(t)][\dot{z}/z_R(t)] \\ \dot{y}/z_R(t) - [y_R(t)/z_R(t)][\dot{z}/z_R(t)] \\ -[\dot{x}/z_R(t)][\dot{z}/z_R(t)] \\ -[\dot{y}/z_R(t)][\dot{z}/z_R(t)] \\ -[\dot{z}/z_R(t)]^2 \\ 0.5(\omega_z q_2 - \omega_y q_3 + \omega_x q_4) \\ 0.5(-\omega_z q_1 + \omega_x q_3 + \omega_y q_4) \\ 0.5(\omega_y q_1 - \omega_x q_2 + \omega_z q_4) \\ 0.5(-\omega_x q_1 - \omega_y q_2 - \omega_z q_3) \\ 0 \\ 0 \\ 0 \\ 0 \\ -[x_1/z_R(t)][\dot{z}/z_R(t)] \\ -[y_1/z_R(t)][\dot{z}/z_R(t)] \\ -[z_1/z_R(t)][\dot{z}/z_R(t)] \\ \vdots \\ -[x_M/z_R(t)][\dot{z}/z_R(t)] \\ -[y_M/z_R(t)][\dot{z}/z_R(t)] \\ -[z_M/z_R(t)][\dot{z}/z_R(t)] \end{pmatrix} = \begin{pmatrix} s_3 - s_1 s_5 \\ s_4 - s_2 s_5 \\ -s_3 s_5 \\ -s_4 s_5 \\ -s_5^2 \\ 0.5(s_{12} s_7 - s_{11} s_8 + s_{10} s_9) \\ 0.5(-s_{12} s_6 + s_{10} s_8 + s_{11} s_9) \\ 0.5(s_{11} s_6 - s_{10} s_7 + s_{12} s_9) \\ 0.5(-s_{10} s_6 - s_{11} s_7 - s_{12} s_8) \\ 0 \\ 0 \\ 0 \\ 0 \\ -s_{13} s_5 \\ -s_{14} s_5 \\ -s_{15} s_5 \\ \vdots \\ -s_{3M+10} s_5 \\ -s_{3M+11} s_5 \\ -s_{3M+12} s_5 \end{pmatrix}_t \tag{6.2}$$

The vector-valued measurement function ($2M \times 1$) is

$$\mathbf{p}(t_k) = \mathbf{h}\,[\mathbf{s}(t_k)]+\mathbf{v}(t_k) = \left(\begin{array}{c} \frac{s_1(t_k) + R_x(\mathbf{s},1,t_k)}{1 + R_z(\mathbf{s},1,t_k)} \\ \frac{s_2(t_k) + R_y(\mathbf{s},1,t_k)}{1 + R_z(\mathbf{s},1,t_k)} \\ \frac{s_1(t_k) + R_x(\mathbf{s},2,t_k)}{1 + R_z(\mathbf{s},2,t_k)} \\ \frac{s_2(t_k) + R_y(\mathbf{s},2,t_k)}{1 + R_z(\mathbf{s},2,t_k)} \\ \vdots \\ \frac{s_1(t_k) + R_x(\mathbf{s},M,t_k)}{1 + R_z(\mathbf{s},M,t_k)} \\ \frac{s_2(t_k) + R_y(\mathbf{s},M,t_k)}{1 + R_z(\mathbf{s},M,t_k)} \end{array} \right)_{t_k} + \mathbf{v}(t_k) = \left(\begin{array}{c} X_1(t_k) \\ Y_1(t_k) \\ X_2(t_k) \\ Y_2(t_k) \\ \vdots \\ X_M(t_k) \\ Y_M(t_k) \end{array} \right) + \mathbf{v}(t_k). \tag{6.3}$$

The abbreviations represent components of matrix-vector products, for example the scalar term $R_x(\mathbf{s},i,t_k)$ 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, $R(\mathbf{s},t_k) \cdot (s_{3i+10}, s_{3i+11}, s_{3i+12})^T$. For example, denoting the rs component of $R(\mathbf{s},t_k)$ as R_{rs},

$$\begin{aligned} R_x(\mathbf{s},1,t_k) &= R_{11}\,s_{13} + R_{12}\,s_{14} + R_{13}\,s_{15} \\ &= (q_1^2 - q_2^2 - q_3^2 + q_4^2)\,x_1/z_R(t) + 2(q_1q_2 - q_3q_4)\,y_1/z_R(t) \\ &\qquad + 2(q_1q_3 + q_2q_4)\,z_1/z_R(t). \end{aligned} \tag{6.4}$$

This problem formulation is appropriate for solution with a filter such as an EKF. The measurement function $\mathbf{h}[\cdot]$ (ratios of functions of states) is highly nonlinear, which requires the iteration of the measurement function. Hence an IEKF is used for estimation. The relevant equations are given in Appendix A.

In order to implement an IEKF, we need to obtain expressions for the linearized plant and measurement functions. With the state vector $\mathbf{s}$ as in

(6.1), $\mathbf{h}[\mathbf{s}]$ as in (6.3), the linearized measurement equation is given by

$$H(\mathbf{s}) \triangleq \frac{\partial \mathbf{h}[\mathbf{s}]}{\partial \mathbf{s}} = H(\mathbf{s}) = H^s(k) = \begin{pmatrix} P_1 & 0 & W_1 & 0 & S_1 & 0 & 0 & \cdots & 0 \\ P_2 & 0 & W_2 & 0 & 0 & S_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_M & 0 & W_M & 0 & 0 & 0 & 0 & \cdots & S_M \end{pmatrix} \tag{6.5}$$

where

$$P_i = \begin{pmatrix} \frac{1}{1+R_z(i)} & 0 \\ 0 & \frac{1}{1+R_z(i)} \end{pmatrix}; 1 \leq i \leq M, \tag{6.6}$$

the W_i submatrices are of dimension 2×4 with elements

$$W_i(1,k) = \frac{(1+R_z(i))\frac{\partial R_x(i)}{\partial q_k} - (s_1 + R_x(i))\frac{\partial R_z(i)}{\partial q_k}}{(1+R_z(i))^2}; 1 \leq k \leq 4, \tag{6.7}$$

$$W_i(2,k) = \frac{(1+R_z(i))\frac{\partial R_y(i)}{\partial q_k} - (s_2 + R_y(i))\frac{\partial R_z(i)}{\partial q_k}}{(1+R_z(i))^2}; 1 \leq k \leq 4, \tag{6.8}$$

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

$$S_i(1,1) = \frac{(1+R_z(i))R_{11} - (s_1 + R_x(i))R_{31}}{(1+R_z(i))^2} \tag{6.9}$$

$$S_i(1,2) = \frac{(1+R_z(i))R_{12} - (s_1 + R_x(i))R_{32}}{(1+R_z(i))^2} \tag{6.10}$$

$$S_i(1,3) = \frac{(1+R_z(i))R_{13} - (s_1 + R_x(i))R_{33}}{(1+R_z(i))^2} \tag{6.11}$$

$$S_i(2,1) = \frac{(1+R_z(i))R_{21} - (s_2 + R_x(i))R_{31}}{(1+R_z(i))^2} \tag{6.12}$$

$$S_i(2,2) = \frac{(1+R_z(i))R_{22} - (s_2 + R_x(i))R_{32}}{(1+R_z(i))^2} \tag{6.13}$$

$$S_i(2,3) = \frac{(1+R_z(i))R_{23} - (s_2 + R_x(i))R_{33}}{(1+R_z(i))^2} \tag{6.14}$$

The linearized plant function is given by

$$F(\mathbf{s}) = \frac{\partial \mathbf{f}(\mathbf{s})}{\partial \mathbf{s}} \begin{pmatrix} F_1 & 0 & 0 \\ 0 & F_2 & 0 \\ F_3 & 0 & F_4 \end{pmatrix} \tag{6.15}$$

where

$$F_1 = \begin{pmatrix} -s_5 & 0 & 1 & 0 & -s_1 \\ 0 & -s_5 & 0 & 1 & -s_2 \\ 0 & 0 & -s_5 & 0 & -s_3 \\ 0 & 0 & 0 & -s_5 & -s_4 \\ 0 & 0 & 0 & 0 & -2s_5 \end{pmatrix} \tag{6.16}$$

$$F_2 = 0.5 \begin{pmatrix} 0 & s_{12} & -s_{11} & s_{10} & s_9 & -s_8 & s_7 \\ -s_{12} & 0 & s_{10} & s_{11} & s_8 & s_9 & -s_6 \\ s_{11} & -s_{10} & 0 & s_{12} & -s_7 & s_6 & s_9 \\ -s_{10} & -s_{11} & -s_{12} & 0 & -s_6 & -s_7 & -s_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.17}$$

$$F_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & -s_{13} \\ 0 & 0 & 0 & 0 & -s_{14} \\ 0 & 0 & 0 & 0 & -s_{15} \\ 0 & 0 & 0 & 0 & -s_{16} \\ 0 & 0 & 0 & 0 & -s_{17} \\ 0 & 0 & 0 & 0 & -s_{18} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & -s_{3M+12} \end{pmatrix} \tag{6.18}$$

$$F_4 = \begin{pmatrix} -s_5 & 0 & 0 & 0 & \cdots & 0 \\ 0 & -s_5 & 0 & 0 & \cdots & 0 \\ 0 & 0 & -s_5 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & & -s_5 \end{pmatrix} \tag{6.19}$$

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 [6], 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 [69]. Unfortunately, Gibbs parameters do not evolve in time in a way conducive to simple modeling such as equations (4.4) and (4.5).

6.2 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 section describing the details of the implementation of the IEKF—such as parameter selection—and related numerical issues.

6.2.1 Experiments with Simulated Imagery

The object whose motion is to be studied is a rigid transparent cube of side four units. The axes and origin of the object-centred 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 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 the batch procedure described in [9].

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 centre of the object-centred coordinate-system. In the cases discussed, four 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 four 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. 6.2) A moderately low measurement noise level of 2.5% was used in this case. A crude initial state estimate (with errors of 20% 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 can also be used for this purpose.

Case 2: (Fig. 6.3) The measurement noise level here was fairly high (10%), 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 to 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. 6.4) 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 the batch algorithm).

Case 4: (Fig. 6.5) In this case all the initial values of the state vector were set to zero (except the quaternions, which can be trivially initialized if it is assumed that the object-centred 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 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 (as low as 2.5%) were added to the measurements.

6.2.2 Experiments with Real Imagery

This experiment involves randomly selected points on the side of the tyre of a car approaching the camera (Fig. 6.6). 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 inches between each frame, corresponding to a tyre rotation of about 14.8 degrees. 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 component due to the positioning of the camera. The object image size (i.e. the size of the tyre) is about 2 inches at the start of the sequence, and about 3 inches at the end. The total rotation was about 4 radians, and the total translation about 45 inches. The photographs were digitized to a resolution of 50 pixels/inch. 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

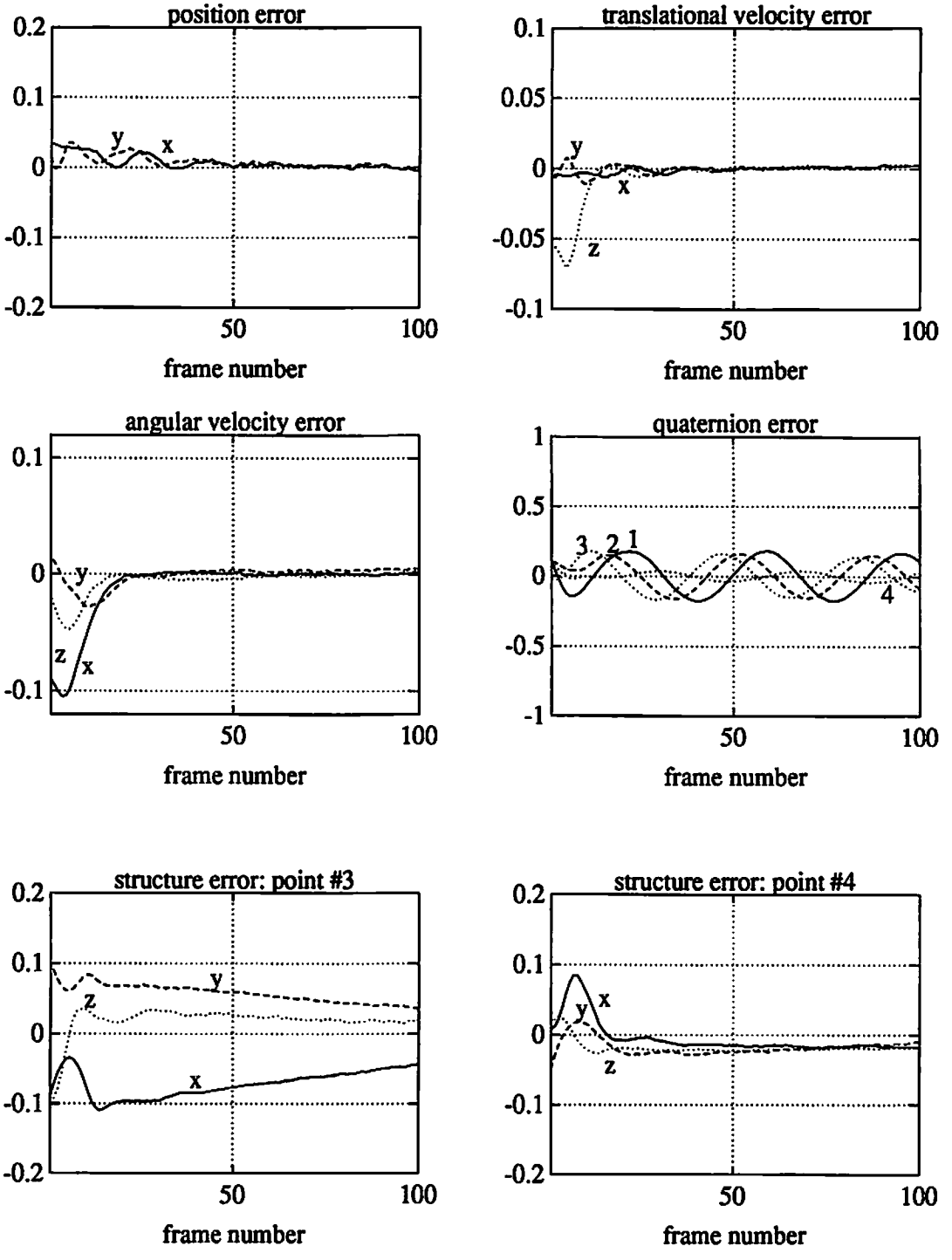

Figure 6.2: Estimation errors: Case 1

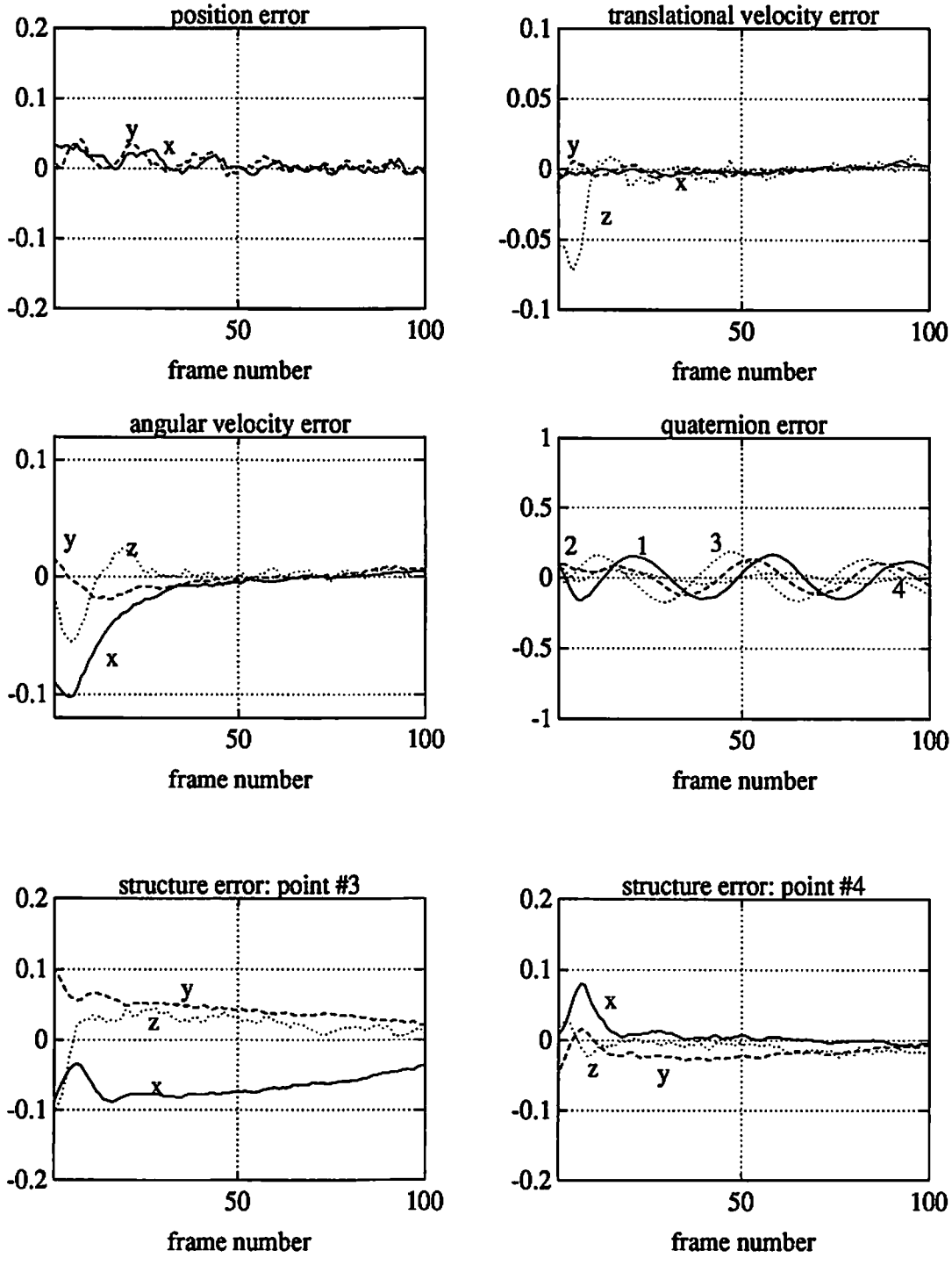

Figure 6.3: Estimation errors: Case 2

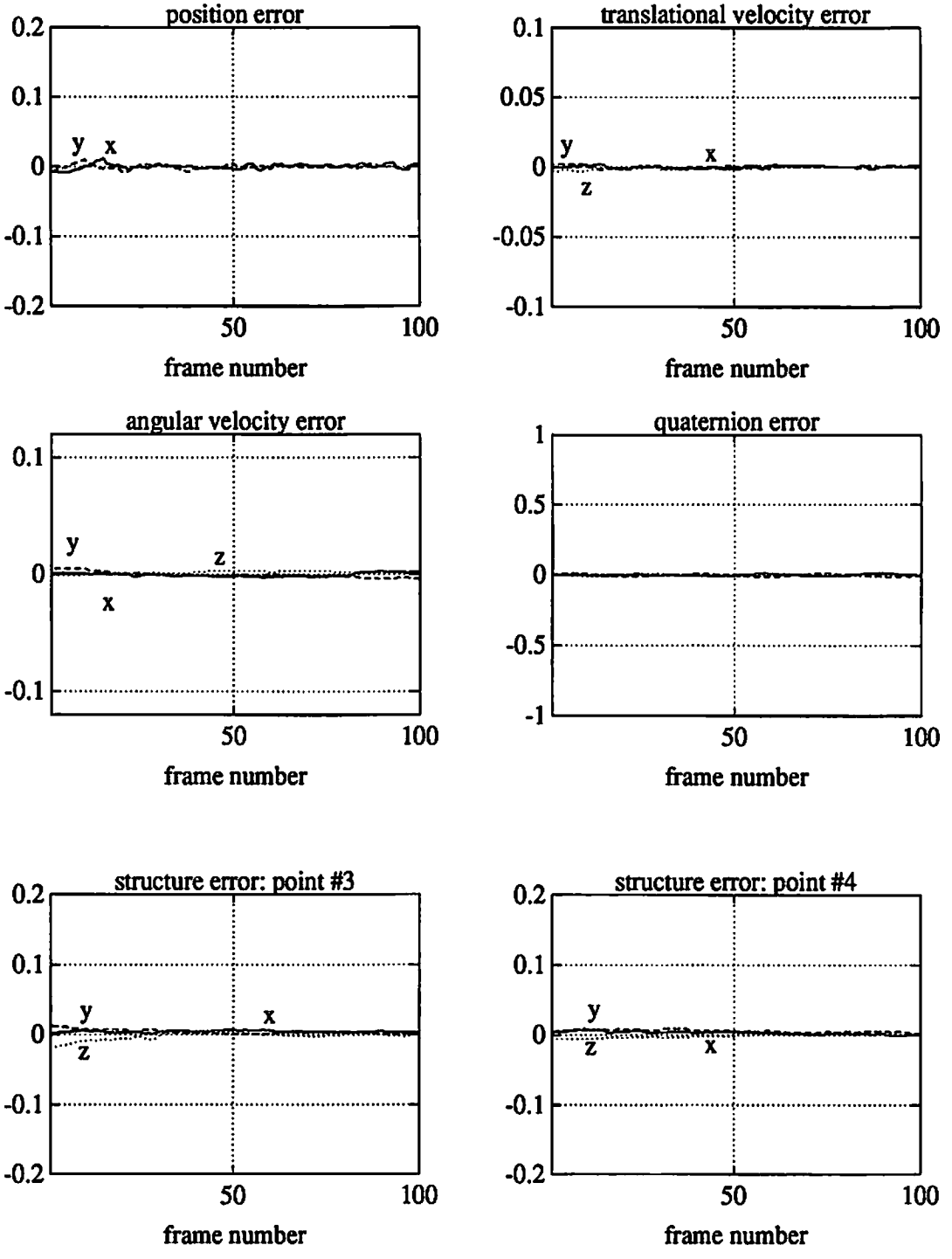

Figure 6.4: Estimation errors: Case 3

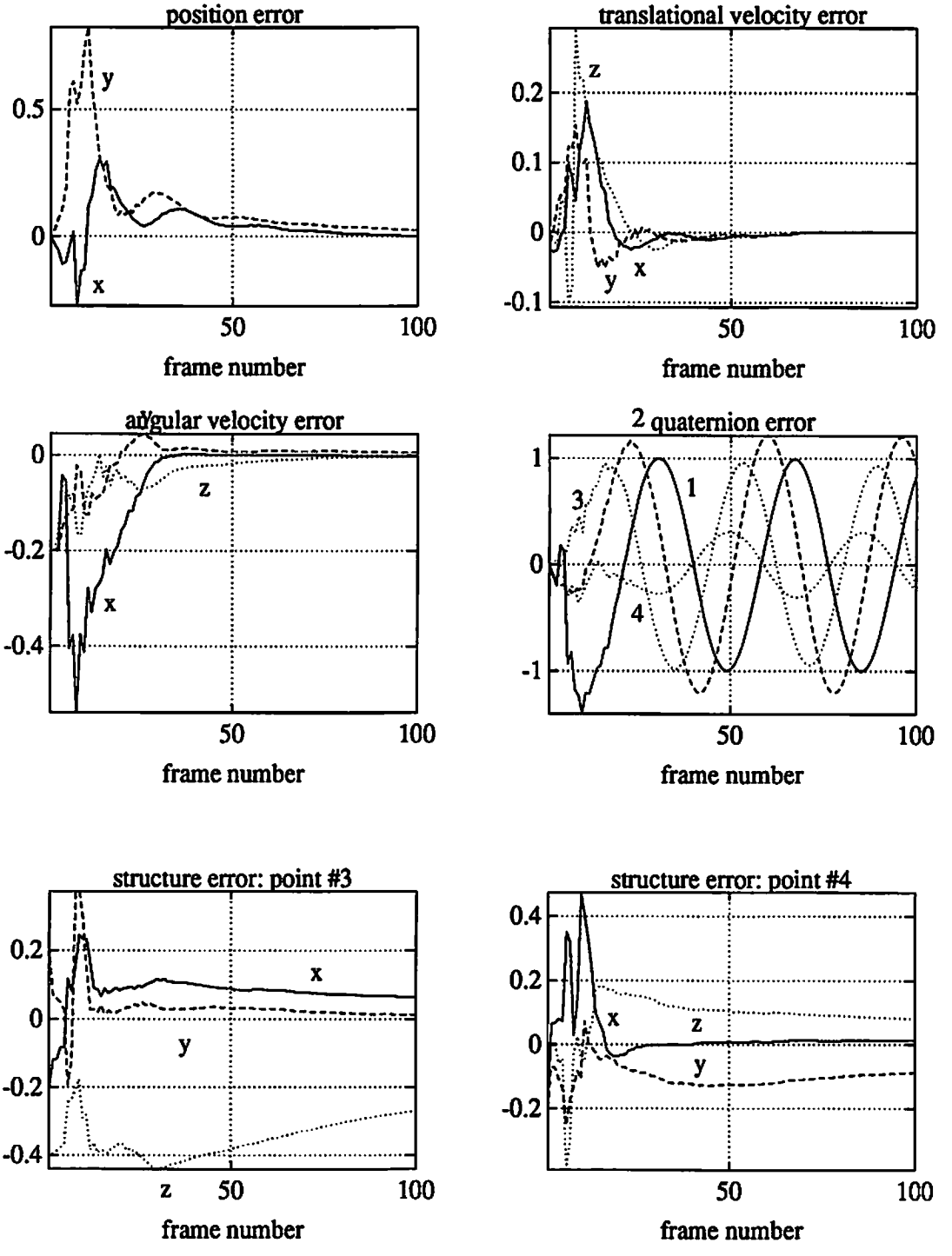

Figure 6.5: Estimation errors: Case 4

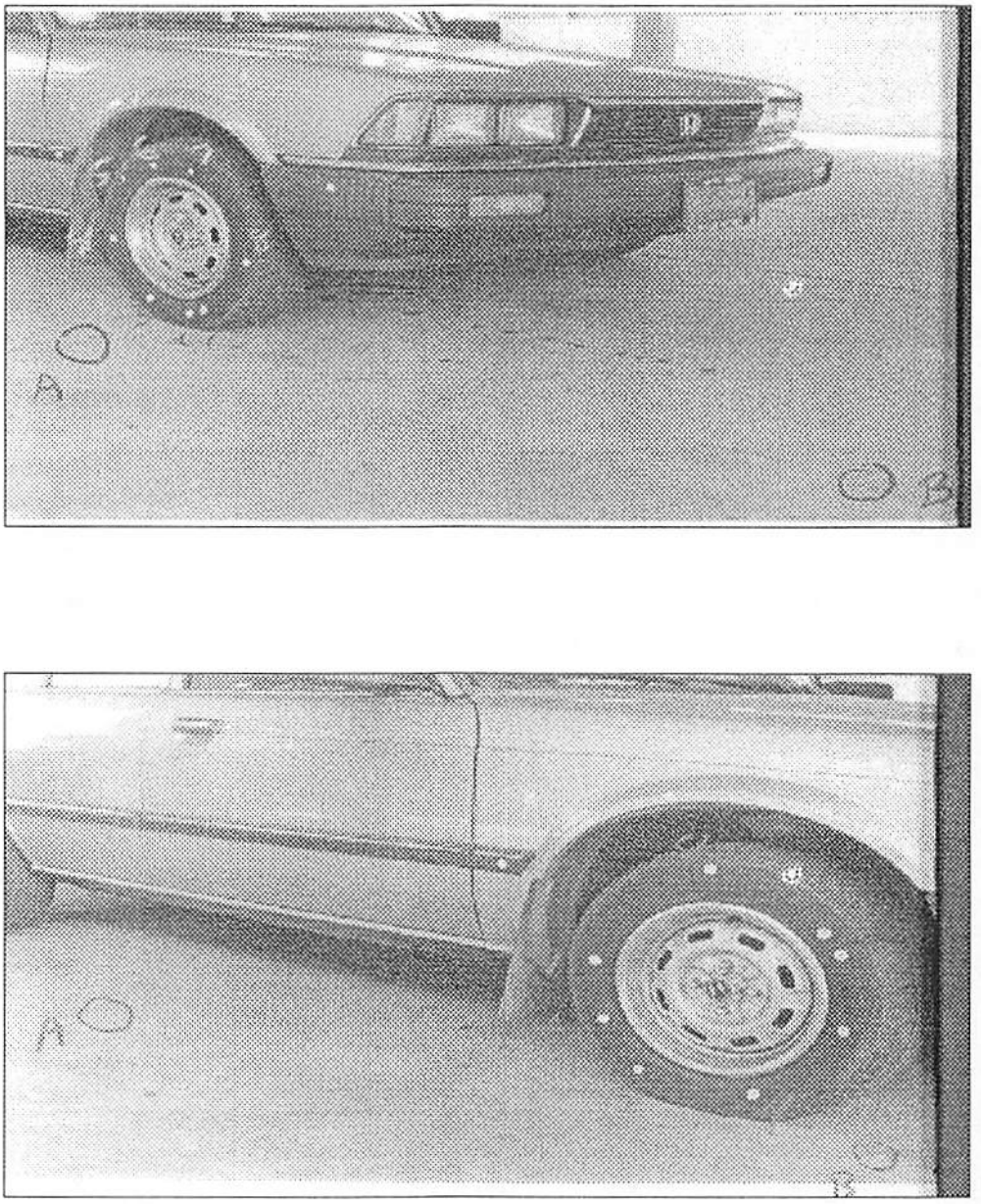

Figure 6.6: First and last frames of the real image sequence

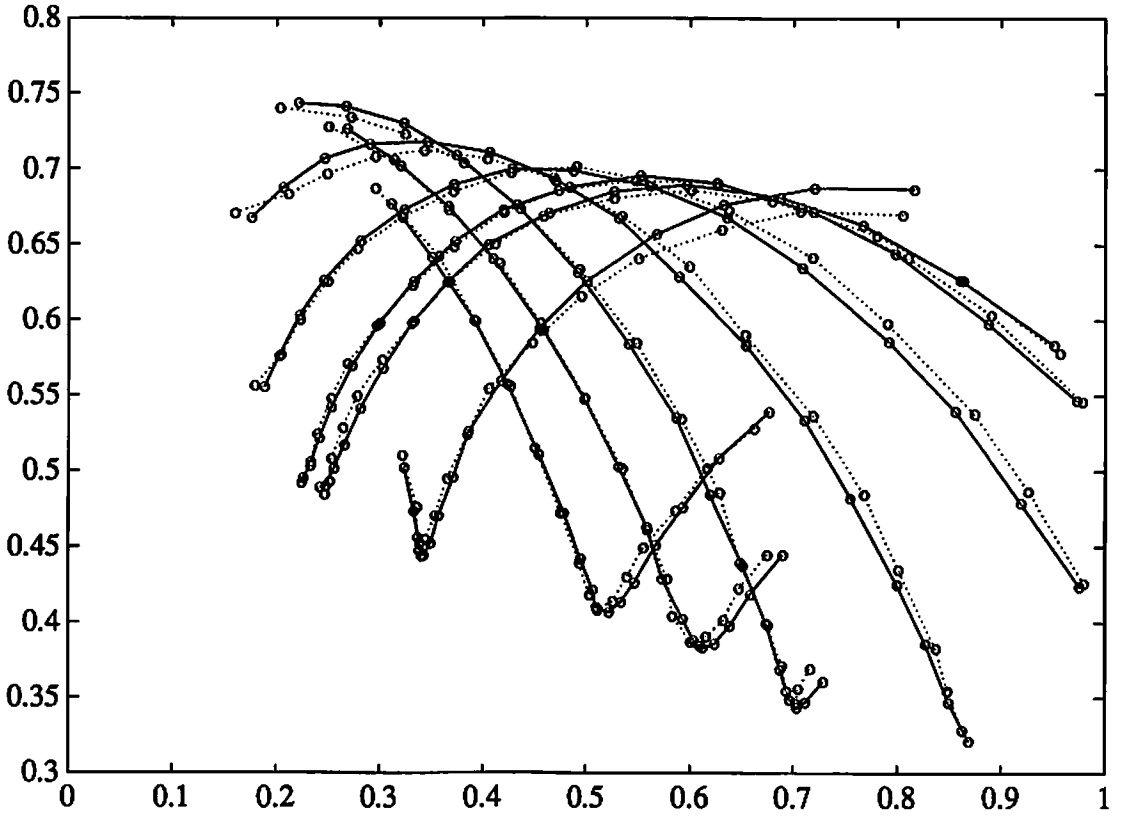

Figure 6.7: Actual and estimated image point trajectories for car sequence

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 anyway 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. 6.7. 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. 6.7 seems to confirm that the IEKF is doing a reasonably good job of tracking the moving object.

6.3 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 $\hat{\mathbf{s}}(0)$
2. Initial error covariance $P(0)$
3. Plant noise covariance matrices Q_k
4. Measurement (observation) noise covariances matrices R_k .

A reliable way to obtain a good initial estimate is to run the batch estimation algorithm on the first few frames. For the particular motion parameters chosen in the simulations, the batch algorithm required about 250 iterations to converge. For Cases 1 & 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.

If a sufficiently accurate batch solution is available, approximate Cramer-Rao lower bounds (CRLBs) can be found for the error in the initial guess i.e. the initial error covariance [14]. 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 behaviour. 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

$$K(k)_{n+1} = \hat{P}(k|k-1)H(k)_n^T \left[H(k)_n \hat{P}(k|k-1)H(k)_n^T + R_k\right]^{-1} \qquad (6.20)$$

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 (A.21) 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. 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 [50]. 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 (6.20). The H matrix in the equation is very sparse, and the pre- and post-multiplication 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 (6.20) 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 (6.20). 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. 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 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, other numerically stable approaches such as the square root information filter should be examined.

Chapter 7

Obstacle Avoidance

The analysis of traffic scenes is the subject of intense research as a part of the European project Prometheus. The scenes are viewed from a land vehicle (henceforth to be referred to as the "observer") which could be stationary or in motion, in an environment which contains one or more stationary or independently moving objects (referred to as "obstacles"). One of the basic objectives here is to detect and understand the relative motion (with respect to the observer) of the various obstacles in its vicinity. This goal has to be accomplished primarily with the help of the image sequences provided by one or more cameras in the observer.

In this chapter, it is assumed that the observer is equipped with two cameras, which provide stereo image pairs at a sufficiently high frame rate to enable the detection and analysis of the fastest-moving obstacle which may be encountered in a realistic situation. The theory of stereo vision has been extensively studied [49, 51] and several methods have been developed for the processing of stereo image pairs to obtain 3-D information about a scene (for instance [36, 46, 53]). We are, however, constrained by the need for real-time processing, and the need to handle images of complex and diverse objects. The algorithm discussed in [55], which matches contour chain points in a pyramidal framework, is ideal for our purpose, since it is fast and deals equally well with

natural and man-made objects, however complex their shape. We use this algorithm both for stereo matching as well as for the determination of optic flow [54], using contour chain points as primitives.

The outputs of the stereo and optic flow algorithms can be combined to determine the (noisy) 3-D trajectories of the contour chain points over an arbitrarily long image sequence. These noisy trajectories can then be filtered, either recursively or in a batch framework to determine the filtered 3-D positions and the parameters of the 3-D motion of selected points at any given time instant. The motion parameters may be computed to any order of complexity, but a second-order model (assuming constant acceleration) is usually sufficient for most purposes.

In order to perform the filtering, it is necessary to know the accuracy of the 3-D information provided by the stereo algorithm. To be more specific, it is necessary to know the mean and covariance of the 3-D position error[1] given a measurement of the 3-D position of a point in the scene (as computed by the stereo algorithm). Many methods exist in the literature to analyse the effects of quantization error on the accuracy of stereo triangulation (e.g.[59, 4, 8]). However, most of these methods do not result in explicit expressions for the statistics of the 3-D error, as is required for trajectory estimation. In this chapter, we present a simple method of obtaining the mean and covariance of the position error from the 3-D position vector obtained using stereo triangulation.

7.1 Formulation of the Problem

The basic problem environment consists of a land vehicle (the observer) surrounded by various objects (the obstacles), any one or more of which could be moving independently. In this work, no knowledge about the motion of the observer is known, although it will be shown later how such information may be obtained from the output of the trajectory estimation algorithms. In such a situation it is necessary to decide the coordinate system in which the data will

[1]Strictly speaking, one needs to know the complete probability density function of the error, but for most practical purposes a knowledge of its mean and covariance is sufficient.

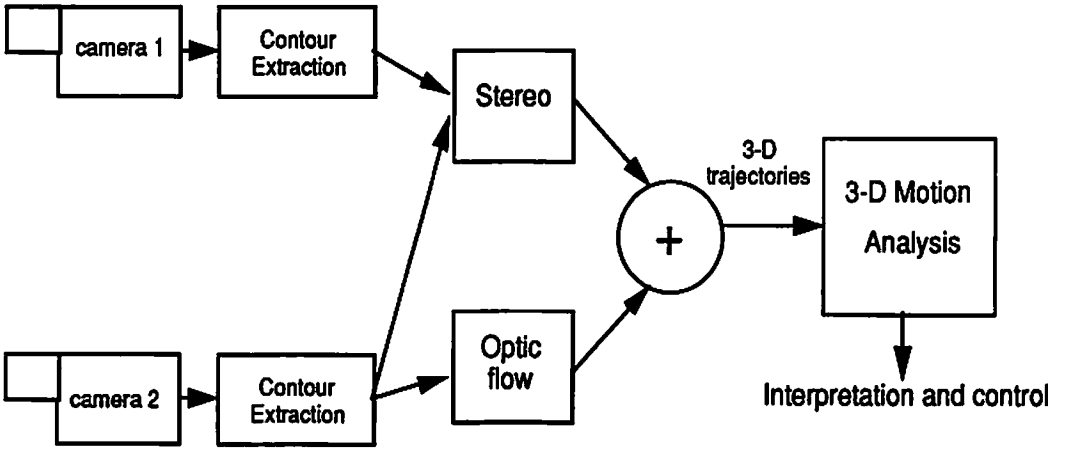

Figure 7.1: Schematic diagram of the temporal analysis

be represented. One could represent all the 3-D information in the coordinate system of the observer (OCS), or in a (hypothetical) world coordinate system (WCS). It is also possible to imagine a coordinate system attached to each independently moving obstacle. Since the goal is to analyse the movement of objects with respect to the observer, it seems reasonable to represent data in the observer's coordinate system.

It is also necessary to decide the primitives to be used in the analysis—points, lines, curves or entire objects. An object-based formulation [10, 60, 66] would require complex (possibly nonlinear) motion models, as well as segmentation of the 3-D data before the temporal analysis. Even if the motion of the observer and the obstacles can be modelled in a simple manner, the relative motion of an obstacle (as a whole) with respect to the observer may not be easy to model. Therefore, object-based models should be used only when more information (such as segmentation or observer motion) is available. A line-based approach [19, 42, 65, 75] would require fewer preconditions to be met, but would in general work well only for scenes composed of man-made

objects with polygonal surfaces. A method based on curves [39, 28] would be applicable to a wider variety of scenes. However, curves are hard to model and manipulate, and present several mathematical and practical difficulties. Points, then, are the obvious choice of primitives for our problem. Furthermore, if we restrict ourselves to points lying on contour chains, there are other advantages [55].

In order to estimate the trajectory of 3-D scene point in time, it is useful to make some assumptions about its motion. The fundamental assumption we make is that the trajectory is "smooth" i.e. it can be represented by a small number of parameters, and that this number is not dependent on the number of image frames over which the motion is analysed. To be precise, it is assumed that the motion of a point $p(t)$ in the WCS can be modelled as:

$$p(t) = p(0) + \dot{p}(0)\ t + \ddot{p}(0)\ t^2/2! + \cdots + p^{(n)}(0)\ t^n/n!, \tag{7.1}$$

where where n is small compared to the number of frames in the sequence. (In other words, it is assumed that derivatives of $p(t)$ higher than the nth are negligible for all t.) Typically, one would select $n = 1$ or $n = 2$.

Let us now consider the motion in the WCS of two points, one (p_o) on a moving obstacle, and the other (p_c) being the origin of the coordinate system of the observer. Using the model in (7.1), their trajectories in the WCS can be written in the form

$$p_o(t) = p_o(0) + \dot{p}_o(0)\ t + \ddot{p}_o(0)\ t^2/2! + \cdots + p_o^{(n)}(0)\ t^n/n! \tag{7.2}$$

and

$$p_c(t) = p_c(0) + \dot{p}_c(0)\ t + \ddot{p}_c(0)\ t^2/2! + \cdots + p_c^{(n)}(0)\ t^n/n!. \tag{7.3}$$

Since we are interested in the relative trajectories of points in the scene with respect to the observer, we have to express (7.2) in the OCS. If the OCS is assumed to be aligned with the WCS, this can be done simply by subtracting from (7.2) the trajectory of the origin of the OCS, given by (7.3), as follows:

$$p_o(t) - p_c(t) \ = \ p_o(0) - p_c(0) + (\dot{p}_o(0) - \dot{p}_c(0))\ t \tag{7.4}$$

$$+(\ddot{p}_o(0) - \ddot{p}_c(0))\ t^2/2! + \cdots + (p_o^{(n)}(0) - p_c^{(n)}(0))\ t^n/n!$$

defining

$$p(t) \triangleq p_o(t) - p_c(t), \tag{7.5}$$

we can reduce (7.4) to the form (7.1), thereby establishing the validity of the model in (7.1) even for the trajectories determined in the OCS. Equation (7.1) will henceforth be the basic motion model used in the estimation of point trajectories relative to the observer.

7.2 The Stereo Algorithm

The stereo algorithm used in this work has been developed as a part of the Prometheus project referred to earlier. In this section, we present its salient features. A more detailed description is given in [55]. The basic issues of interest are the primitives used for matching, the matching criteria and the organization of the data.

As mentioned earlier, due to the complex nature of the images encountered, contour chain points are the obvious choice of matching primitives. The extraction of contour chain points is performed in three steps: gradient computation [22], hysteresis thresholding to eliminate noise, and the chaining of contour points [35]. The contour chains serve two purposes: they are used to propagate disparities, and they help in reducing false targets. They provide a rich 3-D description of the environment, greatly facilitating the further interpretation of the scenes.

Stereo matching is performed by optimizing a similarity function. For two contour points, (x_l, y_l) in the left image and (x_r, y_r) in the right image the similarity function is defined as:

$$f(x_l, y_l, x_r, y_r) = \frac{[G(x_l, y_l) - G(x_r, y_r)]^2}{S_G^2} + \frac{[\theta(x_l, y_l) - \theta(x_r, y_r)]^2}{S_\theta^2}$$

where $G(x, y)$ and $\theta(x, y)$ are, respectively, the gradient norm and the orientation at the point (x, y); and S_G and S_θ are respectively the thresholds on the

gradient norm and the difference in orientation .

For all (x_i, y_i) in the left image (x_j, y_j) is a potential match in the right image if the following conditions are satisfied.

1. $|G(x_{li}, y_{li}) - G(x_{rj}, y_{rj})| \leq S_G$
2. $|\theta(x_{li}, y_{li}) - \theta(x_{rj}, y_{rj})| \leq S_\theta$
3. $f(x_{li}, y_{li}, x_{rj}, y_{rj})$ minimum w.r.t. j

If a point (x_{rjo}, y_{rjo}) in the right image satisfies the above criteria, then the correspondence between the pair $((x_{li}, y_{li}), (x_{rjo}, y_{rjo}))$ is validated if $f(x_{li}, y_{li}, x_{rjo}, y_{rjo})$ is minimal w.r.t. i. The matching process is then symmetric, and uniqueness is guaranteed.

Although the above criteria are fairly stringent, matching based on similarity alone is prone to error. Ambiguities in matching often occur, and suitable criteria have to be selected for validation of matches. The so-called second stereo law, which basically requires local continuity of disparities, is then applied. The consistency of matches obtained using the similarity criteria is tested, using contour chains as local support—matches which do not have supporting local evidence are rejected. For each pair of matched points the disparity vector (consisting of its norm L, and its orientation β) is determined. The vectors are then checked for continuity along the chain. A matching pair of points i is validated if:

$$\sum_{j \in V_i} \frac{\frac{|L_i - L_j|}{(L_i + L_j)} \frac{1}{|i-j|^n}}{\sum_{j \in V_i} \frac{1}{|i-j|^n}} \leq S_L$$

and

$$\sum_{j \in V_i} \frac{\frac{|\beta_i - \beta_j|}{(\beta_i + \beta_j)} \frac{1}{|i-j|^n}}{\sum_{j \in V_i} \frac{1}{|i-j|^n}} \leq S_\beta$$

where V_i is a local neighborhood of the pair i along the chain, S_L and S_β are thresholds and n defines the neighborhood size.

A four-level pyramidal structure is used for data representation. A coarse-to-fine strategy is used for matching, starting with the level of the pyramid corresponding to the coarsest resolution. The disparity information obtained at a particular resolution is used to restrict the search space for matching at the next higher resolution, all the way up to the highest resolution. This approach leads to greater speed of processing, and also helps to limit the number of false matches.

7.3 Stereo Error Analysis

In analyzing the errors in the 3-D position $\hat{p}$ of a point calculated by a stereo algorithm (using the corresponding 2-D measurements from the stereo pair), our goal is to obtain an expression for the error Δp, and to determine its mean μ and covariance C. We are not interested in the "absolute" statistics of the error, such as those calculated in [59], but rather in the mean and covariance of the error *given the true 3-D position.* The error is assumed to be entirely due to the quantization of the image plane, and other factors such as stereo mismatches and calibration errors are not considered in this chapter. As will be seen later, it is simpler to calculate the statisics of the error using spherical rather than Cartesian coordinates. The error is defined by

$$\Delta p = \hat{p} - p \tag{7.6}$$

where p is the (unknown) true 3-D position, and $\hat{p}$ is its stereo estimate. The mean and covariance of the error are defined by

$$\mu = \mathcal{E}\{\Delta p\} \tag{7.7}$$

and

$$C = \mathcal{E}\left\{(\Delta p)(\Delta p)^T\right\} - \mu\mu^T. \tag{7.8}$$

If (x_L, y_L) and (x_R, y_R) are the (true) image coordinates of the point p in the left and right image, respectively, the basic equations of the stereo algorithm

are as follows:

$$d = x_L - x_R \tag{7.9}$$

$$x = \frac{bx_L}{d} \tag{7.10}$$

$$y = \frac{by_L}{d} \tag{7.11}$$

$$z = \frac{bf}{d} \tag{7.12}$$

where d is the stereo disparity, and b and f are, respectively, the baseline and focal length of the imaging system.

The measurements of the image coordinates x_L, y_L, etc. are corrupted by quantization noise, which causes errors in the quantities computed using (7.9–7.12). Denoting the actual measurements by $\hat{x_L}$, $\hat{y_L}$ etc., and the errors by Δd, Δx etc. we obtain

$$\begin{aligned} \Delta d &= \hat{d} - d \\ &= (\hat{x_L} - \hat{x_R}) - (x_L - x_R) \\ &= \Delta x_L - \Delta x_R \end{aligned} \tag{7.13}$$

If the grid size in the x-direction is ρ_x, the random variables Δx_L and Δx_R will be uniformly distributed in $[-\rho_x/2, \rho_x/2]$; and if they are assumed to be independent, the p.d.f. of Δd can be obtained as

$$p(\Delta d) = \begin{cases} -\frac{1}{\rho_x^2}|\Delta d| + \frac{1}{\rho_x} & |\Delta d| \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{7.14}$$

The error in z is given by:

$$\begin{aligned} \Delta z &= \hat{z} - z \\ &= \frac{bf}{d + \Delta d} - \frac{bf}{d} \\ &= bf\left(\frac{-\Delta d}{d(d + \Delta d)}\right) \end{aligned}$$

$$= -\frac{\Delta d}{d+\Delta d}\,\frac{bf}{d}$$
$$= -\frac{\Delta d}{d+\Delta d}\,z \tag{7.15}$$

The error in x is given by:

$$\begin{aligned}
\Delta x &= \hat{x} - x \\
&= \frac{b\hat{x_L}}{d+\Delta d} - \frac{bx_L}{d} \\
&\approx \frac{bx_L}{d+\Delta d} - \frac{bx_L}{d} \quad \text{if } x_L \gg 0 \qquad (7.16)\\
&= bx_L\left(\frac{-\Delta d}{d(d+\Delta d)}\right) \\
&= -\frac{\Delta d}{d+\Delta d}\,\frac{bx_L}{d} \\
&= -\frac{\Delta d}{d+\Delta d}\,x \qquad (7.17)
\end{aligned}$$

Similarly, the error in y is given by

$$\begin{aligned}
\Delta y &= \hat{y} - y \\
&= \frac{b\hat{y_L}}{d+\Delta d} - \frac{by_L}{d} \\
&\approx \frac{by_L}{d+\Delta d} - \frac{by_L}{d} \qquad (7.18)\\
&= by_L\left(\frac{-\Delta d}{d(d+\Delta d)}\right) \\
&= -\frac{\Delta d}{d+\Delta d}\,\frac{by_L}{d} \\
&= -\frac{\Delta d}{d+\Delta d}\,y \qquad (7.19)
\end{aligned}$$

Combining the above results, we can write

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = -\frac{\Delta d}{d+\Delta d}\begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{7.20}$$

or

$$\Delta p = -\frac{\Delta d}{d + \Delta d}\, p. \tag{7.21}$$

The approximations (7.16) and (7.18) are needed to obtain simple expressions for Δx and Δy, as functions of only one random variable, i.e., Δd. But the resulting expressions give only the error in the radial direction, as is evident from (7.21). The other (lateral) components of the error have to be derived separately. It is convenient at this point to switch to a spherical coordinate system, computing the statistics of the error in the range r, the azimuth ϕ and the elevation θ. The error statistics in the Cartesian coordinates can then be obtained using the Jacobian of the Cartesian-to-spherical transformation.

The range r is defined by

$$\begin{aligned} r &= \|p\| \\ &= \sqrt{x^2 + y^2 + z^2} \\ &= \frac{1}{d}\sqrt{(bx_L)^2 + (by_L)^2 + (bf)^2}. \end{aligned} \tag{7.22}$$

The observed range is given by

$$\begin{aligned} \hat{r} &= \|\hat{p}\| \\ &= \sqrt{\hat{x}^2 + \hat{y}^2 + \hat{z}^2} \\ &= \frac{1}{d + \Delta d}\sqrt{(b\hat{x_L})^2 + (b\hat{y_L})^2 + (bf)^2} \\ &\approx \frac{1}{d + \Delta d}\sqrt{(bx_L)^2 + (by_L)^2 + (bf)^2}, \end{aligned} \tag{7.23}$$

using the same approximation as before. The range error is given by

$$\begin{aligned} \Delta r &= \hat{r} - r \\ &= \left[\frac{1}{d + \Delta d} - \frac{1}{d}\right] \sqrt{(bx_L)^2 + (by_L)^2 + (bf)^2} \\ &= -\left(\frac{\Delta d}{d(d + \Delta d)}\right) \sqrt{(bx_L)^2 + (by_L)^2 + (bf)^2} \\ &= -\left(\frac{\Delta d}{d + \Delta d}\right)\, r. \end{aligned} \tag{7.24}$$

It is interesting to compare the above result with (7.21).

It is now possible to derive expressions for the (conditional) mean μ_r and variance σ_r^2 of Δr, defined by

$$\mu_r \triangleq \mathcal{E}\{\Delta r \,/r\} \tag{7.25}$$

and

$$\sigma_r^2 \triangleq \mathcal{E}\left\{(\Delta r)^2 \,/r\right\} - \mu_r^2. \tag{7.26}$$

Using (7.24), we get

$$\begin{aligned}
\mu_r &= \mathcal{E}\left\{-\left(\frac{\Delta d}{d+\Delta d}\right) r \,/r\right\} \\
&= -r\mathcal{E}\left\{-\left(\frac{\Delta d}{d+\Delta d}\right) /r\right\} \\
&= -r\int_{-\rho_x}^{\rho_x}\left(\frac{u}{d+u}\right)\left(-\frac{1}{\rho_x^2}|u|+\frac{1}{\rho_x}\right) du,
\end{aligned} \tag{7.27}$$

using (7.14). The above integration can be performed, and the final result is

$$\mu_r = \left[1 - \frac{1}{a^2}\ln(1-a^2) - \frac{1}{a}\ln\left(\frac{1+a}{1-a}\right) - 1\right] r, \tag{7.28}$$

where

$$a = \frac{\rho_x}{d}. \tag{7.29}$$

The integral converges only if $\rho_x/d < 1$. If $\rho_x/d << 1$, the above expression simplifies to

$$\mu_r \approx \left(-\frac{a^2}{6}\right) r. \tag{7.30}$$

The derivation of σ_r^2 is similar. Since the mean of the range error is small, one may write

$$\begin{aligned}
\sigma_r^2 &\approx \mathcal{E}\left\{\left(\frac{\Delta d}{d+\Delta d}\right)^2 r^2 \,/r\right\} \\
&= r^2\mathcal{E}\left\{\left(\frac{\Delta d}{d+\Delta d}\right)^2 /r\right\}
\end{aligned}$$

$$= r^2 \int_{-\rho_x}^{\rho_x} \left(\frac{u}{d+u}\right)^2 \left(-\frac{1}{\rho_x^2}|u| + \frac{1}{\rho_x}\right) du, \tag{7.31}$$

which gives

$$\sigma_r^2 = \left[1 - \frac{3}{a^2}\ln(1-a^2) - \frac{2}{a}\ln\left(\frac{1+a}{1-a}\right)\right] r^2, \tag{7.32}$$

with the same definition of a and the same convergence criterion as before. For $a << 1$

$$\sigma_r^2 \approx \left(\frac{a^2}{6}\right) r^2. \tag{7.33}$$

The azimuth angle ϕ and the angle of elevation θ are defined by the following equations.

$$\tan\phi = \frac{x}{z} = \frac{x_L}{f} \tag{7.34}$$

$$\tan\theta = \frac{y}{\sqrt{x^2+z^2}} = \frac{y_L}{\sqrt{x_L^2+f^2}} \tag{7.35}$$

The measurement error in the azimuth angle is given by

$$\begin{aligned}
\Delta\phi &\triangleq \hat{\phi} - \phi \\
&\approx \tan(\hat{\phi} - \phi) \\
&= \frac{\tan\hat{\phi} - \tan\phi}{1 + \tan\hat{\phi}\tan\phi} \\
&\approx \frac{(\hat{x_L} - x_L)/f}{1 + \tan^2\phi} \\
&= \frac{\Delta x_L}{f(1+\tan^2\phi)}
\end{aligned} \tag{7.36}$$

Since the quantization error Δx_L is assumed to be uniformly distributed in $[-\rho_x/2, \rho_x/2]$, we can write the p.d.f. of $\Delta\phi$ as

$$\Delta\phi \sim \mathcal{U}\left(-\frac{\rho_x}{2f(1+\tan^2\phi)}\ ,\ \frac{\rho_x}{2f(1+\tan^2\phi)}\right) \tag{7.37}$$

From the above, we can obtain the mean μ_ϕ and variance σ_ϕ^2 as

$$\mu_\phi = 0 \tag{7.38}$$

and

$$\sigma_\phi^2 = \frac{1}{f^2(1+\tan^2\phi)^2}\frac{\rho_x^2}{12} \tag{7.39}$$

In a similar manner, we obtain the following expressions for the statistics of the error in θ:

$$\mu_\theta = 0 \tag{7.40}$$

and

$$\sigma_\theta^2 = \frac{1}{(f^2+x_L^2)(1+\tan^2\theta)^2}\frac{\rho_y^2}{12} \tag{7.41}$$

The results obtained for r, θ and ϕ can be expressed in vector notation, to obtain the 3-D mean μ_s and covariance C_s of the stereo error in the spherical coordinate system. The mean is given by

$$\mu_s = \begin{pmatrix} \mu_r \\ \mu_\phi \\ \mu_\theta \end{pmatrix} \tag{7.42}$$

Unlike the errors in the Cartesian coordinates, which have strong mutual correlations, the errors in r, ϕ and θ are nearly uncorrelated. It is reasonable, then, to approximate the error covariance matrix in spherical coordinates as

$$C_s = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \tag{7.43}$$

If J is the Jacobian of the Cartesian-to-spherical transformation, the measurement error mean μ and covariance C in x, y and z are approximately given by:

$$\mu = J\mu_s \tag{7.44}$$

and

$$C = JC_sJ^T \tag{7.45}$$

A final approximation is involved in computing μ and C: the true 3-D position vector p is never available (except possibly during calibration), and the *measured* position vector $\hat{p}$ has to be used instead.

7.4 Batch Estimation

The basic motion model, described by (7.1), assumes the trajectory of a point to be of the form

$$p(t) = p(0) + \dot{p}(0)\ t + \ddot{p}(0)\ t^2/2! + \cdots + p^{(n)}(0)\ t^n/n!,$$

where n is assumed to be small. The noisy measurements of the point's position, available at N discrete time instants, are written as

$$\tilde{p}_i,\ i = 0, 1, \ldots, N-1$$

The ith measurement, $\tilde{p}_i$, is obtained at time instant t_i, $i = 0, 1, \ldots, N-1$. These measurements may be collected in a single measurement vector $\mathbf{z}$, defined by

$$\mathbf{z} = \begin{bmatrix} \tilde{p}_0 \\ \tilde{p}_1 \\ \vdots \\ \tilde{p}_{N-1} \end{bmatrix}. \tag{7.46}$$

The parameters to be estimated may likewise be placed in a single parameter vector $\boldsymbol{\theta}$, given by

$$\boldsymbol{\theta} = \begin{bmatrix} p(0) \\ \dot{p}(0) \\ \ddot{p}(0) \\ \vdots \\ p^{(n)}(0) \end{bmatrix}. \tag{7.47}$$

The batch estimation problem may now be stated as follows: find the best estimate of $\boldsymbol{\theta}$ given the measurements $\mathbf{z}$ and the motion model (7.1). For

our particular problem, wherein no prior information is assumed about θ, the "best" estimate may be considered to be the one with the maximum likelihood. To be more precise, the "best" estimate is the one given by

$$\hat{\theta} = \max_{\theta}{}^{-1}\, p(\mathbf{z}\,/\theta) \tag{7.48}$$

where the inverse sign is used to indicate that we require the value of θ that maximizes the conditional p.d.f of $\mathbf{z}$ (and not the maximum value of the p.d.f.). Assuming that the measurements are mutually independent,

$$p(\mathbf{z}\,/\theta) = \prod_{i=0}^{N-1} p(\tilde{p}_i\,/\theta) \tag{7.49}$$

The mean of the error in each measurement is known, and can be subtracted from it. Assuming this to have been done already, the statistics of $\tilde{p}_i$ are obtained as follows:
The mean is given by

$$\mathcal{E}\,\{\tilde{p}_i\,/\theta\} = p_i \tag{7.50}$$

with

$$\begin{aligned} p_i &= p(0) + \dot{p}(0)\; t_i + \ddot{p}(0)\; t_i^2/2! + \cdots + p^{(n)}(0)\; t_i^n/n! \\ &= T_i\; \theta \end{aligned} \tag{7.51}$$

The matrix T_i in the above equation is given by

$$T_i \triangleq \left[I_3 \vdots t_i\; I_3 \vdots t_i^2/2!\; I_3 \vdots \cdots \vdots t_i^n/n!\; I_3\right] \tag{7.52}$$

where I_3 is the 3×3 unit matrix. The covariance is given by

$$\mathcal{E}\left\{(\tilde{p}_i - p_i)(\tilde{p}_i - p_i)^T\right\} = C_i, \tag{7.53}$$

where C_i is obtained as explained in the section on stereo error analysis.

At this stage, it is useful to assume that the error is Gaussian in nature. This assumption is obviously not valid in a rigorous sense, since errors due to

quantization are usually bounded. However, it is a reasonable approximation when the errors are small in magnitude compared to the measurements [42]. The advantage of making this assumption is that it allows us to obtain a closed-form expression for the batch estimate. With this assumption, and using (7.50) and (7.53),

$$\begin{aligned} p(\mathbf{z}\ /\boldsymbol{\theta}) &= \prod_{i=0}^{N-1} \frac{1}{(2\pi)^{3/2}|C_i|^3}\, \mathrm{e}^{-\frac{1}{2}(\tilde{p}_i - p_i)^T C_i^{-1}(\tilde{p}_i - p_i)} \\ &= \prod_{i=0}^{N-1} \frac{1}{(2\pi)^{3/2}|C_i|^3}\, \mathrm{e}^{-\frac{1}{2}(\tilde{p}_i - T_i\boldsymbol{\theta})^T C_i^{-1}(\tilde{p}_i - T_i\boldsymbol{\theta})} \end{aligned} \tag{7.54}$$

The above Gaussian likelihood function has to be maximized w.r.t $\boldsymbol{\theta}$. It is well known that this is equivalent to minimizing the negative of the log-likelihood w.r.t $\boldsymbol{\theta}$. Applying this principle, and eliminating the terms independent of $\boldsymbol{\theta}$, we get

$$\hat{\boldsymbol{\theta}} = \min_{\boldsymbol{\theta}}\ ^{-1} \sum_{i=0}^{N-1} (\tilde{p}_i - T_i\boldsymbol{\theta})^T C_i^{-1} (\tilde{p}_i - T_i\boldsymbol{\theta}) \tag{7.55}$$

Expanding the r.h.s. of (7.55) and setting it to zero,

$$\frac{\partial}{\partial\boldsymbol{\theta}} \sum_{i=0}^{N-1} \left[\tilde{p}_i^T C_i^{-1}\tilde{p}_i + \boldsymbol{\theta}^T T_i^T C_i^{-1} T_i\boldsymbol{\theta} - \boldsymbol{\theta}^T T_i^T C_i^{-1}\tilde{p}_i - \tilde{p}_i^T C_i^{-1} T_i\boldsymbol{\theta}\right] |_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = 0$$

Using the following two identities of matrix calculus:

$$\frac{\partial(M\mathbf{a})}{\partial\mathbf{a}} = M \text{ and } \frac{\partial(\mathbf{a}^T M\mathbf{a})}{\partial\mathbf{a}} = 2M\mathbf{a},$$

we obtain

$$2\sum_{i=0}^{N-1} T_i^T C_i^{-1} T_i\hat{\boldsymbol{\theta}} - 2\sum_{i=0}^{N-1} T_i^T C_i^{-1}\tilde{p}_i = 0 \tag{7.56}$$

Defining

$$M \triangleq \sum_{i=0}^{N-1} T_i^T C_i^{-1} T_i \tag{7.57}$$

and

$$\mathbf{b} \triangleq \sum_{i=0}^{N-1} T_i^T C_i^{-1}\tilde{p}_i, \tag{7.58}$$

we can write (7.56) as

$$M\hat{\theta} = \mathrm{b} \tag{7.59}$$

The rank of the square, symmetric matrix M depends on the number of terms in the summation (N). In general, if the number of point measurements N is much greater than the number of derivatives estimated (n), M will be nonsingular. In that case,

$$\hat{\theta} = M^{-1}\mathrm{b} \tag{7.60}$$

The statistical properties of the above estimate can be easily derived. Let us define

$$\Delta\theta \triangleq \hat{\theta} - \theta \tag{7.61}$$

The statistics of interest are the mean and the covariance of the estimation error $\Delta\theta$.

$$\begin{aligned}
\mu_\theta &\triangleq \mathcal{E}\{\Delta\theta\} \\
&= \mathcal{E}\left\{\hat{\theta}\right\} - \theta \\
&= M^{-1}\mathcal{E}\{\mathrm{b}\} - \theta \\
&= M^{-1}\sum_{i=0}^{N-1} T_i^T C_i^{-1}\mathcal{E}\{\tilde{p}_i\} - \theta
\end{aligned}$$

Using (7.50) and (7.51)

$$\begin{aligned}
\mu_\theta &= M^{-1}\sum_{i=0}^{N-1} T_i^T C_i^{-1} T_i\theta - \theta \\
&= M^{-1}M\theta - \theta \\
&= 0
\end{aligned} \tag{7.62}$$

The covariance of the estimate is computed as follows:

$$\begin{aligned}
C_\theta &= \mathcal{E}\left\{(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T\right\} \\
&= \mathcal{E}\left\{\left[M^{-1}\sum_{i=0}^{N-1} T_i^T C_i^{-1}(\tilde{p}_i - p_i)\right]\left[M^{-1}\sum_{i=0}^{N-1} T_i^T C_i^{-1}(\tilde{p}_i - p_i)\right]^T\right\}
\end{aligned}$$

$$= M^{-1}\left[\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} T_i^T C_i^{-1} \underbrace{\mathcal{E}\left\{(\tilde{p}_i - p_i)(\tilde{p}_j - p_j)^T\right\}}_{C_i} C_j^{-1} T_j\right] M^{-1}$$

Using (7.53) and the assumption that measurement errors in two different 3-D measurements are uncorrelated,

$$\begin{aligned} C_\theta &= M^{-1}\underbrace{\left[\sum_{i=0}^{N-1} T_i^T C_i^{-1} T_i\right]}_{M} M^{-1} \\ &= M^{-1} \end{aligned} \tag{7.63}$$

7.5 Recursive Estimation

The idea here is to formulate the estimation of a point's trajectory as a recursive tracking problem, based on a plant model and a measurement model. The quantities to be estimated i.e. the position of the point and the derivatives thereof are contained in a state vector $\mathbf{s}$, defined by

$$\mathbf{s} \triangleq \begin{bmatrix} p(t) \\ \dot{p}(t) \\ \ddot{p}(t) \\ \vdots \\ p^{(n)}(t) \end{bmatrix} \tag{7.64}$$

The difference between the above state vector and the parameter vector θ estimated by the batch procedure is that the terms in $\mathbf{s}$ are referenced to the current time instant t, rather than to the initial time instant $t = 0$.

The plant model describes the time evolution of the state vector. Using (7.1), which expresses the assumption that $p^{(m)}(t) = 0 \,\forall\, m > n$, it can be written as

$$\dot{\mathbf{s}}(t) = \mathcal{F}\mathbf{s}(t) + \mathbf{w}(t) \tag{7.65}$$

where $\mathbf{w}$ is a noise term included to take into account modelling errors, and the matrix $\mathcal{F}$ is of the form

$$\mathcal{F} = \begin{bmatrix} 0 & I_3 & 0 & 0 & \cdots & 0 \\ 0 & 0 & I_3 & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & I_3 \\ 0 & 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix} \tag{7.66}$$

Equation (7.65) has to be discretised, in order that it can describe the evolution of the state vector from one sampling instant to another. The discrete version of the plant model can be found by integration of (7.65) over the sampling interval (i.e. interframe period), and the result is

$$\mathbf{s}(k) = F\,\mathbf{s}(k-1) + \mathbf{w}_k \tag{7.67}$$

where the matrix F is given by

$$F = \begin{bmatrix} I_3 & t\,I_3 & \frac{t^2}{2!}I_3 & \cdots & \cdots & \frac{t^n}{n!}I_3 \\ 0 & I_3 & t\,I_3 & \frac{t^2}{2!}I_3 & \cdots & \frac{t^{n-1}}{(n-1)!}I_3 \\ \vdots & \vdots & \ddots & & & \vdots \\ 0 & 0 & \cdots & \cdots & I_3 & t\,I_3 \\ 0 & 0 & \cdots & \cdots & \cdots & I_3 \end{bmatrix} \tag{7.68}$$

The input data to the estimator is in the form of 3-D point positions, one for each sampling instant. The measurement model, which shows the relationship between the state vector $\mathbf{s}$ and the obervation (measurement) vector $\mathbf{z}$ is given by

$$\mathbf{z}(k) = H\,\mathbf{s}(k) + \mathbf{v}_k \tag{7.69}$$

where

$$H = \left[I_3 \vdots 0 \vdots 0 \vdots \cdots \vdots 0 \right] \tag{7.70}$$

and the measurement noise $\mathbf{v}$ has the statistics described in Section 7.3.

7.6 The 3D Segmentation

Once the motion parameters of the 3D contour chain points are estimated by this recursive method we need to interpret them in terms of the motion of the various physical objects in the scene. A segmentation of the contour chain points into groups of points close in 3D is performed. Then for each labeled group of points we compute the statistics (mean, standard deviation and number of points) of its motion parameters. The segmentation algorithm is based on a criteria checking the continuity of the disparity (calculated by the stereo algorithm) between chains close in 2D [54].

The algorithm has four phases:

- Noise removal: the contour chain points for which the disparity is below a certain threshold are removed, and the contour chains for which the ratio

$$\frac{\text{number of 3D points associated with the chain}}{\text{number of points in the chain}}$$

  is below a certain threshold are removed.

- Neighborhood creation: for each chain a neighborhood is created by growing around its extremities. Once a chain coherent in terms of the disparity with the current chain has been found the growing is stopped.

- Labelling: each chain belonging to this neighborhood and coherent in terms of the disparity with the current chain receives the label of the current chain.

- Fusion: once all the chains have been processed, the labels are refreshed and the output of the algorithm are groups of chains spatially coherent in 3D.

7.7 Experimental Results

The recursive algorithm was tested on several real image sequences, and sample results on one stereo image sequence (with seven image pairs[2]) are discussed in this section. The estimation was done for $n = 1$ i.e. assuming constant velocities. The first and last pairs of the sequence are shown in Figs. 7.2 and 7.3. The contours extracted from the first and last right image in the sequence are shown in Fig. 7.4.

The points lying on the extracted contour chains were chosen as primitives for the stereo matching, and the reconstructed 3-D results are shown in Fig. 7.8, as visualised from above (i.e. in the x-z plane). It is possible to identify the van, the cyclist and the marks on the road. Fig. 7.6 shows a sample of typical results for optic flow, using the same primitives and the same algorithm used for the stereo matching.

The optic flow between successive images can be used to obtain the image plane trajectories of contour chain points, as shown in Fig. 7.7. The trajectories that begin in the first image and reach the final one are shown in Fig. 7.8, superimposed on the first and final right image. (This eliminates the shorter trajectories.) The trajectories contained inside the box in the lower image in Fig. 7.8 are selected to demonstrate the performance of the recursive filter.

The selected (noisy) trajectories are shown (in plan) in the upper image in Fig. 7.9. The effects of the quantization errors are evident. The same trajectories, after filtering, are shown in the lower image in Fig. 7.9. The trajectories are smoother, although some of them appear to be still fairly noisy. Further smoothness can be attained by reducing the initial covariance of the velocity states, but this may reduce the accuracy of the velocity estimates. The estimates of the velocities are shown in Figs. 7.11 and 7.10, superimposed on the last right image, for the second and seventh sampling instants respectively. The velocities in the x-, y- and z-directions are shown, respectively, with horizontal, vertical and diagonal arrows, whose lengths are proportional to the

[2]The original sequence had 16 image pairs. Alternate image pairs, from the first through the 13th are used for the temporal analysis.

	lifetime	count	V_x	Σ_x	V_y	Σ_y	V_z	Σ_z
van	5	293	0.095	0.90	8.72	6.53	-0.74	0.99
	6	220	0.125	0.895	9.45	4.495	-0.68	0.56
	7	197	0.085	0.715	9.525	4.26	-0.685	0.54
motor-cycle	5	41	-0.925	0.41	4.055	3.35	-0.225	0.195
	6	20	-0.95	0.515	5.215	3.95	-0.28	0.2
	7	19	-0.985	0.505	4.875	3.735	-0.265	0.195
road mark	5	20	0.095	0.225	-6.58	2.415	-0.585	0.285
	6	18	0.15	0.215	-7.265	1.32	-0.515	0.195
	7	16	0.14	0.225	-7.145	1.27	-0.545	0.17

Table 7.1: Statistical analysis of the velocity estimates (V_x, V_y, V_z) expressed in km/h.

magnitudes of the velocities they represent. The velocity of the observer can be recovered from the (apparent) velocities of the points on the road. For the image sequence under consideration, velocities of 10-15 km/hr for the observer, and 20-25 km/hr for the van were obtained, all in the direction of increasing depth.

The results of segmentation are shown in Fig. 7.12. Contours belonging to different objects are shown with different grey levels. We can distinguish the van, the motorcycle and the markings on the surface of the road. Table 7.7 summarizes the statistical analysis of the velocities of three objects in the scene (van, motorcycle and road line mark) at three different time instants. The velocities are expressed in kilometers per hour. As this scene was taken in a busy urban street the various objects were moving slowly relative to each other When a sufficient number of points are observed, and the velocities are significant, we can verify that velocity estimates are more consistent as the frame number increases.

7.8 Conclusions

In this chapter, we have presented a method of tracking points in a scene over an image sequence of arbitrary length, using stereo and optical flow, and to

analyse the point trajectories in 3D using estimation theoretic methods. The algorithms are shown to give meaningful results on real image data obtained from moving vehicles in real traffic situations. The estimates of the point trajectories can be used to assist in navigation and obstacle avoidance. A detailed discussion of the possible ways in which the estimates can be interpreted to achieve these objectives is beyond the scope of this chapter.

Figure 7.2: First and final images taken by the left camera.

Figure 7.3: First and final images taken by the right camera.

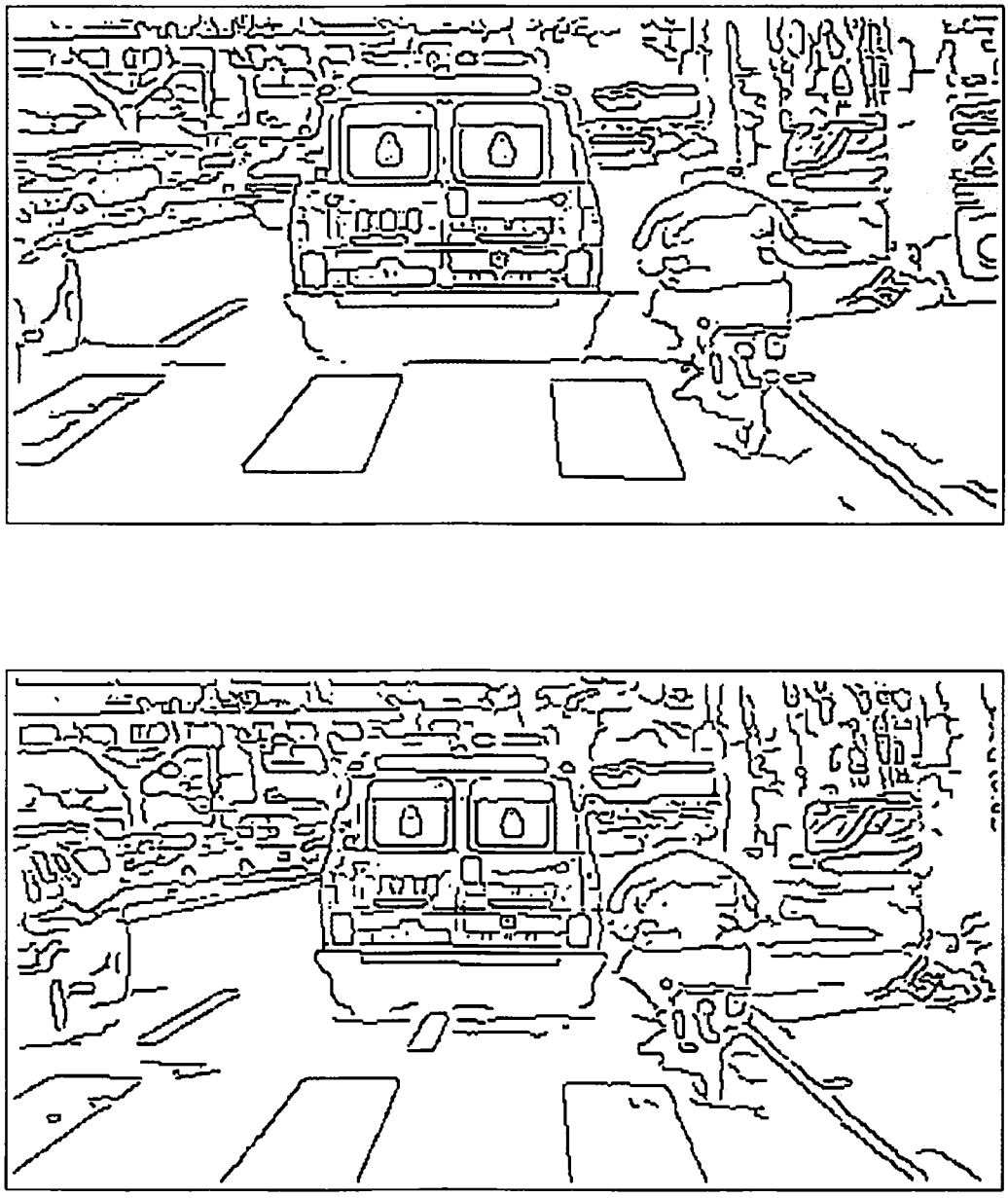

Figure 7.4: Contours extracted from the first and final right images

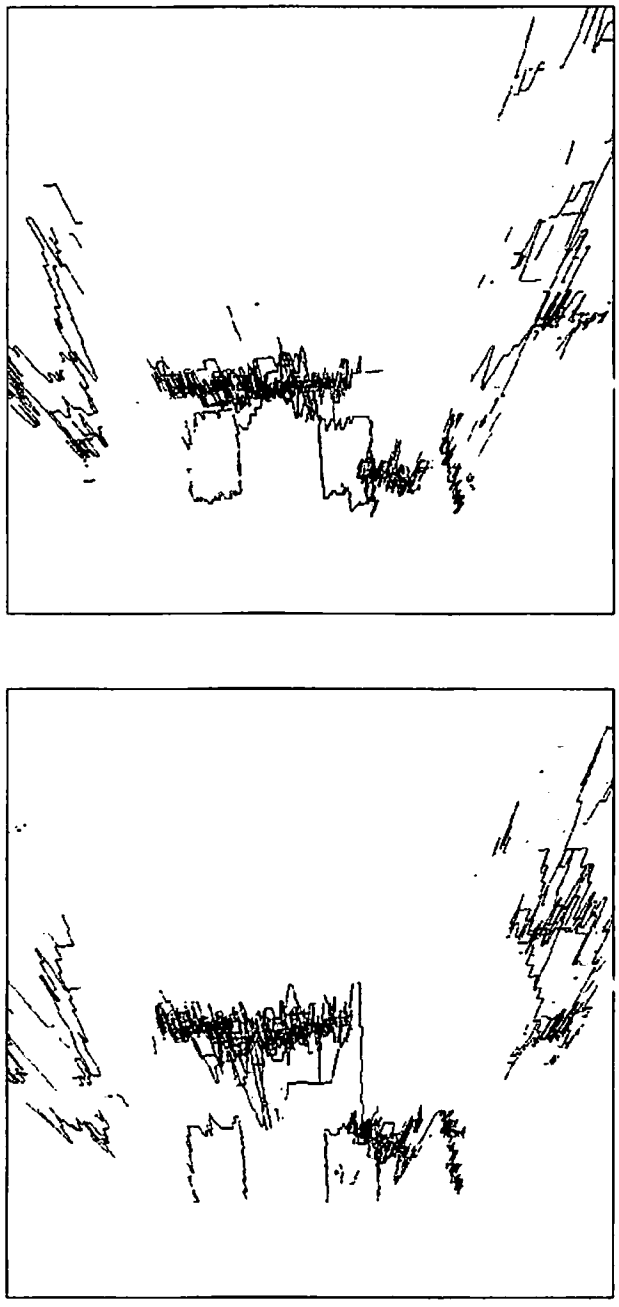

Figure 7.5: 3-D (stereo) results for the first and last image pairs

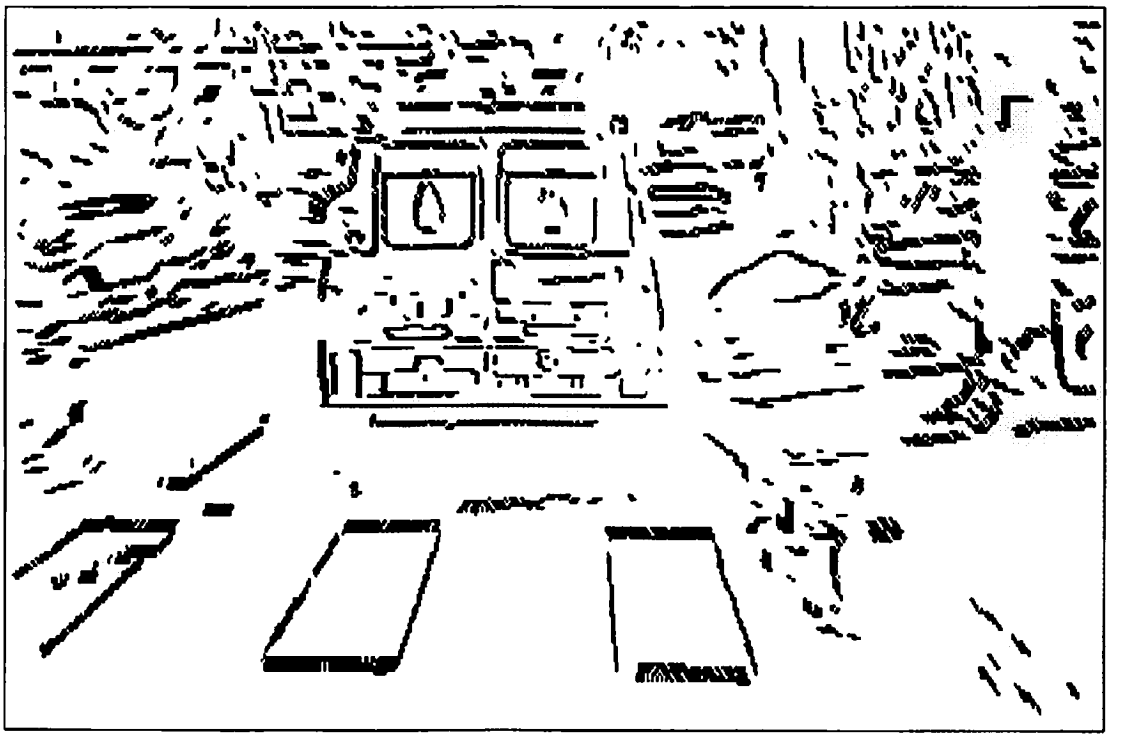

Figure 7.6: Optic flow between right images 1 & 2

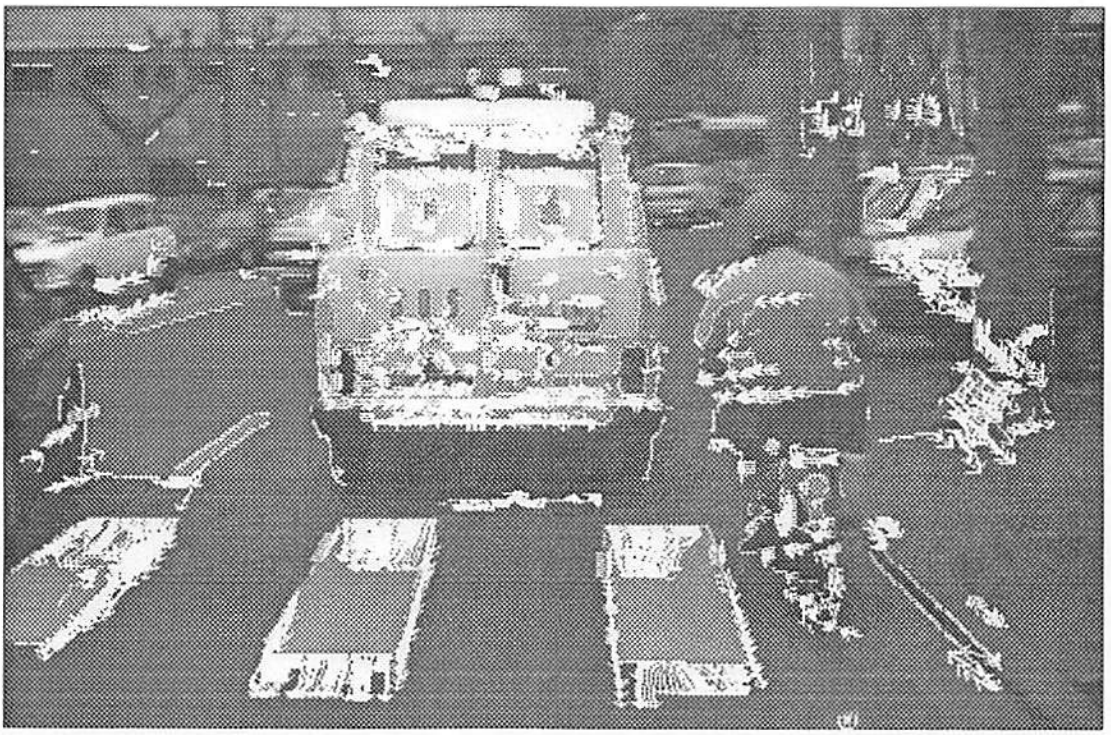

Figure 7.7: Image plane trajectories of points detected in the first image

Figure 7.8: Image plane trajectories of length = 7

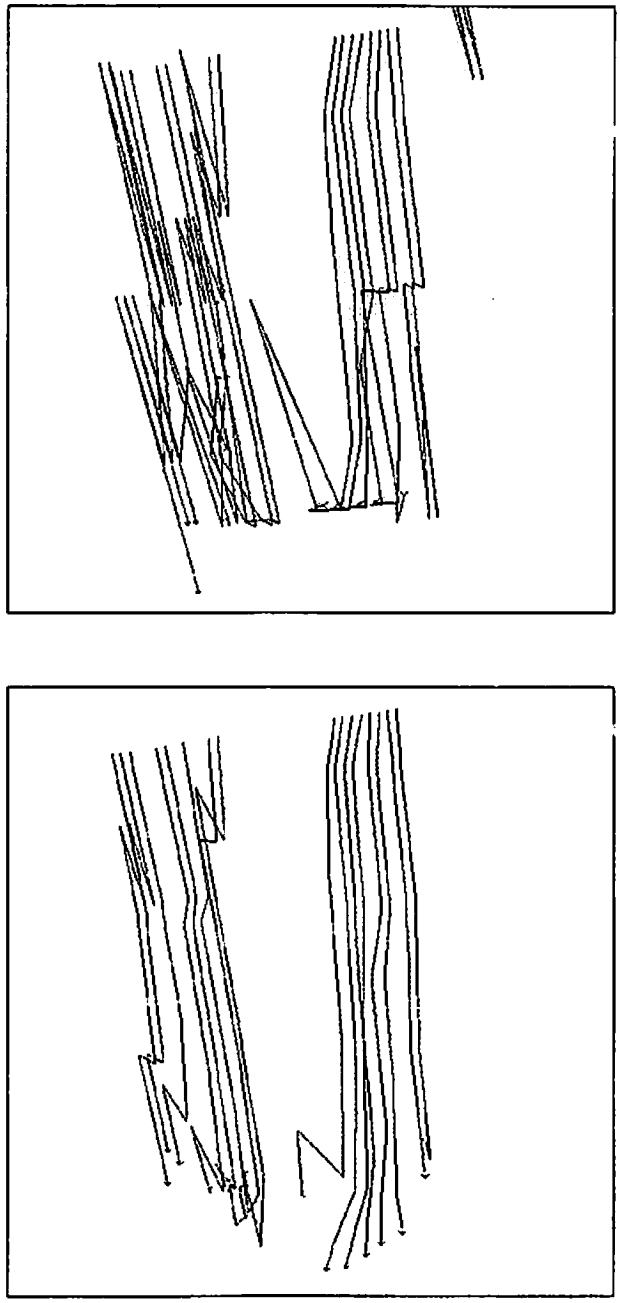

Figure 7.9: Selected point trajectories seen from above, before and after filtering

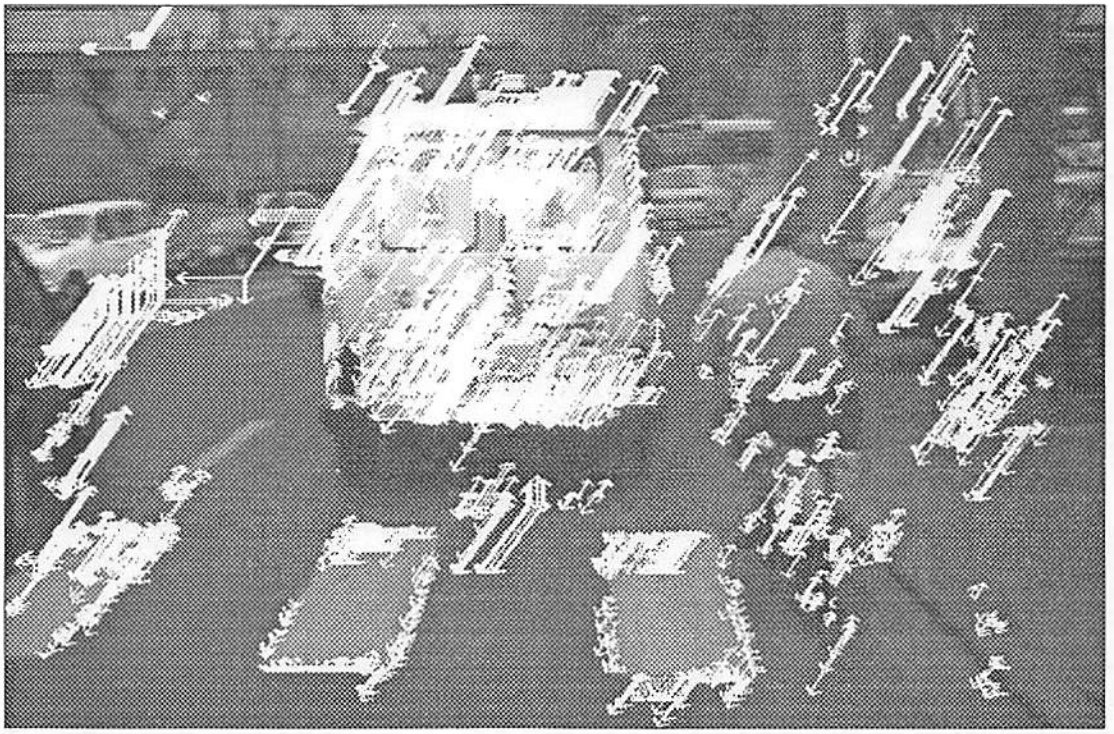

Figure 7.10: Velocity estimates at $t = 2$

Figure 7.11: Velocity estimates at $t = 7$

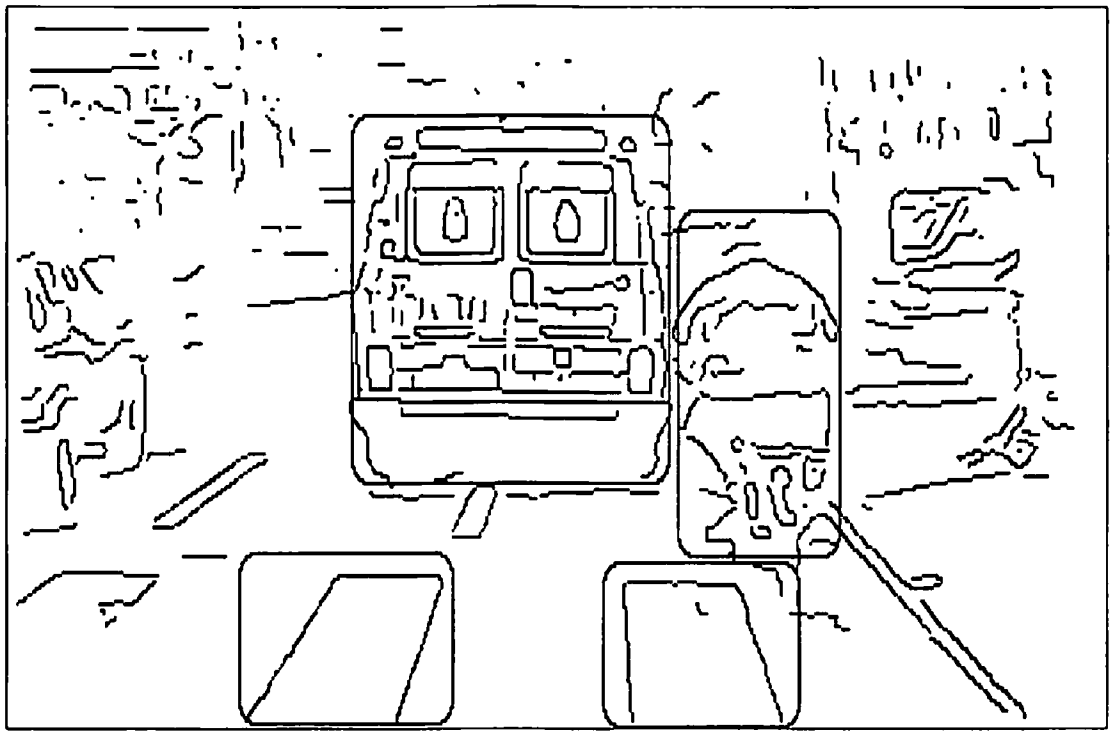

Figure 7.12: 3-D segmentation

Chapter 8

Conclusions and Directions for Future Research

Existing methods of visual motion estimation and analysis are deficient in many respects; a method which is robust, realistic, practical and general-purpose does not exist for this problem. The only systems which have experienced at least some success are those that have been designed for very specific applications, and for highly restricted environments.

It is our opinion that this state of affairs is due to the shortcomings of the current approaches to the problem, and not due to the nature of the problem itself. We feel that the mathematical and algorithmic tools needed to address the problem already exist; designing a successful motion analysis system will involve the effective use of these tools to perform and to integrate the different stages of processing required to solve the problem.

In this dissertation we have explained some of these tools, and how they may be effectively combined into a model-based long-frame estimation-theoretic motion paradigm, which is applicable to a variety of situations, such as the single-camera navigation of a land vehicle and the tracking of the motion of a rigid object.

8.1 Conclusions

One of the main discoveries made during this research was the fact that simple motion models in conjunction with recursive methods can lead to very robust algorithms. The experimental results on real image sequences demonstrate the ability of the paradigm to cope with fairly large modelling and measurement errors. None of the assumptions made (such as smoothness of motion) seems to be critical to its success in a real application. The motion models used are treated as "soft" rather than "hard" models, in the sense that the data are only expected to conform to them approximately. For instance, the motion model used for the experiments in Chapter 4 on passive navigation assumed constant camera orientation, but the data for one of the synthetic sequences and the real image sequence violate this assumption. Nonetheless, the recursive estimator does not fail, although its performance is slightly degraded compared to the experiment in which the model assumptions are obeyed exactly. The real image experiment involved other modelling errors such as discrepancies between the actual image point locations and those predicted by the structure and pose ground truth and camera calibration parameters. The actual source of error is not known—it could be a combination of incorrect ground truth values and inadequate camera modelling—but its knowledge is not crucial to the success of the recursive procedure. It would be helpful, of course, to have a good model for the errors in the measured image-plane coordinates of feature points. Currently we assume the image-plane errors to be independent and identically distributed. If a more appropriate measurement error model is available, it should be used instead.

Another important aspect of the motion analysis problem is the need to integrate different stages of processing. The traditional approach views the motion problem into a mapping from M feature correspondences to motion and structure parameters. This attempt to treat motion analysis as a mathematical abstraction is now widely acknowledged as inadequate, because it does not take into account the various stages of processing required, such as feature extraction and matching. Treating the feature correspondence problem as distinct from the motion estimation problem can lead to methods that look good on paper

but are difficult to apply on a practical problem. In this thesis, we have shown how feature point matching can be integrated with motion estimation, in a mutually beneficial way.

The models used in this research should not be viewed as definitive, but as specific examples. The kind of motion models to be used depends on the application. The selection of a suitable model is non-trivial; different, but equivalent parametrizations of a problem can lead to recursive algorithms with different strengths and weaknesses. For example, Euler angles can be used to represent rotation instead of quaternions. This has the advantage of reducing the number of states by one, but has the drawback that introduces trigonometric nonlinearities into the formulation. Therefore it is likely that in this case the batch formulation (nonlinear least squares) will work better than before, and the recursive formulation (linearized incremental least-squares) will not perform as well as before. Concepts like this can be made somewhat more precise using the ideas presented in Chapter 5, on the evaluation of model-based formulations.

8.2 Directions for Future Work

There are various possible extensions to our work. One could experiment with higher order models. However it has been our experience that increasing the complexity of the motion models could result in less robust parameter estimates, unless the amount of data available is increased proportionately. The issue of multiple rigid motion has to be addressed. This would involve segmentation of the scene into different objects based on motion, and a separate recursive filter for each independently moving component. A combination of optic flow and feature-based methods may be necessary in this case.

In this section, we discuss some possible areas for future work in long-frame motion analysis.

8.2.1 A Different Model for Passive Navigation

The model used in Chapter 4 implicitly assumes that the origin of the CCS coincides with the centre of rotation. As explained earlier, model assumptions such as this need not be strictly valid; minor deviations are easily handled by the recursive algorithm. But there may be applications where it is necessary to determine the exact axis of rotation and its relationship to the camera's position. In such a case, a slightly different formulation, which has three additional parameters to express this relationship, is proposed in this section. This formulation also incorporates line features in addition to point features. The basic models of motion and imaging are shown in Fig. 8.1 and the geometry of line features in Fig. 8.2.

Imaging Model

A point in space is represented as usual by its coordinates (x, y, z), and its projection on the image (X, Y) is given by

$$X = f \cdot \frac{x}{z}, \quad Y = f \cdot \frac{y}{z} \tag{8.1}$$

The camera focal length f is set to unity w.l.o.g. A line in space is represented by two vectors

$$\mathbf{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}$$

and

$$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}.$$

The vector $\mathbf{p}$ is the perpendicular to the line from the origin, and $\mathbf{v}$ is a vector in the direction of the line, so that

$$\mathbf{p} \cdot \mathbf{v} = 0. \tag{8.2}$$

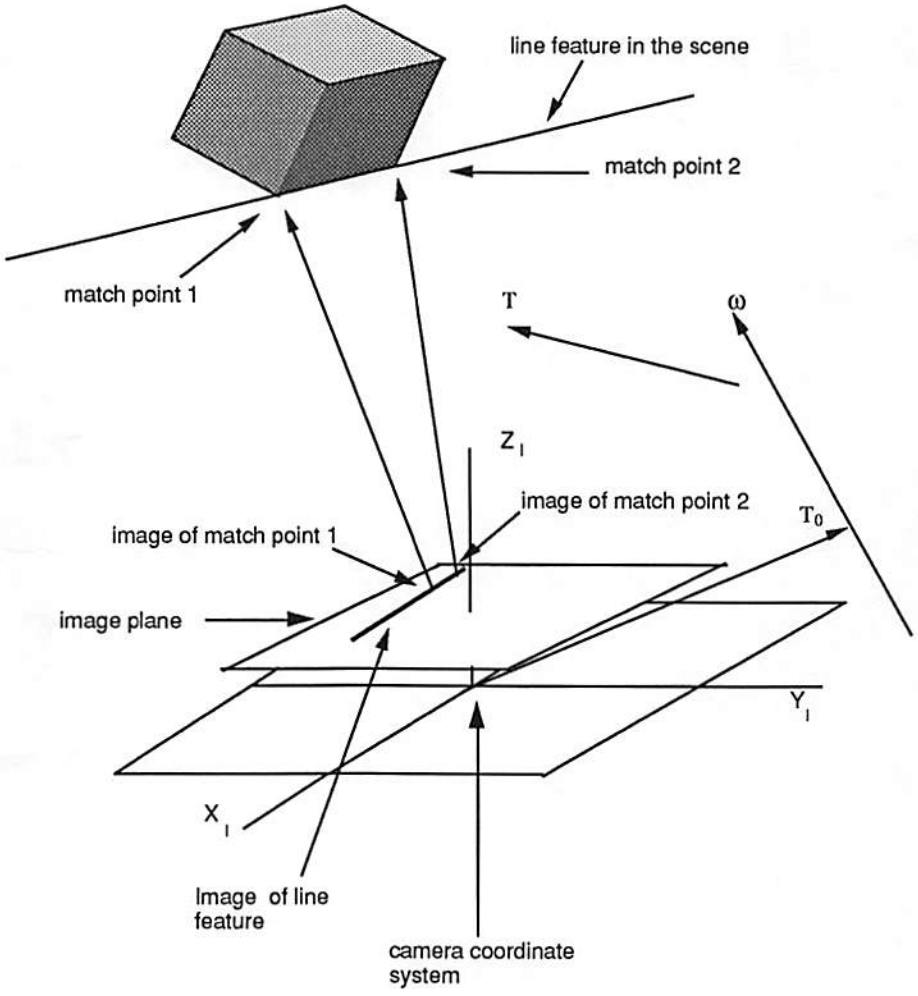

Figure 8.1: Alternative models of motion and imaging for passive navigation

Any point $\mathbf{r} = (x, y, z)^T$ on the line can be expressed as

$$\mathbf{r} = \mathbf{p} + \alpha \mathbf{v}, \tag{8.3}$$

for some scalar α. The following constraint is used to identify $\mathbf{v}$ uniquely:

$$\|\mathbf{v}\| = 1. \tag{8.4}$$

A line in the image can be represented by its slope m and its y-intercept c. The equations relating the spatial(3-D) parameters of a line to those of its projection on the image plane are found by considering the projection of a general point on the line. Using (8.3) and (8.1), we get

$$X = \frac{x}{z} = \frac{p_x + \alpha v_x}{p_z + \alpha v_z} \tag{8.5}$$

$$Y = \frac{y}{z} = \frac{p_y + \alpha v_y}{p_z + \alpha v_z} \tag{8.6}$$

$$m = \frac{dY}{dX} = \frac{\frac{dY}{d\alpha}}{\frac{dX}{d\alpha}} = \frac{p_z v_y - p_y v_z}{p_z v_x - p_x v_z} \tag{8.7}$$

$$c = Y - m\ X \tag{8.8}$$

The above equation should hold for any value of α. After some manipulation, we can eliminate α to obtain

$$c = \frac{p_x v_y - p_y v_x}{p_z v_x - p_x v_z} \tag{8.9}$$

In practice, it is better to represent a 2-D line by the parameters d and ϕ as shown in Fig. 8.2. These may be determined from m and c using simple geometric relations.

Motion Model

All velocities are assumed to be constant in time, and the sampling period is assumed to be fixed. The rotation of the camera is represented by the vector ω. The translational velocity is $\mathbf{u}$. The translation vector T is the product

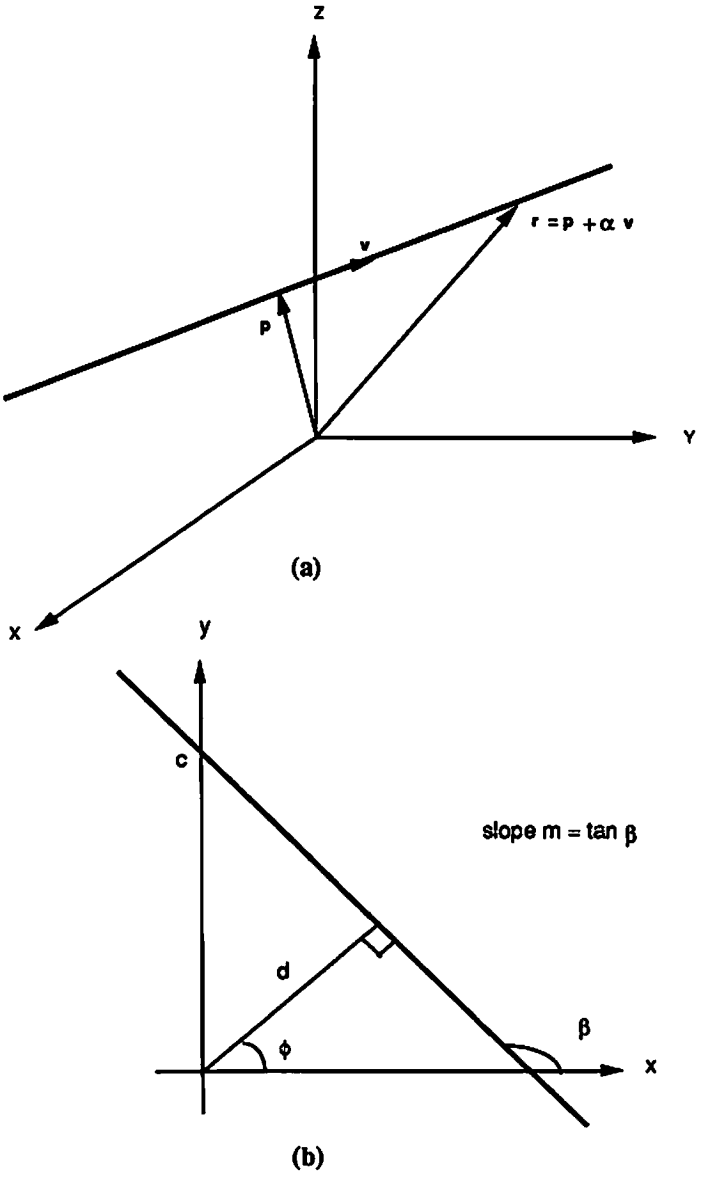

Figure 8.2: Line feature in 3-D (a) and its projection onto the image plane (b)

of the translational velocity and the sampling period T_{samp}. The interframe rotation angle θ is given by

$$\theta = \|\omega\| \, T_{samp} \, .$$

The rotation matrix R can be obtained from θ using a standard formula [63]. The initial displacement of the axis of rotation from the origin is represented by T_0. To identify it uniquely, we impose the condition

$$\omega \cdot T_0 = 0. \tag{8.10}$$

In addition, we use the following constraint to fix the global scale factor of the system (which is required because of the loss of absolute depth information in central projection)

$$\|T_0\| = 1. \tag{8.11}$$

The (relative) motion of a point $\mathbf{q}$ in the scene is given by a recursive equation [60] of the form:

$$\mathbf{q}_{i+1} = R(\mathbf{q}_i - (i-1)T - T_0) + i\,T + T_0. \tag{8.12}$$

The equations relating to the motion of a line are obtained by considering the motion of a general point on the line.

$$\begin{aligned} (\mathbf{p} + \alpha\mathbf{v})_{i+1} \;&=\; R\{(\mathbf{p} + \alpha\mathbf{v})_i - (i-1)T - T_0\} + i\,T + T_0 = \\ &\quad [\, R(\mathbf{p}_i - (i-1)T - T_0) + i\,T + T_0 \,] + \alpha R\mathbf{v}_i \end{aligned} \tag{8.13}$$

from which, by observation, we can write:

$$\mathbf{p}'_{i+1} = R(\mathbf{p}_i - (i-1)T - T_0) + i\,T + T_0. \tag{8.14}$$

and

$$\mathbf{v}'_{i+1} = R\mathbf{v}_i. \tag{8.15}$$

The reason for denoting the above quantities with primes is that we have not verified whether they satisfy the constraints (8.2) and (8.4). It is easy to see that (8.4) is satisfied, since R is an orthonormal matrix, and hence we may write:

$$\mathbf{v}_{i+1} = R\mathbf{v}_i. \tag{8.16}$$

However, condition (8.2), which specifies that $\mathbf{p}$ should be orthogonal to $\mathbf{v}$ is violated, and hence we have to reimpose this condition as follows:

$$\mathbf{p}_{i+1} = \mathbf{p}'_{i+1} - (\mathbf{p}'_{i+1} \cdot \mathbf{v}_{i+1})\mathbf{v}_{i+1} \tag{8.17}$$

Batch Formulation

Let us assume that we have P point correspondences and L line correspondences over N frames. The vector $\mathbf{u}$ of unknown parameters is

$$\mathbf{u} = \begin{pmatrix} \mathbf{w} \\ \mathbf{u} \\ T_0 \\ \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_P \\ \mathbf{p}_1 \\ \mathbf{v}_1 \\ \mathbf{p}_2 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{p}_L \\ \mathbf{v}_L \end{pmatrix} \tag{8.18}$$

The input data (i.e. the observations) are given by the following equations:

$$X_i(k) \;=\; h_X[\mathbf{r}_i(\mathbf{u}, k)] + n_X(k) \tag{8.19}$$

$$Y_i(k) = h_Y[\mathbf{r}_i(\mathbf{u},k)] + n_Y(k) \tag{8.20}$$

$$d_i(k) = h_d[\mathbf{p}_i(\mathbf{u},k), \mathbf{v}_i(\mathbf{u},k)] + n_d(k) \tag{8.21}$$

$$\phi_i(k) = h_\phi[\mathbf{p}_i(\mathbf{u},k), \mathbf{v}_i(\mathbf{u},k)] + n_\phi(k) \tag{8.22}$$

The functions h_X etc. are known nonlinear functions obtained from the central projection model for imaging. The terms n_X etc. account for measurement noise. The argument k denotes the frame number of the image in the sequence.

If the noise terms are assumed to be independent, a weighted least-square estimate of $\mathbf{u}$ can be found by minimizing with respect to $\mathbf{u}$ the weighted sum of the squared residuals

$$G(\mathbf{u}) = \sum_{k=1}^{P} \Bigg[\sum_{i=1}^{M} \{(X_i(k) - h_X[\mathbf{r}_i(\mathbf{u},k)])^2 + \beta(Y_i(k) - h_Y[\mathbf{r}_i(\mathbf{u},k)])^2\} + \sum_{i=1}^{L} \{\gamma(d_i(k) - h_d[\mathbf{p}_i(\mathbf{u},k), \mathbf{v}_i(\mathbf{u},k)])^2 + \delta(\phi_i(k) - h_\phi[\mathbf{p}_i(\mathbf{u},k), \mathbf{v}_i(\mathbf{u},k)])^2\} \Bigg] \tag{8.23}$$

subject to

$$\omega \cdot T_0 = 0 \tag{8.24}$$

$$\|T_0\| = 1 \tag{8.25}$$

$$\mathbf{p}_i \cdot \mathbf{v}_i = 0, \quad i = 1, \ldots, L \tag{8.26}$$

and

$$\|\mathbf{v}_i\| = 1, \quad i = 1, \ldots, L. \tag{8.27}$$

The weights β, γ and δ have to be chosen depending on the expected variance of the noise in each term.

The above minimization problem may converge very slowly on account of the nonlinear constraints. To overcome this problem, we can change some of the constraints, in such a way that the (modified) parameter vector can be obtained by an unconstrained minimization procedure. The following modifications can be made:

- Instead of (8.11), the scale factor is set by fixing the z-component of T_0 to be unity. z_0 can now be removed from $\mathbf{u}$. Using (8.10), one component of ω (say ω_z) can be removed form $\mathbf{u}$.

- Instead of (8.4), the constraint used is

$$\mathbf{v} \cdot \hat{\mathbf{y}} = 1, \ or \ v_y = 1.$$

  Using this and (8.2), we can now remove the y-component of all the $\mathbf{v}_i$'s and one component (say the z-component) of all the $\mathbf{p}_i$'s from $\mathbf{u}$.

The modified minimization problem is unconstrained, and can be solved with much greater ease than the original constrained minimization.

8.2.2 Improvements in Object Tracking

In the current implementation, as explained in Chapter 6, the observation vector $\mathbf{z}$ in the state-space representation (6.3) contains only the image plane coordinates of the feature points, which do not depend directly on the velocity parameters of the state vector. Consequently, the estimation of these velocity parameters are very much dependent on an accurate estimate of the initial state error covariance matrix. The approximate CRLB's computed from the batch solution may not be close enough to the true variances to ensure convergence. One possible solution would be to augment the measurement vector so as to contain the image plane velocities of the feature points along with the image

plane coordinates, so that

$$\mathbf{z}(k) = \begin{pmatrix} x_1(t_k) \\ y_1(t_k) \\ x_2(t_k) \\ y_2(t_k) \\ \vdots \\ x_M(t_k) \\ y_M(t_k) \\ u_1(t_k) \\ v_1(t_k) \\ u_2(t_k) \\ v_2(t_k) \\ \vdots \\ u_M(t_k) \\ v_M(t_k) \end{pmatrix}. \tag{8.28}$$

In the above equation, x and y are image plane positions, while u and v are image plane velocities. The above modification prevents the algorithm from being "purely recursive" since at least two images have to be processed simultaneously to compute observed velocities, but may be worthwhile if it results in a substantial improvement in convergence.

Appendix A

Kalman Filtering and its Extensions

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 continuous time plant (or signal) model evolves in time according to

$$\dot{\mathbf{x}}(t) = F\mathbf{x}(t) + G_t\mathbf{w}_t. \tag{A.1}$$

The discrete plant model is then given by

$$\mathbf{x}(k+1) = \Phi_{k+1,k}\,\mathbf{x}(k) + G_k\mathbf{w}_k, \tag{A.2}$$

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

$$\mathbf{z}(k) = H(k)\mathbf{x}(k) + \mathbf{v}(k), \tag{A.3}$$

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

Following [41], 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

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + K(k)\Big[\mathbf{z}(k) - H(k)\hat{\mathbf{x}}(k|k-1)\Big] \tag{A.4}$$

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

$$K(k) = P(k|k-1)H(k)^T\Big[H(k)P(k|k-1)H(k)^T + R_k\Big]^{-1}. \tag{A.5}$$

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

$$P(k|k-1) = \Phi_{k,k-1}\, P(k-1|k-1)\, \Phi_{k,k-1}^T + G_k Q_k G_k^T. \tag{A.6}$$

The smoothed covariance matrix is

$$P(k|k) = \Big[I - K(k)H(k)\Big]P(k|k-1), \tag{A.7}$$

and the time update for the state estimate is

$$\hat{\mathbf{x}}(k+1|k) = \Phi_{k+1,k}\,\hat{\mathbf{x}}(k|k). \tag{A.8}$$

The iteration is initialized with

$$P(0|0) = P_0 = \mathrm{E}\{(\mathbf{x}(0) - \mu_{x0})(\mathbf{x}(0) - \mu_{x0})^T\} \tag{A.9}$$

and

$$\hat{\mathbf{x}}(0|0) = \mathrm{E}\{\mathbf{x}(0)\} = \mu_{x0} = \hat{\mathbf{x}}(0). \tag{A.10}$$

In the case of the the Extended Kalman Filter (EKF), the measurement update equation is simply

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + K(k)\Big[\mathbf{z}(k) - \mathrm{h}[\hat{\mathbf{x}}(k|k-1)]\Big], \tag{A.11}$$

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

$$H(k) = \left.\frac{\partial \mathbf{h}[\mathbf{x}]}{\partial \mathbf{x}}\right|_{\mathbf{x} = \hat{\mathbf{x}}(k|k-1)}. \tag{A.12}$$

The gain for the EKF is then computed as

$$K(k) = P(k|k-1)H(k)^T \left[H(k)P(k|k-1)H(k)^T + R_k\right]^{-1}. \tag{A.13}$$

If $\mathbf{h}[\cdot]$ is "close" to linear in the states, and state estimate errors are not too large, the EKF can be reasonably effective. However, if $\mathbf{h}[\cdot]$ 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 local iteration is the key feature of the IEKF.

The iterated measurement update equation is

$$\hat{\mathbf{x}}(k|k)_{n+1} = \hat{\mathbf{x}}(k|k-1) + K(k)_n \left[\mathbf{z}(k) - \mathbf{h}[\hat{\mathbf{x}}(k|k)_n] - H(k)_n \left\{\hat{\mathbf{x}}(k|k-1) - \hat{\mathbf{x}}(k|k)_n\right\}\right] \tag{A.14}$$

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

$$\hat{\mathbf{x}}(k|k)_0 = \hat{\mathbf{x}}(k|k-1) \tag{A.15}$$

The gain for the IEKF is included in the iteration as

$$K(k)_{n+1} = \hat{P}(k|k-1)H(k)_n^T \left[H(k)_n \hat{P}(k|k-1)H(k)_n^T + R_k\right]^{-1} \tag{A.16}$$

and the approximate measurement function is re-evaluated at

$$H(k)_n = \left.\frac{\partial \mathbf{h}[\mathbf{x}]}{\partial \mathbf{x}}\right|_{\mathbf{x} = \hat{\mathbf{x}}(k|k)_n} \tag{A.17}$$

Finally, after the iteration is found to yield no further improvement, the approximate smoothed covariance is computed as

$$\hat{P}(k|k)_n = \left[I - K(k)_n H(k)_n\right] \hat{P}(k|k-1). \tag{A.18}$$

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

$$\hat{\mathbf{x}}(t_{k+1}|t_k) = \int_{t_k}^{t_{k+1}} \hat{\dot{\mathbf{x}}}(\tau|t_k) d\tau + \hat{\mathbf{x}}(t_k|t_k). \tag{A.19}$$

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

$$F(\mathbf{x}_t) = \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}_t) \tag{A.20}$$

and by integrating

$$\hat{\dot{P}}(t|t_k) = F(\hat{\mathbf{x}}(t|t_k))\hat{P}(t|t_k) + \hat{P}(t|t_k)F(\hat{\mathbf{x}}(t|t_k))^T + G_t Q_t G_t^T \tag{A.21}$$

as

$$\hat{P}(t_{k+1}|t_k) = \int_{t_k}^{t_{k+1}} \hat{\dot{P}}(t|t_k)\, dt + \hat{P}(t_k|t_k). \tag{A.22}$$

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

Appendix B

Closed-Form Methods

If a very crude initial guess is sufficient, one can use single-frame or two-frame methods. If sufficient data are available, one can develop linear, closed-form algorithms to generate the necessary initial guess. In this section, we present methods of this type to estimate the pose of the camera and structure of scene points.

B.1 Single Frame Pose Estimation

The problem addressed in this section is the determination of the position and orientation (R, T) of the camera relative to the WCS, given the 3-D world coordinates and corresponding image coordinates of some feature points. This is a *pose-from-structure* (PFS) problem.

The traditional approach is to formulate it as a least-squares minimization problem. Given n points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ in the WCS and their image plane locations $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \ldots, \boldsymbol{\rho}_n$, minimize the following:

$$\min_{(R,T)} \sum_{i=1}^{n} \|\boldsymbol{\rho}_i - h[R\mathbf{p}_i + T]\|^2 \tag{B.1}$$

This can be solved using iterative optimization techniques. If six or more points are given, a closed form solution may be obtained as follows. First we write

the rotation matrix in terms of its row vectors:

$$R = \begin{pmatrix} \mathbf{r}_x \\ \mathbf{r}_y \\ \mathbf{r}_z \end{pmatrix} \tag{B.2}$$

We can now write the equations relating the 3-D location of a point $\mathbf{p}$ to its image $\boldsymbol{\rho} = (X', Y')^T$.

$$X' = f_x \frac{\mathbf{p} \cdot \mathbf{r}_x + t_x}{\mathbf{p} \cdot \mathbf{r}_z + t_z} + X_0 \tag{B.3}$$

$$Y' = f_y \frac{\mathbf{p} \cdot \mathbf{r}_y + t_y}{\mathbf{p} \cdot \mathbf{r}_z + t_z} + Y_0 \tag{B.4}$$

$$\tag{B.5}$$

Let $X = (X' - X_0)/f_x$ and $Y = (Y' - Y_0)/f_y$. In the rest of this section, it will be assumed, without loss of generality, that these normalized values are used for image coordinates. Substituting and cross-multiplying, we get two equations per point

$$X\mathbf{p} \cdot \mathbf{r}_z + Xt_z - \mathbf{p} \cdot \mathbf{r}_x - t_x = 0 \tag{B.6}$$

$$Y\mathbf{p} \cdot \mathbf{r}_z + Yt_z - \mathbf{p} \cdot \mathbf{r}_y - t_y = 0 \tag{B.7}$$

which can be written as

$$\begin{pmatrix} -\mathbf{p}' & \mathbf{0}' & X\mathbf{p}' & 1 & 0 & X \\ -\mathbf{p}' & \mathbf{0}' & Y\mathbf{p}' & 0 & 1 & Y \end{pmatrix} \begin{pmatrix} \mathbf{r}'_x \\ \mathbf{r}'_y \\ \mathbf{r}'_z \\ T \end{pmatrix} = 0 \tag{B.8}$$

Thus each point yields two linear homogeneous equations in the 12 unknowns. If six points are given, the system of equations can be solved using singular value decomposition. However, the resulting rotation matrix may not be orthonormal, since this condition has not been enforced. In order to do so, it is convenient to decouple the rotation and translation components in the equations. This is done as follows. We write the first three equations in a slightly

different way:

$$\underbrace{\begin{pmatrix} -1 & 0 & X_1 \\ 0 & -1 & Y_1 \\ -1 & 0 & X_2 \end{pmatrix}}_{\triangleq A} \underbrace{\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}}_{T} = \underbrace{\begin{pmatrix} \mathrm{p}'_1 & 0' & -X_1\mathrm{p}'_1 \\ 0' & \mathrm{p}'_1 & -Y_1\mathrm{p}'_1 \\ \mathrm{p}'_2 & 0' & -X_2\mathrm{p}'_2 \end{pmatrix}}_{\triangleq C} \underbrace{\begin{pmatrix} \mathrm{r}'_x \\ \mathrm{r}'_y \\ \mathrm{r}'_z \end{pmatrix}}_{\triangleq \mathrm{r}} \tag{B.9}$$

From the above equations, we can express T in terms of r:

$$T = A^{-1}C\mathrm{r} \tag{B.10}$$

The translation terms from the remaining nine equations can now be factored out as follows. Let M be the 9×12 matrix corresponding to these nine equations such that

$$M \begin{pmatrix} \mathrm{r} \\ T \end{pmatrix} = 0 \tag{B.11}$$

We partition M into two components as follows

$$M = \left[[M_r]_{9\times9} \mid [M_t]_{9\times3} \right] \tag{B.12}$$

The nine equations can now be written as

$$M_r\mathrm{r} + M_tT = 0 \tag{B.13}$$

substituting for T from (B.10),

$$M_r\mathrm{r} + M_tA^{-1}C\mathrm{r} = 0 \tag{B.14}$$

Defining the 9×9 matrix

$$M = M_r + M_tA^{-1}C \tag{B.15}$$

we get a system of equations of the form

$$M\mathrm{r} = 0 \tag{B.16}$$

So our goal is to solve (B.16) in the least-squares sense subject to the constraint that R is orthonormal. Once R has been determined, T can be computed from (B.10). We have not thus far succeeded in obtaining a closed form expression for the rotation matrix using (B.16). If the data are not very noisy, the following method can be used:
(1) Solve (B.16) using the singular value decomposition without the orthonormality constraint and (2) Project the solution into the space of orthonormal matrices as is done in [67]

A similar approach has been used by Ganapathy in [32, 33].

B.2 Two-frame Motion Stereo

The objective here is to determine the 3-D world coordinates $\mathbf{p}$ of a point, given its image coordinates for two camera positions, ρ_1 and ρ_2, and the corresponding camera poses in the WCS. This is a *structure from motion* (SFM) problem, and is in a sense the inverse of the previous one. Let (R_1, T_1) and (R_2, T_2) represent, respectively, the relationship between the first and second camera positions, and the WCS. Let $\mathbf{p}_1$ and $\mathbf{p}_2$ be the 3-D position vectors of the point with respect to the two camera positions.

$$\mathbf{p}_1 = R_1\mathbf{p} + T_1 \tag{B.17}$$

$$\mathbf{p}_2 = R_2\mathbf{p} + T_2 \tag{B.18}$$

Let the corresponding (normalized) image points be ρ_1 and ρ_2, given by

$$\rho_1 = (x_1/z_1, y_1/z_1)^T \tag{B.19}$$

$$\rho_2 = (x_2/z_2, y_2/z_2)^T \tag{B.20}$$

We can rewrite equations (B.17 and B.18) as

$$z_1\rho_1 = R_1\mathbf{p} + T_1 \tag{B.21}$$

$$z_2\rho_2 = R_2\mathbf{p} + T_2 \tag{B.22}$$

Using (B.21), we get

$$\mathbf{p} = R_1^T (z_1 \rho_1 - T_1) \tag{B.23}$$

Substituting for $\mathbf{p}$ in (B.22),

$$- z_1 R_2 R_1^T \rho_1 + z_2 \rho_2 = -R_2 R_1^T T_1 + T_2 \tag{B.24}$$

This equation can be written in the matrix form

$$\underbrace{\left[- R_2 R_1 \rho_1 | \rho_2 \right]}_{\triangleq A} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = -R_2 R_1^T T_1 + T_2 \tag{B.25}$$

We have therefore three linear equations in the two unknowns z_1 and z_2, which can be directly solved using the generalized inverse.

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = A^+ \left(-R_2 R_1^T T_1 + T_2 \right) \tag{B.26}$$

We can then obtain $\mathbf{p}_1$ and $\mathbf{p}_2$ as

$$\mathbf{p}_1 = z_1 \rho_1 \tag{B.27}$$

$$\mathbf{p}_2 = z_2 \rho_2 \tag{B.28}$$

Using (B.17 and B.18), we can now write down the expression for $\mathbf{p}$ as

$$\mathbf{p} = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}^+ \begin{pmatrix} T_1 - \mathbf{p}_1 \\ T_2 - \mathbf{p}_2 \end{pmatrix} \tag{B.29}$$

References

[1] G. Adiv, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. PAMI-7, pp. 384–401, July 1985.

[2] J. K. Aggarwal and A. Mitiche, "Structure and Motion from Images: Fact and Fiction," in *Proc. Third Workshop on Computer Vision: Representation and Control*, pp. 127–128, Oct. 1985.

[3] N. Ayache and O. D. Faugeras, "Building, Registrating, and Fusing Noisy Visual Maps," in *First International Conf. on Computer Vision*, pp. 73–82, June 1987.

[4] N. Ayache and O. D. Faugeras, "Maintaining Representations of the Environment of a Mobile Robot," *IEEE Transactions on Robotics and Automation*, Vol. RA-5, pp. 804–819, Dec. 1989.

[5] D. H. Ballard and O. A. Kimball, "Rigid Body Motion from Depth and Optical Flow," *Comput. Vision, Graph. and Image Processing*, Vol. 22, pp. 95–115, Apr. 1983.

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

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

[8] S. D. Blostein and T. S. Huang, "Error analysis in stereo determination of 3-D point positions," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. 9, pp. 752–765, Nov. 1987.

[9] 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.

[10] 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*, Vol. AES-26, pp. 639–656, July 1990.

[11] T. J. Broida and R. Chellappa, "Estimation of Object Motion Parameters from Noisy Images," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. PAMI-8, pp. 90–99, Jan. 1986.

[12] T. J. Broida and R. Chellappa, "Kinematics and Structure of a Rigid Object from a Sequence of Noisy Images," in *Proc. of IEEE Workshop on Motion: Representation and Analysis*, May 1986.

[13] T. J. Broida and R. Chellappa, "Kinematics of a Rigid Object from a Sequence of Noisy Images: a Batch Approach," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 176–182, June 1986.

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

[15] 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.*, June 1991.

[16] J. Buhmann *et al.*, "Object Recognition in the Dynamic Link Architecture - Parallel Implementation on a Transputer Network," in *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence* (B. Kosko, ed.), Prentice-Hall, 1990.

[17] J. Buhmann, J. Lange, and C. von der Malsburg, "Distortion Invariant Object Recognition by Matching Hierarchically Labelled Graphs," in *International Joint Conf. on Neural Networks*, (Washington D.C.), June 1989.

[18] S. Chandrashekhar and R. Chellappa, "A two-step approach to passive navigation using a monocular image sequence," USC-SIPI Technical Report 170, University of Southern California, Los Angeles, CA, Feb. 1991.

[19] 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, Dec. 1988.

[20] N. Cui, J. Weng, and P. Cohen, "Extended Structure and motion analysis from monocular image sequences," in *Third International Conf. on Computer Vision*, (Osaka, Japan), pp. 222–229, Dec. 1990.

[21] J. G. Daugman, "Relaxation neural network for non-orthogonal image transforms," in *Proc. Int. Conf. on Neural Networks*, vol. 1, (San Diego, CA), pp. 547–560, June 1988.

[22] R. Deriche, "Optimal Edge Detection Using Recursive Filtering," in *First International Conf. on Computer Vision*, pp. 501–505, June 1987.

[23] E. D. Dickmanns, "An Integrated Approach to Feature Based Dynamic Vision," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 820–825, June 1988.

[24] E. D. Dickmanns and V. Graefe, "Dynamic Monocular Machine Vision," *Machine Vision and Applications*, Vol. 1, No. 4, pp. 233–240, 1988.

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

[26] R. Dutta and M. A. Snyder, "Robustness of correspondence-based structure from motion," in *Proceedings of the Image Understanding Workshop*, pp. 428–432, DARPA, Sept. 1990.

[27] J.-Q. Fang and T. S. Huang, "Some Experiments on Estimating the 3-D Motion Parameters of a Rigid Body from Two Consecutive Image Frames," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. PAMI-6, pp. 545–554, Sept. 1984.

[28] O. Faugeras, "On the motion of 3D curves and its relationship to optical flow," Tech. Rep. 1183, INRIA, Sophia-Antipolis, Mar. 1990.

[29] O. D. Faugeras, F. Lustman, and G. Toscani, "Motion and structure from point and line matches," in *First International Conf. on Computer Vision*, pp. 25–34, June 1987.

[30] W. Franzen, *Structure from chronogeneous motion*, Ph.D. Thesis, University of Southern California, Jan. 1991.

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

[32] S. Ganapathy, "Camera Location Determination Problem," tech. rep., AT & T Bell Laboratories, Holmdel, New Jersey, Nov. 1984.

[33] S. Ganapathy, "Decomposition of transfomation matrices for robot vision," *Pattern Recognition Letters*, Vol. 2, pp. 401–412, Dec. 1989.

[34] D. B. Gennery, "Tracking Known 3-D Objects," in *Proc. National Conf. on Artificial Intell.*, pp. 13–17, Aug. 1982.

[35] G. Giraudon, "Chainage rapide sur des images de contour," tech. rep., INRIA, Sophia-Antipolis, 1987.

[36] W. E. L. Grimson, *From Images to Surfaces*, Cambridge, Massachusetts: MIT Press, 1985.

[37] A. Grossman and J. Morlet, "Decomposition of functions into wavelets of constant shape, and related transforms," in *Mathematics and Physics, Lecture on Recent Results*, Singapore: World Scientific Publishing, 1985.

[38] J. Heel, "Direct estimation of structure and motion from multiple frames," AI memo 1190, MIT Artificial Intelligence Laboratory, Mar. 1990.

[39] E. C. Hildreth, "Computations Underlying the Measurement of Visual Motion," *Artificial Intelligence*, Vol. 23, pp. 309–354, Aug. 1984.

[40] I.J.Cox and G.T.Wilfong, eds., *Autonomous Robot Vehicles*, Springer-Verlag, 1990.

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

[42] D. J. Kriegman, E. Triendl, and T. O. Binford, "Stereo Vision and Navigation in Buildings for Mobile Robots," *IEEE Transactions on Robotics and Automation*, Vol. RA-5, pp. 792–803, Dec. 1989.

[43] R. Kumar and A. R. Hanson, "Robust estimation of camera location and orientation from noisy data with outliers," in *IEEE Workshop on the Interpretation of 3-D scenes*, (Austin, Texas), Nov. 1989.

[44] R. Kumar and A. R. Hanson, "Sensitivity of the pose refinement problem to accurate estimation of camera parameters," in *Third International Conf. on Computer Vision*, (Osaka, Japan), Dec. 1990.

[45] R. V. R. 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*, (San Diego, CA), pp. 136–143, June 1989.

[46] P. Limozin-Long, "Stereo Matching Using Contextual Line Region Primitives," in *IEEE International Conf. on Pattern Recognition*, (Paris), Oct. 1986.

[47] B. S. Manjunath and R. Chellappa, "A unified approach to boundary perception : edges, textures and illusory contours," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, (Maui,Hawaii), June 1991.

[48] R. Manmatha *et al.*, "Issues in extracting motion parameters and depth from approximate translational motion," in *IEEE Workshop on Visual Motion*, (Irvine, California), Mar. 1989.

[49] D. Marr and T. Poggio, "A theory of Human Stereo Vision," *Proceedings of the Royal Society of London*, Vol. B 204, pp. 301–328, 1979.

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

[51] J. Mayhew and J. Frisby, "Psychophysical and computational studies towards a theory of human stereopsis," *Artificial Intelligence*, Vol. 17, pp. 349–385, 1981.

[52] M.C.Morrone and D.C.Burr, "Feature Detection in human vision : A phase-dependent energy model," *Proceedings of the Royal Society of London*, Vol. B 235, 1988.

[53] G. Medioni and R. Nevatia, "Segment-Based Stereo Matching," *Computer Vision, Graphics and Image Processing*, Vol. 31, pp. 2–18, July 1985.

[54] A. Meygret and M. Thonnat, "Segmentation of optical flow and 3D data for the interpretation of mobile objects," in *Third International Conf. on Computer Vision*, (Osaka, Japan), Dec. 1990.

[55] A. Meygret, M. Thonnat, and M. Berthod, "A Pyramidal Stereovision Algorithm Based On Contour Chain Points," in *European Conf. on Computer Vision*, Apr. 1990.

[56] M. Porat and Y. A. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. PAMI-10, pp. 452–468, July 1988.

[57] J. Roach and J. Aggarwal, "Determining the Movement of Objects from a Sequence of Images," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. PAMI-2, pp. 554–562, Nov. 1980.

[58] 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, Sept. 1990.

[59] J. J. Rodriguez and J. Aggarwal, "Stochastic Analysis of Stereo Quantization Error," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. PAMI-12, pp. 467–470, May 1990.

[60] H. Shariat, "The Motion Problem: How to Use More Than Two Frames," Tech. Rep. IRIS-202, University of Southern California, Institute for Robotics and Intelligent Systems, Los Angeles, CA, Oct. 1986.

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

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

[63] R. Y. Tsai and T. S. Huang, "Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, pp. 1147–1152, Dec. 1981.

[64] R. Y. Tsai and T. S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. PAMI-6, pp. 13–27, Jan. 1984.

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

[66] 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.*, Vol. PAMI-9, pp. 370–389, May 1987.

[67] J. Weng, T. S. Huang, and N. Ahuja, "Motion and Structure from Two Perspective Views : Algorithms, Error Analysis, and Error Estimation," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. PAMI-11, pp. 451–476, May 1989.

[68] J. Weng, T. S. Huang, and N. Ahuja, "Motion estimation from images: image matching, parameter estimation and intrinsic stability," in *IEEE Workshop on Visual Motion*, (Irvine, California), pp. 359–366, Mar. 1989.

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

[70] 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.*, Vol. PAMI-8, pp. 464–471, July 1986.

[71] G. S. Young, *3-D motion estimation from a sequence of noisy stereo images*, Ph.D. Thesis, University of Southern California, Aug. 1991.

[72] G. S. Young and R. Chellappa, "3-D Motion Estimation Using a Sequence of Noisy Stereo Images," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, (Ann Arbor, MI), June 1988.

[73] 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.*, Vol. PAMI-12, pp. 735–759, July 1990.

[74] G. S. Young and R. Chellappa, "Statistical Analysis of Inherent Ambiguities in Recovering 3-D Motion from a Noisy Flow Field," in *IEEE International Conf. on Acoustics, Speech, and Signal Processing*, Apr. 1990.

[75] Z. Zhang and O. D. Faugeras, "Tracking and motion estimation in a sequence of stereo frames," in *European Conf. on Artificial Intelligence*, Aug. 1990.

[76] Q. Zheng and R. Chellappa, "A novel image registration algorithm," in *Proceedings of the Image Understanding Workshop*, (San Diego, CA), pp. 899–909, DARPA, Jan. 1992.