Efficient Recursive Factorization Methods for Determining Structure from Motion

Yanhua Li, B.Comp.Sci.

Department of Computer Science
The University of Adelaide

Cooperative Research Center for
Sensor Signal and Information Processing

A thesis submitted for the degree of

Doctor of Philosophy

in the Department of Computer Science

The University of Adelaide

July 2000

Declaration

This work contains no material that has been accepted for the award of any other degree or diploma in any university or other tertiary institution. To the best of my knowledge and belief, it contains no material previously published or written by any other person, except where due reference is made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

Yanhua Li

July 2000

Acknowledgments

I would like to express my sincerest gratitude and appreciation to my supervisor, Professor Mike Brooks, whose excellent guidance and advice throughout the course of my research work, whose enthusiasm and support in the project contributed to a great extent to my progress and completion of this thesis. His understanding and kindness are also sincerely appreciated.

I would like to thank Darren Gawley for the use of his CATE [23] system to track feature points in the real image sequences. Thanks to Anton van den Hengel and Professor Wojciech Chojnacki for their assistance and friendship during my study in the University of Adelaide.

I am thankful to the Cooperative Research Center for Sensor Signal and Information Processing (CSSIP) for providing me computer access during my study and financial support via a CSSIP Supplementary Scholarship. I also acknowledge the financial support provided by the Australian Government in the form of an Australian Postgraduate Award.

My special thanks go to my husband, Yongfeng for his insightful advice and encouragement to my research work, and to my beloved son, George, whose growth has brought me happiness and motivation to my study. My last but not least gratitude is attributed to my parents for their love and encouragement.

Abstract

This thesis addresses the *structure from motion* (SFM) problem in computer vision. We study recursive algorithms for efficient shape and motion recovery at each frame of a sequence of images. In real-time applications where image data is extensive, efficiency of an SFM method becomes very important for estimating shape and motion at each frame. The proposed recursive method in this thesis improves the efficiency, both in computational cost and in storage, of a class of innovative SFM methods — the *factorization methods* (FMs).

Our work in this thesis may be viewed as an extension of the original [71], the sequential [55] and the paraperspective [60] factorization methods. A critical aspect of these factorization approaches is the estimation of the *shape space*, and their computational complexity is dominated by their shape space computing algorithms. If P object feature points are tracked through a sequence of F frames, the shape space updating complexity at each frame is $O(P)$ in the recursive least-squares (RLS) method proposed in this thesis, while that in the sequential FM is $O(P^2)$. In contrast, the batch-mode original and paraperspective FMs, which are not intended to be used frame by frame, compute the shape space at a cost of $O(FP^2)$ after all F frames are tracked. Further, the RLS shape space updating algorithm is an adaptive data driven algorithm. Hence it does not require storage of a large measurement or covariance matrix, while other FMs usually do.

The proposed recursive method uses the general affine camera model, while the original and the sequential FMs assumed an orthographic model. Like the paraperspective FM, we give Euclidean shape and motion recovery from the estimated shape space under one of the three specific affine camera models — orthography, weak perspective and paraperspective, in order to apply the recursive method to a wider camera motion range.

We also extend the recursive method to accommodate the situation in which some feature points are occluded or leave the field of view during the sequence. The extended recursive method still retains the low computational complexity of $O(P)$ and is simpler than the occlusion solutions proposed in other FMs.

Experiments with real and synthetic image sequences confirm the recursive method's low computational complexity and good performance and indicate that it is well suited to real-time applications.

Publications

The work shown in this thesis was partially published in the following papers. Aspects of the introduction and review sections of the papers appear in Chapters 1 and 2. Chapter 3 is based on the MSE formulation, and Chapter 4 is based on the RLS algorithm, of both papers. Chapters 5, 6 are based on the relevant sections of [2]. Also some of the experiment results in the thesis are taken from [1].

1. Y. Li and M.J. Brooks, "An efficient recursive factorization method for determining structure from motion," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 138-143, Fort Collins, Colorado, Jun. 1999.

2. Y. Li and M.J. Brooks, "Efficient shape and motion recovery from image sequences using a recursive factorization method," Technical Report, Dept. Computer Science, Univ. of Adelaide, Sept. 1999.

Contents

List of Figures

Glossary

Abbreviations

2D	:	two dimension
3D	:	three dimension
eq.	:	equation
EVD	:	eigenvalue decomposition
FM	:	factorization method
GAP	:	general affine projection
LLS	:	linear least squares
MSE	:	mean square errors
PAST	:	projection approximation subspace tracking
RLS	:	recursive least squares
SST	:	Subspace Tracking
SFM	:	structure from motion
SVD	:	singular value decomposition

Notation

$\mathbf{A}^T$	:	transpose of $\mathbf{A}$
$\mathbf{A}^*$	:	complex conjugate of $\mathbf{A}$
$\mathbf{A}^H$	:	complex conjugate transpose of $\mathbf{A}$
$range(\mathbf{A})$	:	column space of $\mathbf{A}$
$rank(\mathbf{A})$	:	rank of $\mathbf{A}$
$tr\{\mathbf{A}\}$	:	trace of $\mathbf{A}$
$E\{\mathbf{A}\}$	:	expectation of $\mathbf{A}$
$\|\|\mathbf{A}\|\|_F$	:	Frobenius norm of $\mathbf{A}$
$\|\|\mathbf{a}\|\|$	:	norm of vector $\mathbf{a}$
$\hat{\mathbf{A}}$	:	estimate of $\mathbf{A}$
$\mathbf{I}_n$	:	$n \times n$ identity matrix
$diag(d_1, \cdots, d_n)$	:	$n \times n$ diagonal matrix whose i-th diagonal entry equals d_i with $i = 1, \cdots, n$
σ^2	:	noise variance
$dist(S_1, S_2)$	:	subspace distance between subspaces S_1 and S_2
$\mathbf{0}_{m \times n}$	:	$m \times n$ null matrix
$\mathbf{1}_{m \times n}$	:	$m \times n$ matrix whose entries are all 1
$\Re^{m \times n}$	:	$m \times n$ matrix whose entries are all real numbers
$d\Phi(\mathbf{X})$	:	differential of function $\Phi(\mathbf{X})$ with respect to $\mathbf{X}$
$\nabla\Phi(\mathbf{X})$	:	gradient of $\Phi(\mathbf{X})$ with respect to $\mathbf{X}$

Chapter 1

Introduction

The *structure from motion* (SFM) problem — recovering object shape and object or camera motion from a given sequence of images — is one of the main problems in computer vision. It has wide applicability in many areas such as robot navigation, non-tactile parts inspection, manipulation, and CAD. In the last decade, a class of innovative methods called factorization methods (FMs) has been proposed and has attracted much attention in the vision community. The FMs have advantages over the traditional nonlinear methods by their simplicity, accuracy and robustness. This thesis studies some novel recursive FMs based on several previously presented FMs. Our new methods can estimate 3D shape and motion at each frame efficiently by using the recursive least squares (RLS) technique. This study is therefore useful for real-time applications.

1.1 Structure from Motion

Given a sequence of images taken by a moving camera viewing a three-dimensional (3D) scene, the question arises: can we extract simultaneously the 3D shape of the scene and the relative motion between the camera and the scene? This question involves two fundamental problems in computer vision. The first is called *correspondence*, which decides how points (or other features) "flow" between consecutive images in the sequence. The second is *structure-from-motion*, which determines the 3D shape and motion from matched points (or other features) in the images. Therefore, solving the SFM problem assumes the correspondence problem has first been solved; that is, the feature points (or other features) have already been tracked throughout the image sequence.

The SFM problem has been extensively studied in the last two decades. Approaches to this problem differ in the numbers of images being processed, the camera projection models they use, and the assumptions they make about the objects (or the scene) and the motion. They may require different types of tracked features as input primitives, and produce different forms of output structure and motion.

Early research on this subject [46, 75] focused on computing structure and motion from a small set of points matched in two frames. Longuet-Higgins [46] presented a closed-form solution to the problem using eight corresponding points in two frames. Tsai and Huang [75] present similar work, but in addition consider the uniqueness issues in the determination of the motion parameters. Others have extended the essential parameter approach to lines [19, 64], performed more detailed error analyses [81, 83, 49], and developed non-linear least squares algorithms for the two-frame problem [82].

Recent research focuses on estimating shape and motion from longer image sequences

[41, 17, 16, 71, 14, 60, 55, 44, 18] by taking advantage of the redundancy of measurements to obtain better accuracy. Among these long-frame approaches, some are based on batch mode and process the whole sequence only after all the frames are available [17, 41, 71, 14, 60], while others use incremental methods which process a new frame on-line as it becomes available [29, 51, 16, 6, 55, 44, 66, 31, 48].

Although there are some SFM methods that address the most general problems in real applications, many existing approaches are based on various assumptions about the scene, the motion, and the imaging model. For example, lots of the existing work assumes that the scene contains only a single rigid object [74, 75, 71, 55, 60, 44], while others process multiple independently moving objects in the scene [79, 5, 14, 15]. As to the motion assumptions, there are assumptions of pure translation [51, 29], smooth motion [6], constant motion [17], and one of the motion components being known [70], etc.

SFM approaches also differ in the camera projection model they assume and the form of output (recovered structure and motion) they produce. In Ullman's original proof of existence of a solution [77, 78] for the structure from motion problem, the orthographic camera model was used under a world coordinate system. However, after that, most computer vision researchers prefer using the perspective camera model and a camera-centered coordinate system [75, 4, 64, 29, 51, 6]. At the same time, although some previous work employed a calibrated camera to recover 3D Euclidean structure, in more recent research, emphasis has been on the use of uncalibrated cameras [25, 8] to compute projective structure (Euclidean structure up to a projective transformation) under a perspective camera model [20, 68, 26, 73, 3, 30, 76, 32, 48], or an affine structure (Euclidean structure up to an affine transformation) [35, 62, 36, 63] under an affine camera model (a linear approximation to perspective projection). In quite a lot of applications, projective structure or affine structure provides enough information. In the case that Euclidean structure and motion recovery is required, projective structure or affine structure can be transformed to Euclidean structure when suitable constraints on the camera, the motion or the scene are given [3, 76, 86].

On the other hand, most SFM approaches are feature based. They assume that there exists a set of features on the objects that are trackable throughout the image sequence and the correspondence problem of extracting features and establishing the required correspondences between images have been accomplished beforehand. This assumption enables the SFM methods to focus on geometric considerations and ignore all of the image processing tasks. Object shape is then represented by the 3D locations of its features with respect to a reference system. Object (or camera) motion is represented by the rotation and translation of the object (or camera) with respect to this reference system. The image sequence means simply the coordinates of the projected features in the images. Essentially, three types of features have been used in solving SFM problem, namely points, straight lines and line segments. Points are the most frequently used features in the SFM methods [46, 75, 20, 71, 60, 44]. Lines and line segments are also used in some work [64, 68, 26, 69, 62] because they are easier to track than points and provide valuable constraints for shape recovery. However, the reconstruction process using lines is more sensitive to noise and being trapped by local minima than that using point features. As a compromise, some algorithms use both point and line features [19, 27, 56, 59].

1.2 Factorization Methods and our Motivation

Among the numerous approaches to SFM over the last two decades, the *factorization* method presented by Tomasi and Kanade [70, 71] is both elegant and useful because of its simplicity, accuracy and robustness. In the method, under the assumption of an orthographic projection model and using an object centered world coordinate system, the SFM problem is formulated as a simple bilinear relationship between the image measurements and the motion, shape parameters. The powerful singular value decomposition (SVD) technique is then used to compute shape and motion recovery after all the frames are tracked. Due to the numerical stability of SVD and the even treatment of every frame and every point in the algorithm, the Tomasi-Kanade factorization method is normally more stable than other nonlinear SFM algorithms. At the same time, it permits a large number of points and frames to be processed so that the redundancy of the measurements in the image sequence can reduce the effects of noise and provide more accurate and robust results.

The emergence of the Tomasi-Kanade original factorization method has inspired extension and generalization in several research directions. For example, the projection model was extended from orthography to paraperspective [60, 61], and then to perspective [73, 67]; the features used were extended from points to lines [62, 63, 36, 59], line segments and planes [56]; the number of objects was generalized from one to multiple independently moving objects [14, 15]; and the original batch-mode method was extended to sequential and recursive algorithms [55, 44, 31, 32, 66, 48].

The key feature of the class of FMs is their bilinear formulation which relates the image measurements with the shape and motion parameters [37]. This is captured schematically as:

$$\textbf{image measurements} \Leftrightarrow \textbf{motion} \times \textbf{shape}$$

The images are assumed to be due to two factors: the relative *motion* between the camera and the object, and the object *shape*. They are composed in a bilinear form such that if either motion or shape is constant, then the image measurements will be a linear function of the other. The motion parameters refer to the extrinsic parameters given by the relative orientation and translation between the object and the camera, as well as the intrinsic parameters that are interval to the camera. These parameters may vary frame by frame in the sequence, but are the same for all features in a single frame. Object shape is represented by the 3D locations of the features on the object (usually on the object surface) relative to each other. Under an object-centered world coordinate system, which is used in the FMs, the shape of a rigid object (assumed in the FMs) is constant over the sequence.

The general affine projection (GAP), a linear approximation to perspective projection, can be used to model the camera projection when the object size is small compared to its depth from the camera. FMs that use the GAP model generally consist of two steps. The first step computes affine shape and motion. From the subspace viewpoint, this step can be viewed as the estimation of a subspace associated with object shape, namely *shape space* (see Chapter 2). It is performed via a number of techniques including SVD. After shape space is estimated, affine shape and motion can be derived directly from it. The second step recovers Euclidean shape and motion from the results of the first step. It is implemented by using an appropriate affine model (orthography, paraperspective or weak perspective) to determine the affine transformation matrix.

Although the original FM is useful and powerful, it is not applicable to real-time applications for the following reasons. First, it is a batch-type method and so shape and motion can only be recovered after all the frames in a sequence are given. Second, the key SVD procedure in the method requires $O(FP^2)$ operations to compute the shape space from P feature points in F frames. It is very expensive when P and F are large. Third, it needs to store a $2F \times P$ measurement matrix (its size therefore increasing with the number of frames) prior to computing structure from motion. To overcome these drawbacks, Morita and Kanade developed a sequential FM [55] that enables shape and motion to be estimated at each frame. Under their scheme, shape space is updated with each incoming frame at a cost of $O(P^2)$ via a power iteration technique. A $P \times P$ covariance matrix is updated and stored as part of the processing of each frame. However, in the applications where P (the number of features) is large, the shape space updating cost of the sequential method is still very high and the storage of a large covariance matrix is also required. Therefore, investigating some more efficient (both in computation and in storage) shape space updating algorithms so as to improve the efficiency and applicability of the factorization methods, is of practical importance in real-time applications. This is the main motivation of this thesis.

Estimating shape space at each frame is actually a subspace tracking task. There are two classes of techniques for deriving subspace tracking algorithms. The first class consists of using standard algorithms of numerical analysis to compute an exact or approximate subspace decomposition of a time-varying sample covariance matrix or a corresponding data matrix when a new column or row is added. The second class is based on formulation of a cost function for optimization in terms of a weight matrix. The weight matrix yields useful subspace information at its optimum. The traditional SVD of a data matrix (used by the batch-mode FMs) and EVD or power method of a covariance matrix (used by the sequential FM) belong to the first class. Since the first class algorithms are normally computationally expensive to update the subspace when new data arrives, and they require storage of a large data or covariance matrix, the SVD or EVD is not well suited for real-time subspace tracking. The second class of techniques usually involves two steps for deriving subspace tracking algorithms. The first step is to formulate the determination of the desired subspace as the optimization of a certain cost function in terms of a weight matrix. Various cost functions have been proposed, including the mean-square errors (MSE) minimization [85], the constrained output variance maximization [40], and the novel information criterion maximization [53]. While the second step is to derive data-driven recursive algorithms, where the optimization is carried out adaptively via different stochastic search techniques, such as the conjugate gradient method [22], the Gauss-Newton method [50], and the recursive least-squares (RLS) method [28].

On the other hand, the assumption of the orthographic camera model in the original and the sequential FMs limits the range of motions they can accommodate. Orthography can only be used to model image projection when the object size and the camera translation is small compared to the object's depth from the camera. If the camera motion involves significant translation, then using orthography-based methods will cause deformed shape and motion recovery because several perspective effects cannot be ignored. Instead, other forms of affine camera can be used in such situations to closely approximate perspective projection. For example, weak perspective projection, also called scaled orthography, can be used instead of orthography to account for the perspective scaling effect caused by the depth changes when the camera has large translation along depth direction between images. Also paraperspective projection, which models the perspective position effect as well as the scaling effect, can replace orthography when the camera motion involves significant translation across the scene. Since the bilinear form of the factorization methods holds under the GAP model (see Chapter 3), extension of the orthography-based factorization methods to weak perspective and paraperspective model is possible. Although Poelman and Kanade have presented a FM under these two models [60], it still uses batch-type SVD to estimate the shape space like the original method. Thus it is essentially a batch method and not applicable to real-time applications.

In this thesis, we overcome some drawbacks of the previously described FMs and propose a new efficient recursive FM under the GAP model. In this new method, the shape space estimation is formulated as the minimization of an MSE cost function. We show that this cost function provides the desired affine shape space at its global minimum. By using the standard RLS technique, an RLS algorithm is developed for the MSE minimization that uses as input the coordinate vectors of feature points at each image frame. Shape space is computed with complexity $O(P)$ per frame. Since it is a data-driven algorithm, the method proceeds without computing and storing any large matrices. After the shape space is estimated, Euclidean shape and motion can be recovered from it under one of the three forms of affine camera models — orthography, weak perspective, and paraperspective — according to the motion pattern of the camera.

Also the occlusion issue will be addressed. Some feature points cannot be tracked in some frames of the sequence because they are occluded or leave the field of view. The RLS shape space updating algorithm is extended to accommodate the occlusion case. We show that the computational cost for shape space updating in the extended algorithm is still $O(P)$.

The performance of the new recursive method and a comparison with other FMs are also covered in the thesis. We show that the recursive method attains the same accuracy as those FMs, while the computation time is far less. Experiments with synthetic data and real image sequences confirm the recursive method's accuracy and efficiency. Euclidean reconstruction results of orthography, weak perspective, and paraperspective camera models are compared for a wide range of camera/object motions.

1.3 Outline of the Thesis

Chapter 2 serves to review several related factorization methods, including the original Tomasi-Kanade method, the sequential Morita-Kanade method, and the paraperspective Poelman-Kanade method. The bilinear formulation in these methods is reviewed along with the techniques they used to estimate the *shape space*, whose definition is given in the chapter. The first step of these FMs (estimation of the affine structure) is characterized as shape space tracking. The second step (Euclidean reconstruction from the affine structure under the camera models assumed by individual FM) is then discussed in detail.

Chapter 3 proposes an MSE minimization formulation for the shape space tracking under the GAP model. The GAP model and the bilinear formulation is first introduced. Then an MSE cost function is formulated for tracking the shape space. Finally the properties of the MSE cost function are studied.

Chapter 4 first reviews the standard RLS technique and the matrix inversion lemma, then derives a recursive algorithm for the MSE minimization, i.e., shape space estimation, by using the RLS technique. Initial conditions of the RLS shape space tracking algorithm are discussed. It is shown that the computational complexity of updating shape space at each frame is only $O(P)$ for the RLS method. Advantages of the RLS algorithm over the shape space computing algorithms in the other FMs are discussed. Simulation results on synthetic data compare the performance of the RLS algorithm with that of the original and sequential FMs. The actual processing time of the three shape space algorithms is also compared.

Chapter 5 gives Euclidean reconstruction of object shape and camera motion under orthographic, weak perspective, and paraperspective projection models respectively. Metric constraint equations, in terms of the affine transformation matrix under each model, are derived and solved using a linear least squares (LLS) technique. Then the 3D shape and motion are recovered from the estimated shape space by the affine transformation. Experiments on synthetic data show the performance of Euclidean reconstruction under each model over a wide range of translation and initial depth conditions.

Real image reconstruction also confirms the accuracy of the recursive method.

Chapter 6 extends the RLS algorithm to accommodate the occlusion situation. The image locations of the occluded points in a frame are first estimated from the other tracked points in the same frame and the estimated shape at the previous frame. Then the estimated and the tracked image coordinates are used together as the input to the RLS shape space updating algorithm. The computational complexity of the extended RLS algorithm is shown to be still $O(P)$. A comparison with those occlusion processing techniques presented in the original and the paraperspective FMs shows the simplicity and efficiency of the proposed method. Also simulation results demonstrate its good performance.

Chapter 7 draws conclusions and provides suggestions for future research.

1.4 Contributions

This thesis makes the following contributions:

1. Enabling 3D shape and motion to be recovered recursively at each frame.
2. A new interpretation of the first step of the FMs as the *shape space* tracking task.
3. Proposal of a new MSE formulation for the shape space estimation under GAP model.
4. Development of an efficient RLS algorithm for shape space updating at each frame.
5. Significant reduction of the shape space updating cost to $O(P)$ per frame.

6. No storage requirement for a large matrix in the RLS algorithm.

7. Recursive recovery of 3D shape and motion at each frame under the orthographic, weak perspective and paraperspective projection models, respectively, to accommodate a wide range of camera motion.

8. Proposal of a new efficient occlusion processing method and incorporation into the RLS algorithm.

Chapter 2

Review of Factorization Methods

This chapter reviews the three factorization methods that are relevant to our work in this thesis, and uses the novel notion of *shape space* tracking. A definition of *shape space* is presented and the first step of these FMs is interpreted as shape space estimation. The techniques used for this shape space estimation in each method are introduced. The second step of Euclidean reconstruction from the shape space is also reviewed.

2.1 The Original Method

The factorization method was first introduced by Tomasi and Kanade in [70] and [71] by assuming an orthographic projection model and using an object-centered world coordinate system as shown in Figure 2.1. In the method, singular value decomposition (SVD) is used to compute shape space at a complexity $O(FP^2)$ after P feature points have been tracked over a sequence of F image frames. Euclidean shape and motion are then recovered from the shape space without computing camera-centered depth. When the object is far away from the camera, this method generally produces robust and accurate results.

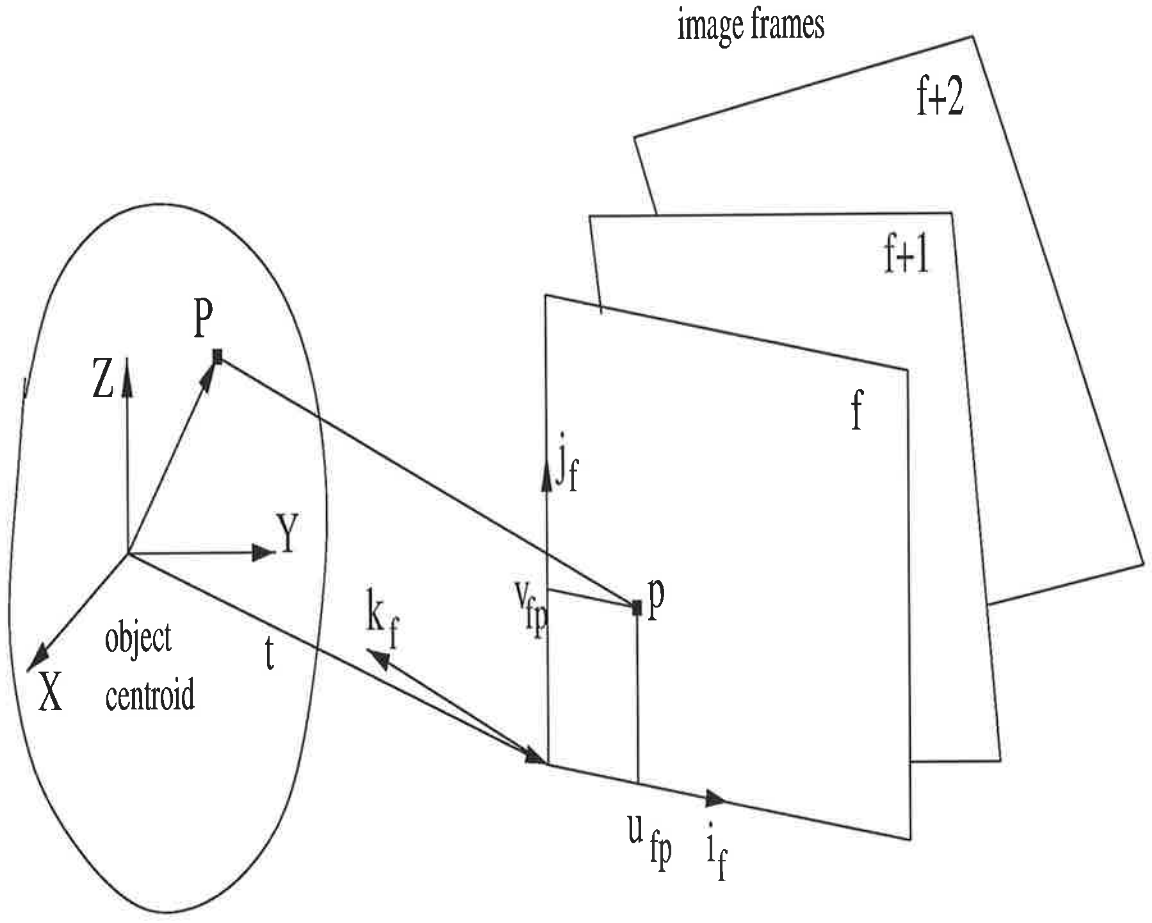

Figure 2.1: The object-centered orthographic model in the original FM

2.1.1 The Bilinear Formulation and the Rank Theorem

Assume that there is only one rigid object in the scene, and P feature points are tracked through F image frames of a sequence. Let (u_{ij}, v_{ij}) denote the coordinates of the j-th image point in the i-th frame. Then the image coordinates of P feature points in F

frames form the following $2F \times P$ measurement matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{bmatrix} u_{11} & \cdots & u_{1P} \\ v_{11} & \cdots & v_{1P} \\ \vdots & \ddots & \vdots \\ u_{F1} & \cdots & u_{FP} \\ v_{F1} & \cdots & v_{FP} \end{bmatrix}. \tag{2.1}$$

Here, each column of $\mathbf{W}$ stores the image trajectory of one feature point over the whole sequence, while each pair of rows stores the coordinates of the P points in one of the frames.

Now all P points in each frame are registered by subtracting their image centroid's coordinates:

$$\begin{aligned} \tilde{u}_{ij} &= u_{ij} - a_i \\ \tilde{v}_{ij} &= v_{ij} - b_i \end{aligned} \qquad (i = 1, \cdots, F), \tag{2.2}$$

where (a_i, b_i) is the image centroid of the i-th frame given by

$$\begin{aligned} a_i &= \frac{1}{P}\sum_{j=1}^{P} u_{ij} \\ b_i &= \frac{1}{P}\sum_{j=1}^{P} v_{ij}. \end{aligned} \tag{2.3}$$

Combining eq. (2.2) for all P points in F frames into a single matrix equation, we obtain the registered measurement matrix $\tilde{\mathbf{W}}$. It is equivalent to registering the measurement matrix $\mathbf{W}$ by subtracting from each entry the mean of the entries in the same row, viz:

$$\tilde{\mathbf{W}} = \mathbf{W} - \begin{bmatrix} a_1 \\ b_1 \\ \vdots \\ a_F \\ b_F \end{bmatrix} \underbrace{[1 \cdots 1]}_{P}. \tag{2.4}$$

Assume that the origin of the world coordinate system resides at the centroid of P object feature points. Object shape may then be represented by the $3 \times P$ matrix

$$\mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_P], \tag{2.5}$$

where the 3-vector $\mathbf{s}_j$ describes the location of an object point (corresponding to the j-th image point) expressed with respect to the world coordinate system. We can also infer that

$$\frac{1}{P}\sum_{j=1}^{P} \mathbf{s}_j = \mathbf{0}. \tag{2.6}$$

Camera rotation over the sequence of F frames may be characterized by the $2F \times 3$ matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_1^T \\ \mathbf{j}_1^T \\ \vdots \\ \mathbf{i}_F^T \\ \mathbf{j}_F^T \end{bmatrix}. \tag{2.7}$$

Here, $\mathbf{i}_k$ and $\mathbf{j}_k$ are orthonormal 3-vectors specifying the orientation in the world coordinate system of the horizontal and vertical camera reference axes, respectively, of image frame k.

With the above assumptions and notations in place, the following equation holds under orthographic model:

$$\tilde{\mathbf{W}} = \mathbf{RS}. \tag{2.8}$$

This is the bilinear formulation of the original factorization method. We can see that under orthography, the image projection coordinates are just related to the camera rotation and the object 3D location, without any relation to the object depth and the camera translation. This reflects the nature of orthographic projection.

Since $\tilde{\mathbf{W}}$ is the product of two rank-3 matrices, it is subject to the *rank theorem* which states that:

> *Without noise, the rank of the registered measurement matrix $\tilde{\mathbf{W}}$ is at most 3; and when noise exists, the rank of $\tilde{\mathbf{W}}$ is approximately 3.*

The rank theorem reveals the high redundancy of information present in an image sequence, and permits a large number of points and frames to be processed to reduce the effects of noise. It implies that camera orientation and object shape can be robustly recovered by direct factorization of $\tilde{\mathbf{W}}$.

2.1.2 Shape Space Estimation via SVD Factorization

Assume that $2F \geq P$. Matrix $\tilde{\mathbf{W}} \in \Re^{2F\times P}$ may be decomposed using SVD to obtain

$$\tilde{\mathbf{W}} = \mathbf{O}_1 \mathbf{\Sigma} \mathbf{O}_2, \tag{2.9}$$

where $\mathbf{O}_1 \in \Re^{2F\times P}$ has orthonormal columns, $\mathbf{O}_2 \in \Re^{P\times P}$ is an orthogonal matrix, and $\mathbf{\Sigma} = diag(\sigma_1, \cdots, \sigma_P)$ with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_P > 0$. According to the rank theorem, in the absence of noise, $\tilde{\mathbf{W}}$ has at most 3 nonzero singular values. That is $\sigma_4 = \cdots = \sigma_P = 0$. However, a more realistic situation is that the measurements are corrupted by noise. Then there are more than 3 nonzero singular values. However, as long as the noise level is not very high, we have that $\sigma_3 \gg \sigma_4$. Thus the best rank-3 approximation $\hat{\mathbf{W}}$ of $\tilde{\mathbf{W}}$ is obtained by just considering the first 3 singular values of $\hat{\mathbf{W}}$:

$$\hat{\mathbf{W}} = \mathbf{U}\mathbf{\Sigma}'\mathbf{V}^T, \tag{2.10}$$

where $\mathbf{U}$, $\mathbf{V}$ are the first three columns of $\mathbf{O}_1$, $\mathbf{O}_2$, respectively, and $\mathbf{\Sigma}' = diag(\sigma_1, \sigma_2, \sigma_3)$. Now setting

$$\hat{\mathbf{R}} = \mathbf{U}[\mathbf{\Sigma}']^{1/2} \tag{2.11}$$
$$\hat{\mathbf{S}} = [\mathbf{\Sigma}']^{1/2}\mathbf{V}^T,$$

we have that

$$\hat{\mathbf{W}} = \hat{\mathbf{R}}\hat{\mathbf{S}}. \tag{2.12}$$

However, $\hat{\mathbf{R}}$ and $\hat{\mathbf{S}}$ are in general different from the desired camera rotation matrix $\mathbf{R}$ and object shape matrix $\mathbf{S}$ because the decomposition of $\hat{\mathbf{W}}$ in eq. (2.12) is not unique. For any 3×3 non-singular matrix $\mathbf{A}$, $\hat{\mathbf{W}} = \hat{\mathbf{R}}\mathbf{A}\mathbf{A}^{-1}\hat{\mathbf{S}} = \hat{\mathbf{R}}\hat{\mathbf{S}}$. In fact, $\hat{\mathbf{R}}$ and $\hat{\mathbf{S}}$ just determine the column space and the row space of $\hat{\mathbf{W}}$. Since it is also the case that $\hat{\mathbf{W}} = \mathbf{RS}$, we can infer that $\mathbf{R}$ and $\hat{\mathbf{R}}$ provide two bases for the column space of $\hat{\mathbf{W}}$, and $\mathbf{S}$ and $\hat{\mathbf{S}}$ provide two bases for the row space of $\hat{\mathbf{W}}$.

We then define $range(\mathbf{R})$, equivalent to the column space of $\hat{\mathbf{W}}$, to be the *motion space*; and $range(\mathbf{S}^T)$, equivalent to the row space of $\hat{\mathbf{W}}$, to be the *shape space*. If no noise exists, eq. (2.8) holds and $\tilde{\mathbf{W}} = \hat{\mathbf{W}}$. Then the motion space and the shape space correspond to the column space and row space of $\tilde{\mathbf{W}}$ respectively. From the above definition, it can be inferred that: (1) the dimension of each subspace is at most three; (2) both subspaces are dual in the sense that one can be computed from the other.

Eq. (2.10) reveals that $\mathbf{U}$ and $\mathbf{V}$ provide an orthonormal basis for the motion space and shape space respectively. The original factorization method readily obtains these bases via SVD.

It needs to be pointed out that motion space is only meaningful in a batch FM when all image frames in a sequence are processed at one time, while shape space is meaningful for every frame, and remains stationary throughout the sequence, under the object-centered world coordinate system. Therefore, in a sequential or recursive method, where each frame is processed separately, motion space is meaningless and only shape space is tracked.

2.1.3 Euclidean Shape and Motion Recovery

Now that $\mathbf{R}$ and $\hat{\mathbf{R}}$ are two different bases of motion space and $\mathbf{S}$ and $\hat{\mathbf{S}}$ are two different bases of shape space, there should exist a linear transformation between $\mathbf{R}$ and $\hat{\mathbf{R}}$, and $\mathbf{S}$ and $\hat{\mathbf{S}}$. More specifically, there exists a 3×3 invertible matrix $\mathbf{A}$ such that

$$\begin{aligned} \mathbf{R} &= \hat{\mathbf{R}}\mathbf{A} \\ \mathbf{S} &= \mathbf{A}^{-1}\hat{\mathbf{S}}. \end{aligned} \tag{2.13}$$

We also note that the camera orientation vectors $\mathbf{i}_k$, $\mathbf{j}_k$ in eq. (2.7) of frame k are mutually orthogonal unit vectors, and that they must satisfy

$$\left.\begin{aligned} \|\mathbf{i}_k\| =& \|\mathbf{j}_k\| = 1 \\ \mathbf{i}_k^T\mathbf{j}_k =& 0 \end{aligned}\right. \quad (k = 1, \cdots, F). \tag{2.14}$$

These *metric constraints* yield the $3F$ linear equations

$$\begin{aligned} \hat{\mathbf{i}}_k^T\mathbf{A}\mathbf{A}^T\hat{\mathbf{i}}_k =& 1 \\ \hat{\mathbf{j}}_k^T\mathbf{A}\mathbf{A}^T\hat{\mathbf{j}}_k =& 1 \quad (k = 1, \cdots, F) \\ \hat{\mathbf{i}}_k^T\mathbf{A}\mathbf{A}^T\hat{\mathbf{j}}_k =& 0 \end{aligned} \tag{2.15}$$

with respect to the 6 unknown parameters of $\mathbf{A}\mathbf{A}^T$. This is an over-determined linear problem that can be solved by using a linear least-squares (LLS) technique.

After $\mathbf{A}\mathbf{A}^T$ is determined, $\mathbf{A}$ can be obtained by eigendecomposition of $\mathbf{A}\mathbf{A}^T$. Such a solution of $\mathbf{A}$ is valid up to a rotation because the orientation of the world coordinate system is arbitrary. This arbitrariness can be removed by aligning the x-axis and y-axis of the world frame with those of the camera reference system in the first frame.

2.2 The Sequential Method

Since the original factorization method is not applicable to real-time applications, Morita and Kanade subsequently developed a sequential factorization method [54, 55] that produces estimates of shape and motion at each input frame under an orthographic projection model. In this method, shape space is updated with a cost of $O(P^2)$ at each frame via a power iteration method [24]. Euclidean shape and motion are then immediately determinable from the estimated shape space.

2.2.1 Formulation and Overview

Assume that P feature points are tracked in an F-frame sequence. Let (x_{ij}, y_{ij}) denote the registered image coordinates of the j-th feature point in the i-th frame with respect to the centroid of all the P feature points in the frame. The input to the sequential method is viewed as a vector time series containing feature image positions, so that at frame k, we have

$$\begin{aligned}\mathbf{x}_k &= [x_{k1}, x_{k2}, \cdots, x_{kP}]^T \\ \mathbf{y}_k &= [y_{k1}, y_{k2}, \cdots, y_{kP}]^T.\end{aligned} \tag{2.16}$$

Then the registered measurement matrix at frame k is expressed by

$$\tilde{\mathbf{W}}_k = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{y}_1^T \\ \vdots \\ \mathbf{x}_k^T \\ \mathbf{y}_k^T \end{bmatrix}, \quad k = 1, \cdots, F. \tag{2.17}$$

Now define a correlation matrix time series $\mathbf{Z}_k \in \Re^{\mathbf{P}\times\mathbf{P}}$ containing the image coordinates up to the k-th frame by

$$\mathbf{Z}_k = \sum_{i=1}^{k}(\mathbf{x}_i\mathbf{x}_i^T + \mathbf{y}_i\mathbf{y}_i^T) = \mathbf{Z}_{k-1} + \mathbf{x}_k\mathbf{x}_k^T + \mathbf{y}_k\mathbf{y}_k^T, \quad k = 1, \cdots, F. \tag{2.18}$$

It is easy to verify that

$$\mathbf{Z}_k = \tilde{\mathbf{W}}_k^T\tilde{\mathbf{W}}_k. \tag{2.19}$$

Since the rank of $\tilde{\mathbf{W}}_k$ is at most three, that of $\mathbf{Z}_k$ is also at most three. In the noisy case, if the best rank-3 estimate of $\tilde{\mathbf{W}}_k$ via SVD is

$$\hat{\mathbf{W}}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T, \tag{2.20}$$

where $\mathbf{U}_k \in \Re^{\mathbf{2k}\times\mathbf{3}}$ and $\mathbf{V}_k \in \Re^{\mathbf{P}\times\mathbf{3}}$ have orthonormal columns, and $\mathbf{\Sigma}_k = diag(\sigma_{k1}, \sigma_{k2}, \sigma_{k3})$ whose diagonal entries are the first three singular values of $\tilde{\mathbf{W}}_k$, then the best rank-3 estimate of $\mathbf{Z}_k$ is

$$\hat{\mathbf{Z}}_k = (\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T)^T\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T = \mathbf{V}_k\mathbf{\Sigma}_k^2\mathbf{V}_k^T. \tag{2.21}$$

This means that the first three principal eigenvectors of $\mathbf{Z}_k$ are equivalent to the first three right singular vectors of $\tilde{\mathbf{W}}_k$. Since $range(\mathbf{V}_k)$ is equivalent to the *shape space* (see Section 2.1.2), it is possible to estimate the shape space by computing the first three principal eigenvectors of $\mathbf{Z}_k$.

2.2.2 Sequential Shape Space Computation

To compute the three dominant eigenvectors of the correlation matrix $\mathbf{Z}_k$ that span the shape space, the sequential FM used a generalized power method — essentially an orthogonal power iteration method with the updating of $\mathbf{Z}_k$ by equation (2.18). The sequential shape space updating algorithm is given by:

Initialization:

$$\mathbf{Z}_0 = \mathbf{0}_{P\times P}$$
$$\mathbf{Q}_0, \overline{\mathbf{Q}}_0 \in \Re^{\mathbf{P}\times\mathbf{3}} \text{ with orthonormal columns}$$

Update equations (for $k = 1, \cdots, F$):

$$\mathbf{Z}_k = \mathbf{Z}_{k-1} + \mathbf{x}_k\mathbf{x}_k^T + \mathbf{y}_k\mathbf{y}_k^T \tag{2.22}$$
$$\mathbf{Y}_1 = \mathbf{Z}_k\mathbf{Q}_{k-1} \tag{2.23}$$
$$\mathbf{Q}_k\mathbf{R}_1 = \mathbf{Y}_1 \quad \text{(QR factorization)} \tag{2.24}$$
$$\mathbf{H}_k = \mathbf{Q}_k\mathbf{Q}_k^T \tag{2.25}$$
$$\mathbf{Y}_2 = \mathbf{H}_k\overline{\mathbf{Q}}_{k-1} \tag{2.26}$$
$$\overline{\mathbf{Q}}_k\mathbf{R}_2 = \mathbf{Y}_2 \quad \text{(QR factorization)} \tag{2.27}$$

In the above scheme, the matrix $\mathbf{Q}_k$ is expected to converge to the dominant eigenvectors $\mathbf{V}_k$ of $\mathbf{Z}_k$. However, even though $range(\mathbf{V}_k)$, the shape space, is stationary, $\mathbf{V}_k$ itself changes frame by frame since $\mathbf{Z}_k$ always incorporates new frame information as the number of frames increases. Therefore, $\mathbf{Q}_k$ does not converge to a stationary orthonormal basis though $range(\mathbf{Q}_k)$ converges to the stationary shape space. On the other hand, the desired 3D shape and camera orientation are determined from the shape space by a 3×3 metric transformation matrix $\mathbf{A}_k$. Different $\mathbf{Q}_k$ in each frame make $\mathbf{A}_k$ also change at each frame. To update $\mathbf{A}_k$ sequentially, it is necessary to compute a stationary basis for the shape space. In the above scheme, eqs. (2.25) - (2.27) update such a stationary basis $\overline{\mathbf{Q}}_k$ via QR factorization of the orthogonal projection matrix $\mathbf{H}_k$ of the shape space.

The updating of $\overline{\mathbf{Q}}_k$ at each frame involves the updating of $\mathbf{Z}_k$ and twice QR factorization. Its total computational complexity is $O(P^2)$ and specifically $26P^2 + 44P$.

2.2.3 Sequential Metric Transformation

After the stationary basis $\overline{\mathbf{Q}}_k$ of the shape space is obtained, camera coordinates $\hat{\mathbf{i}}_k$ and $\hat{\mathbf{j}}_k$ are computed as

$$\hat{\mathbf{i}}_k^T = \mathbf{x}_k^T \overline{\mathbf{Q}}_k, \quad \hat{\mathbf{j}}_k^T = \mathbf{y}_k^T \overline{\mathbf{Q}}_k. \tag{2.28}$$

Then the second step of the sequential FM is to determine a 3×3 non-singular matrix $\mathbf{A}_k$ that transforms $\hat{\mathbf{i}}_k$, $\hat{\mathbf{j}}_k$ and $\overline{\mathbf{Q}}_k$ into the desired 3D camera orientation vectors and the shape matrix as follows:

$$\mathbf{i}_k^T = \hat{\mathbf{i}}_k^T \mathbf{A}_k, \quad \mathbf{j}_k^T = \hat{\mathbf{j}}_k^T \mathbf{A}_k \tag{2.29}$$

$$\mathbf{S}_k = \mathbf{A}_k^{-1} \overline{\mathbf{Q}}_k^T. \tag{2.30}$$

Determination of $\mathbf{A}_k$ is carried out as in the original FM. By using the metric constraints of the camera orientation vectors $\mathbf{i}_k$, $\mathbf{j}_k$ of eq. (2.14) for each frame up to the k-th, $3k$ equations of the form eq. (2.15) with respect to the 6 unknown parameters of $\mathbf{A}_k\mathbf{A}_k^T$ are obtained. Solving this over-determined problem by LLS is shown in [54].

2.3 The Paraperspective Method

The original and the sequential FMs are based on the assumption of an orthographic projection model. However, the orthographic model limits the range of motions the methods can accommodate. Paraperspective projection, first introduced by Ohta [58], is a projection model that closely approximates perspective projection by modeling several effects not modeled under orthography, while retaining linear properties. Poelman and Kanade developed a FM [60, 61] by using this projection model. This method can be applied to a much wider range of motion scenarios, such as image sequences containing significant translation towards the camera or across the image plane.

2.3.1 Paraperspective Bilinear Formulation

The paraperspective projection of an object onto an image, illustrated in Figure 2.2, consists of the following two steps:

1. All the object points are projected onto the average depth plane, the plane passing through the object centroid and parallel to the image plane, by the rays parallel to the object centroid perspective projection ray CG.

2. The average depth plane is perspectively projected onto the real image plane. This is actually a uniform scaling of the average depth plane by the ratio of the camera focal length and the distance between the two planes.

As in the original FM, the origin of the world coordinate system is placed at the object centroid $\mathbf{G}$. Thus

$$\mathbf{G} = \frac{1}{P}\sum_{p=1}^{P} \mathbf{s}_p = \mathbf{0}. \qquad (2.31)$$

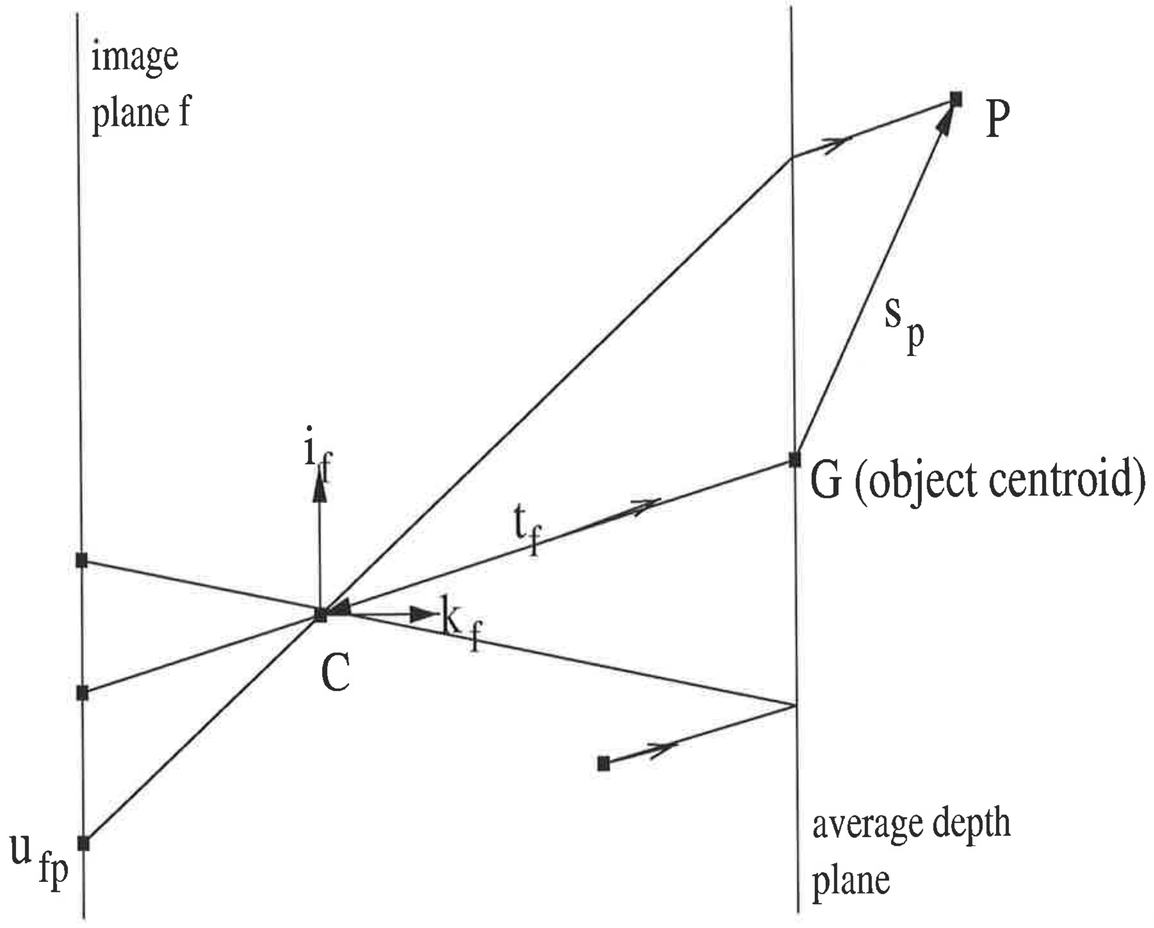

Figure 2.2: Two dimensional paraperspective projection model

From the definition of paraperspective projection described above and the geometric relations in Figure 2.2, it can be shown that the image coordinates of a feature point $\mathbf{p} = [u_{fp},\ v_{fp}]^T$ in frame f are

$$u_{fp} = \frac{1}{z_f}[(\mathbf{i}_f + \frac{\mathbf{i}_f \cdot \mathbf{t}_f}{z_f}\mathbf{k}_f) \cdot \mathbf{s}_p - (\mathbf{t}_f \cdot \mathbf{i}_f)] \tag{2.32}$$
$$v_{fp} = \frac{1}{z_f}[(\mathbf{j}_f + \frac{\mathbf{j}_f \cdot \mathbf{t}_f}{z_f}\mathbf{k}_f) \cdot \mathbf{s}_p - (\mathbf{t}_f \cdot \mathbf{j}_f)].$$

These equations can be rewritten as

$$u_{fp} = \mathbf{m}_f \cdot \mathbf{s}_p + a_f \tag{2.33}$$

$$v_{fp} = \mathbf{n}_f \cdot \mathbf{s}_p + b_f,$$

where

$$z_f = -\mathbf{t}_f \cdot \mathbf{k}_f \tag{2.34}$$
$$a_f = -\frac{\mathbf{t}_f \cdot \mathbf{i}_f}{z_f}, \quad b_f = -\frac{\mathbf{t}_f \cdot \mathbf{j}_f}{z_f} \tag{2.35}$$
$$\mathbf{m}_f = \frac{\mathbf{i}_f - a_f \mathbf{k}_f}{z_f}, \quad \mathbf{n}_f = \frac{\mathbf{j}_f - b_f \mathbf{k}_f}{z_f}. \tag{2.36}$$

If P feature points are tracked in F frames, we obtain the following matrix equation from eq. (2.33):

$$\begin{bmatrix} u_{11} & \cdots & u_{1P} \\ v_{11} & \cdots & v_{1P} \\ \vdots & \ddots & \vdots \\ u_{F1} & \cdots & u_{FP} \\ v_{F1} & \cdots & v_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{n}_1 \\ \vdots \\ \mathbf{m}_F \\ \mathbf{n}_F \end{bmatrix} [\mathbf{s}_1 \cdots \mathbf{s}_P] + \begin{bmatrix} a_1 \\ b_1 \\ \vdots \\ a_F \\ b_F \end{bmatrix} \underbrace{[1 \ \cdots \ 1]}_{P}. \tag{2.37}$$

Eq. (2.37) is equivalent to the short form

$$\mathbf{W} = \mathbf{MS} + \mathbf{T}\underbrace{[1 \ \cdots \ 1]}_{P}, \tag{2.38}$$

where $\mathbf{W}$ is the $2F \times P$ measurement matrix, $\mathbf{M}$ is the $2F \times 3$ motion matrix, $\mathbf{S}$ is the $3 \times P$ shape matrix, and $\mathbf{T}$ is the $2F \times 1$ translation vector.

Using eqs. (2.33) and (2.31), we obtain

$$\sum_{p=1}^{P} u_{fp} = \sum_{p=1}^{P} (\mathbf{m}_f \cdot \mathbf{s}_p + a_f) = \mathbf{m}_f \cdot \sum_{p=1}^{P} \mathbf{s}_p + P a_f = P a_f \tag{2.39}$$
$$\sum_{p=1}^{P} v_{fp} = \sum_{p=1}^{P} (\mathbf{n}_f \cdot \mathbf{s}_p + b_f) = \mathbf{n}_f \cdot \sum_{p=1}^{P} \mathbf{s}_p + P b_f = P b_f$$

Thus a_f and b_f, the elements of the translation vector $\mathbf{T}$, can be derived immediately from the image data as

$$a_f = \frac{1}{P} \sum_{p=1}^{P} u_{fp}, \quad b_f = \frac{1}{P} \sum_{p=1}^{P} v_{fp}, \quad f = 1, \cdots, F. \tag{2.40}$$

The registered measurement matrix $\tilde{\mathbf{W}}$ is then obtained by subtracting the known translation component $\mathbf{T}\underbrace{[1\cdots1]}_{P}$ from $\mathbf{W}$:

$$\tilde{\mathbf{W}} = \mathbf{W} - \mathbf{T}\underbrace{[1\ \cdots\ 1]}_{P} = \mathbf{MS}. \tag{2.41}$$

This has the same bilinear form as that in the original FM under orthography, and proves that the bilinear formulation and rank theorem under orthography also hold under the paraperspective model.

In the paraperspective method, Poelman and Kanade also used SVD to factorize $\tilde{\mathbf{W}}$, and then obtain its best rank-3 approximation $\hat{\mathbf{W}}$ and the two bases $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ for the shape space and motion space respectively by retaining only the largest 3 singular values. That is:

$$\begin{aligned}\hat{\mathbf{W}} &= \mathbf{U}_{2F\times3}\mathbf{\Sigma}_{3\times3}\mathbf{V}_{P\times3}{}^{T} = \hat{\mathbf{M}}\hat{\mathbf{S}}\\ \hat{\mathbf{M}} &= \mathbf{U\Sigma}, \quad \hat{\mathbf{S}} = \mathbf{V}^{T}.\end{aligned}$$

Thus the Poelman-Kanade FM is also a batch method and has the same drawbacks as the original FM — the complexity for shape space computation is high and the method is not applicable to real-time applications. However, since it uses paraperspective projection model, it can be used to applications involving significant camera translation where the orthographic FMs cannot accommodate.

2.3.2 Paraperspective Metric Constraints

As in the original FM, a 3×3 non-singular matrix $\mathbf{A}$ must be determined that transforms $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ into the desired motion matrix $\mathbf{M}$ and shape matrix $\mathbf{S}$. However, under the paraperspective model, the constraints on the motion matrix $\mathbf{M}$ are different from those applying under orthography. Noting that $\mathbf{i}_f$, $\mathbf{j}_f$ and $\mathbf{k}_f$ are mutually orthogonal unit vectors, from eqs. (2.34) - (2.36), the following constraint equations are derived:

$$\frac{\|\mathbf{m}_f\|^2}{1+a_f^2} = \frac{\|\mathbf{n}_f\|^2}{1+b_f^2} \quad \left(= \frac{1}{z_f^2}\right) \tag{2.42}$$

$$\mathbf{m}_f \cdot \mathbf{n}_f = \frac{a_f b_f}{z_f^2} = a_f b_f \frac{1}{2}\left(\frac{\|\mathbf{m}_f\|^2}{1+a_f^2} + \frac{\|\mathbf{n}_f\|^2}{1+b_f^2}\right). \tag{2.43}$$

To avoid the trivial solution $\mathbf{M} = \mathbf{0}$, an additional constraint should be imposed. For instance, let

$$\|\mathbf{m}_1\| = 1. \tag{2.44}$$

Eqs. (2.42), (2.43) and (2.44) give $2F+1$ constraint equations on the row vectors of $\mathbf{M}$. Determination of the transformation matrix $\mathbf{A}$ is similar to that in the original FM [60].

2.3.3 Paraperspective Motion Recovery

Once $\mathbf{A}$ is determined, the desired motion matrix $\mathbf{M}$ and shape matrix $\mathbf{S}$ are recovered as $\mathbf{M} = \hat{\mathbf{M}}\mathbf{A}$, $\mathbf{S} = \mathbf{A}^{-1}\hat{\mathbf{S}}$. Now, for each frame f, the estimates of the camera orientation vectors $\mathbf{i}_f$, $\mathbf{j}_f$, $\mathbf{k}_f$, the average depth z_f, and the camera translation vector $\mathbf{t}_f$ need to be recovered from the known a_f, b_f, $\mathbf{m}_f$, and $\mathbf{n}_f$ (the latter two being rows of $\mathbf{M}$). Noting that $\mathbf{i}_f$, $\mathbf{j}_f$, and $\mathbf{k}_f$ should be orthonormal vectors, the following equation can be derived from eqs. (2.34), (2.35), and (2.36):

$$\mathbf{B}_f \mathbf{k}_f = \mathbf{D}_f, \tag{2.45}$$

where

$$\mathbf{B}_f = \begin{bmatrix} \tilde{\mathbf{m}}_f \times \tilde{\mathbf{n}}_f \\ \tilde{\mathbf{m}}_f \\ \tilde{\mathbf{n}}_f \end{bmatrix}, \quad \mathbf{D}_f = \begin{bmatrix} 1 \\ -a_f \\ -b_f \end{bmatrix}, \tag{2.46}$$

and

$$\tilde{\mathbf{m}}_f = \sqrt{1 + a_f^2}\frac{\mathbf{m}_f}{||\mathbf{m}_f||}, \quad \tilde{\mathbf{n}}_f = \sqrt{1 + b_f^2}\frac{\mathbf{n}_f}{||\mathbf{n}_f||}. \tag{2.47}$$

Then the camera orientation vectors are computed as

$$\mathbf{k}_f = \mathbf{B}_f^{-1}\mathbf{D}_f \tag{2.48}$$

$$\mathbf{i}_f = \tilde{\mathbf{n}}_f \times \mathbf{k}_f \tag{2.49}$$

$$\mathbf{j}_f = \mathbf{k}_f \times \tilde{\mathbf{m}}_f. \tag{2.50}$$

These solutions are unique up to an arbitrary rotation. The arbitrariness can be removed by aligning the world axes with the first frame's camera axes, so that $\mathbf{i}_1 = [1\ 0\ 0]^T$ and $\mathbf{j}_1 = [0\ 1\ 0]^T$.

Finally, the average depth z_f can be computed from eq. (2.42), and the camera translation vector $\mathbf{t}_f$ can be derived from eqs. (2.34) and (2.35).

Chapter 3

MSE Formulation for Shape Space

This chapter first shows that the bilinear formulation can be extended to a general affine projection (GAP) model, and the first step of the FMs under GAP can be viewed as *affine shape space* tracking. A novel MSE cost function is then formulated for the affine shape space tracking. It is shown that this cost function attains a global minimum at the desired affine shape space. The properties of the function are also studied and a comparison with other shape space formulations is made.

3.1 Affine Shape Space under GAP Model

In the last chapter, the bilinear formulation was shown to hold under the orthographic and paraperspective models. We show in this section that it can be generalized to the GAP model. We also show that the first step of the FMs under GAP, computing the *affine shape* and *motion* (the 3D shape and motion matrices up to a 3×3 affine transformation), is equivalent to computing the affine shape space because any basis of the affine shape space provides a solution for the affine shape and motion matrices.

3.1.1 Camera Models

In this subsection we will give a brief review of the perspective camera model and its affine approximation, the GAP model.

The perspective camera

The most frequently used camera model in computer vision is the perspective camera. In Figure 3.1, a 3D-point in the scene with coordinates $\mathbf{X} = [X, Y, Z]^T$ is projected to the image point with coordinates $\mathbf{x} = [x, y]^T$ according to

$$\lambda \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}, \quad \lambda \neq 0, \tag{3.1}$$

where λ is a scale factor and $\mathbf{P}$ is the standard 3×4 camera projection matrix. Using QR-decomposition, $\mathbf{P}$ can be factorized as

$$\mathbf{P} = \mathbf{K}[\mathbf{R}\ \ \mathbf{t}] = \begin{bmatrix} f & sf & x_0 \\ 0 & \gamma f & y_0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R}\ \ \mathbf{t}], \tag{3.2}$$

where $[\mathbf{R}, \mathbf{t}]$ represents a rigid transformation, $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector. The parameters in $\mathbf{K}$ represent intrinsic parameters of the camera where f denotes focal length, γ denotes aspect ratio, s denotes skew, and (x_0, y_0) denotes the coordinates of the principal point.

The affine camera

With the affine camera (first introduced by Mundy and Zisserman [57]), the projection of a 3D point to the image has the same form as in eq. (3.1), but the camera matrix

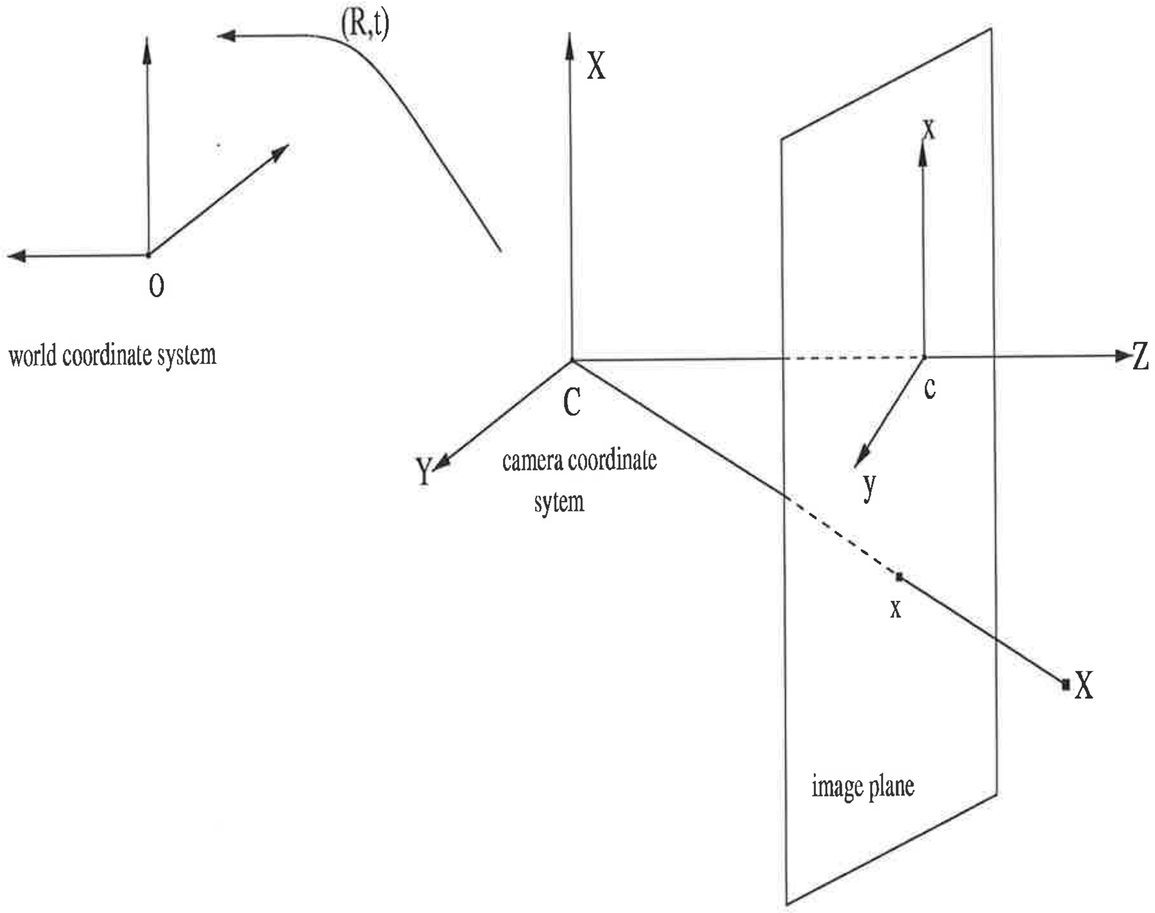

Figure 3.1: Perspective projection model

has the following form

$$\mathbf{P}_A = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & p_{34} \end{bmatrix}. \tag{3.3}$$

and the homogeneous scale factor λ is the same for all points. The affine camera can be seen as a perspective camera with its center located at the plane at infinity. Restricting $\lambda \neq 0$, that is, restricting all affine points within the affine shape space not at infinity, we obtain the following linear equation from eq. (3.1) and (3.3):

$$\mathbf{x} = \mathbf{M}_A\mathbf{X} + \mathbf{t}_A, \tag{3.4}$$

where $\mathbf{M}_A$ is a 2×3 matrix with elements $m_{i,j} = p_{ij}/\lambda$, $(i = 1, 2, \quad j = 1, 2, 3)$, and the 2D vector $\mathbf{t}_A = [p_{14}/\lambda, \; p_{24}/\lambda]^T$. Note that $\lambda = p_{34}$.

The affine camera has the following two important properties [65, 84] that the perspective camera fails to have.

- Under GAP, the parallelism of lines is preserved. That is, lines that are parallel in the space remain parallel in the image.

  Proof. For any two 3D parallel lines $\mathbf{L}_1(\lambda) = \mathbf{X}_a + \lambda\mathbf{d}$ and $\mathbf{L}_2(\mu) = \mathbf{X}_b + \mu\mathbf{d}$, where $\mathbf{X}_a$ and $\mathbf{X}_b$ are two 3D points and $\mathbf{d}$ is the 3D direction vector, their projections under the GAP model are the following two lines:

$$\mathbf{l}_1(\lambda) = (\mathbf{M}_A\mathbf{X}_a + \mathbf{t}_A) + \lambda\mathbf{M}_A\mathbf{d} \tag{3.5}$$
$$\mathbf{l}_2(\mu) = (\mathbf{M}_A\mathbf{X}_b + \mathbf{t}_A) + \mu\mathbf{M}_A\mathbf{d}.$$

  It is easy to see that they are parallel to the 2D direction vector $\mathbf{M}_A\mathbf{d}$.

- Under GAP, the centroid of a set of 3D points and the centroid of their image points correspond to each other.

  Proof. Let $\mathbf{X}_j$ $(j = 1, \cdots, n)$ be a set of 3D points and $\mathbf{x}_j$ their corresponding image points. The centroid of the image points is then

$$\mathbf{o} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{x}_j = \frac{1}{n}\sum_{j=1}^{n}(\mathbf{M}_A\mathbf{X}_j + \mathbf{t}_A) = \mathbf{M}_A\mathbf{G} + \mathbf{t}_A, \tag{3.6}$$

  where $\mathbf{G} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{X}_j$ is the centroid of the set of 3D points.

  Thus, if the 3D points are expressed with respect to their centroid, i.e. $\tilde{\mathbf{X}}_j = \mathbf{X}_j - \mathbf{G}$, and if the image points are also expressed with respect to their centroid, i.e. $\tilde{\mathbf{x}}_j = \mathbf{x}_j - \mathbf{o}$, then we have

$$\tilde{\mathbf{x}}_j = \mathbf{M}_A\tilde{\mathbf{X}}_j. \tag{3.7}$$

The affine camera is a linear approximation to the perspective camera and it generalizes the orthographic, the weak perspective and the paraperspective camera models. These models provide a good approximation to the perspective camera when the size of an object is small compared to the viewing distance, which is satisfied in most practical situations.

3.1.2 The Bilinear Formulation under GAP

Assume that P feature points of a rigid object are tracked through F image frames. Let $\mathbf{w}_{ij} = [u_{ij}, v_{ij}]^T$ denote the coordinates of the j-th image point in the i-th frame. As in the original FM, suppose that the origin of the world coordinate system resides at the centroid $\mathbf{G}$ of P object points. Object shape is then represented by the $3 \times P$ matrix

$$\mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_P], \tag{3.8}$$

where $\mathbf{s}_j$ describes the 3D location of the j-th object point in the world coordinate system. We can infer that the shape matrix $\mathbf{S}$ should be fixed throughout the image sequence in the object-centroid world coordinate system, and that

$$\mathbf{G} = \frac{1}{P}\sum_{j=1}^{P} \mathbf{s}_j = \mathbf{0}. \tag{3.9}$$

Let

$$\mathbf{o}_i = \frac{1}{P}\sum_{j=1}^{P} \mathbf{w}_{ij} \tag{3.10}$$

represent the image centroid of the i-th frame. Subtracting $\mathbf{o}_i$ from $\mathbf{w}_{ij}$, we get the registered image coordinates $\tilde{\mathbf{w}}_{ij} = [x_{ij},\ y_{ij}]^T$ of feature point j in image i as

$$\tilde{\mathbf{w}}_{ij} = \mathbf{w}_{ij} - \mathbf{o}_i. \tag{3.11}$$

According to property 2 of the GAP model, we have

$$\tilde{\mathbf{w}}_{ij} = \mathbf{M}_i \mathbf{s}_j, \tag{3.12}$$

where $\mathbf{M}_i$ is the 2×3 affine matrix of the i-th frame.

Combining eq. (3.12) for all feature points from 1 to P, and all frames from 1 to F, into a single matrix equation, we obtain

$$\begin{bmatrix} \tilde{\mathbf{w}}_{11} & \cdots & \tilde{\mathbf{w}}_{1P} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{w}}_{F1} & \cdots & \tilde{\mathbf{w}}_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_F \end{bmatrix} [\mathbf{s}_1 \cdots \mathbf{s}_P], \tag{3.13}$$

or in short form

$$\tilde{\mathbf{W}} = \mathbf{MS}, \tag{3.14}$$

where $\tilde{\mathbf{W}}$ is the $2F \times P$ registered measurement matrix, $\mathbf{M}$ is the $2F \times 3$ matrix, and $\mathbf{S}$ is the $3 \times P$ shape matrix. Since $\mathbf{M}$ is determined by the camera calibration parameters and the camera motion parameters at all frames, we call it the camera motion matrix. From Eq. (3.14), we can see that the bilinear form under GAP has the same form as that under orthography. Therefore the rank theorem (in section 2.1.1) still holds under GAP model. The implication of the theorem is that the affine shape $\hat{\mathbf{S}} \in \Re^{\mathbf{3\times P}}$ and motion $\hat{\mathbf{M}} \in \Re^{\mathbf{2F\times 3}}$ (the 3D shape $\mathbf{S}$ and motion $\mathbf{M}$ up to an affine transformation $\mathbf{A} \in \Re^{\mathbf{3\times 3}}$) can be readily obtained by a rank-3 decomposition of $\tilde{\mathbf{W}}$, such that

$$\hat{\mathbf{W}} = \hat{\mathbf{M}}\hat{\mathbf{S}} = \mathbf{MS} \tag{3.15}$$

$$\mathbf{M} = \hat{\mathbf{M}}\mathbf{A} \quad \mathbf{S} = \mathbf{A}^{-1}\hat{\mathbf{S}}, \tag{3.16}$$

where $\hat{\mathbf{W}}$ is the best rank-3 estimate of $\tilde{\mathbf{W}}$.

As under orthography, *affine shape space* (simply referred to as shape space) is defined as $range(\mathbf{S}^T)$, and the *affine motion space* (simply referred to as motion space) is defined as $range(\mathbf{M})$. They are equivalent to the row space and column space of $\hat{\mathbf{W}}$ respectively. Hence their ranks are at most 3. From eq. (3.15), we know that any solution of $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ provides a basis for motion space and shape space respectively.

Inversely, we can infer that any basis of motion space or shape space provides a solution to $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$. Since motion space and shape space are dual in the sense that one can be computed from the other, and motion space is only meaningful in the batch FMs when all the frames are processed at one time, we track shape space to compute affine shape and motion recursively at each frame.

3.2 An MSE Formulation for Shape Space

Prior to presenting a new shape space tracking scheme in the next chapter, we first formulate an MSE cost function. It is shown that this cost function attains a global minimum at the desired shape space. Several important properties of the function are also discussed.

3.2.1 Preliminaries

Let

$$\begin{aligned}\mathbf{x}_k &= [x_{k1}, x_{k2}, \cdots, x_{kP}]^T \\ \mathbf{y}_k &= [y_{k1}, y_{k2}, \cdots, y_{kP}]^T\end{aligned} \tag{3.17}$$

represent the registered image vectors of the k-th frame, where $[x_{kj}, y_{kj}]^T$ $(j = 1, \cdots, P)$ are the P image point coordinates with respect to their centroid $\mathbf{o}_k$. The registered measurement matrix $\tilde{\mathbf{W}}$ can then be expressed by

$$\tilde{\mathbf{W}} = [\mathbf{x}_1\ \mathbf{y}_1 \cdots \mathbf{x}_F\ \mathbf{y}_F]^T. \tag{3.18}$$

Define the $P \times P$ correlation matrix $\mathbf{C}$ to be

$$\mathbf{C} = E\{\mathbf{x}_i\mathbf{x}_i^T + \mathbf{y}_i\mathbf{y}_i^T\} \tag{3.19}$$

$$= \frac{1}{F}\sum_{i=1}^{F}(\mathbf{x}_i\mathbf{x}_i^T + \mathbf{y}_i\mathbf{y}_i^T)$$
$$= \frac{1}{F}\tilde{\mathbf{W}}^T\tilde{\mathbf{W}},$$

where $E\{\}$ denotes expectation. From eq. (3.19), we can infer that the rank of $\mathbf{C}$ is at most three since that of $\tilde{\mathbf{W}}$ is at most three.

Theorem 3.1 *The three dominant eigenvectors of* $\mathbf{C}$ *span the shape space.*

Proof. Assuming the SVD factorization of $\tilde{\mathbf{W}}$ is

$$\tilde{\mathbf{W}} = \mathbf{O}_1\mathbf{\Sigma}\mathbf{O}_2^T, \tag{3.20}$$

where $\mathbf{O}_1 \in \Re^{2F\times P}$, $\mathbf{O}_2 \in \Re^{P\times P}$ satisfy $\mathbf{O}_1^T\mathbf{O}_1 = \mathbf{O}_2^T\mathbf{O}_2 = \mathbf{O}_2\mathbf{O}_2^T = \mathbf{I}_P$, and $\mathbf{\Sigma} = diag(\sigma_1, \cdots, \sigma_P)$ contains the P singular values. We then have

$$\mathbf{C} = \frac{1}{F}\mathbf{O}_2\mathbf{\Sigma}\mathbf{O}_1^T\mathbf{O}_1\mathbf{\Sigma}\mathbf{O}_2^T = \mathbf{O}_2(\frac{1}{F}\mathbf{\Sigma}^2)\mathbf{O}_2^T. \tag{3.21}$$

This is the eigenvalue decomposition of $\mathbf{C}$. It is shown that the eigenvectors of $\mathbf{C}$ coincide with the right singular vectors of $\tilde{\mathbf{W}}$.

Since the rank of $\tilde{\mathbf{W}}$ is at most three, by considering only its largest three singular values and setting $\sigma_4 = \cdots = \sigma_P = 0$, the best rank-3 estimate of $\tilde{\mathbf{W}}$ is then

$$\hat{\mathbf{W}} = \mathbf{U}\mathbf{\Sigma}'\mathbf{V}^T, \tag{3.22}$$

where $\mathbf{U}$ and $\mathbf{V}$ are formed by the first three columns of $\mathbf{O}_1$ and $\mathbf{O}_2$ respectively, and $\mathbf{\Sigma}' = diag(\sigma_1, \sigma_2, \sigma_3)$.

Similarly, the best rank-3 estimate of $\mathbf{C}$ is obtained by only considering its largest three eigenvalues and setting all the others to be zero in the eq. (3.21), that is

$$\hat{\mathbf{C}} = \mathbf{V}(\frac{1}{F}\mathbf{\Sigma}'^2)\mathbf{V}^T. \tag{3.23}$$

Eq. (3.23) reveals that the three dominant eigenvectors of $\mathbf{C}$ coincide with the first three right singular vectors of $\tilde{\mathbf{W}}$. We know from Section 2.1.2 that the latter span the shape space. Thus Theorem 3.1 holds.

We then infer from Theorem 3.1 that the shape space can be obtained by computing the rank-3 principal subspace of $\mathbf{C}$.

3.2.2 An MSE Cost Function

If the columns of a matrix $\mathbf{Q} \in \Re^{P\times 3}$ are an orthonormal basis of the shape space, then $\mathbf{Q}\mathbf{Q}^T$ represents the orthogonal projection matrix of the shape space, whereas $\mathbf{I}-\mathbf{Q}\mathbf{Q}^T$ represents its orthogonal complement. Any $P \times 1$ vectors $\mathbf{x}_k$, $\mathbf{y}_k$ can each be decomposed into two components, viz:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{Q}\mathbf{Q}^T\mathbf{x}_k + (\mathbf{I}-\mathbf{Q}\mathbf{Q}^T)\mathbf{x}_k \\ \mathbf{y}_k &= \mathbf{Q}\mathbf{Q}^T\mathbf{y}_k + (\mathbf{I}-\mathbf{Q}\mathbf{Q}^T)\mathbf{y}_k, \end{aligned} \tag{3.24}$$

Here, the first component is the orthogonal projection onto the shape space, while the second component is the projection onto the orthogonal complement. If $\mathbf{x}_k$, $\mathbf{y}_k$ are ideal, noise-free measurements, they will be wholly contained within the shape space, and their second components will be zero. If in contrast $\mathbf{x}_k$, $\mathbf{y}_k$ are corrupted with noise, as is inevitable in practice, they will in general not be wholly contained within the shape space, having second components unequal to zero. Hence the second components are the noise effects in the measurement vectors $\mathbf{x}_k$, $\mathbf{y}_k$. A natural optimization procedure is thus to seek a matrix $\mathbf{Q}$ which minimizes in some sense the magnitude of the associated second components. This suggests that the following MSE cost function be minimized:

$$\begin{aligned} J_{MSE}(\mathbf{Q}) &= E\{\| (\mathbf{I}-\mathbf{Q}\mathbf{Q}^T)\mathbf{x}_k \|^2 + \| (\mathbf{I}-\mathbf{Q}\mathbf{Q}^T)\mathbf{y}_k \|^2\} \qquad (3.25) \\ &= E\{\| \mathbf{x}_k - \mathbf{Q}\mathbf{Q}^T\mathbf{x}_k \|^2 + \| \mathbf{y}_k - \mathbf{Q}\mathbf{Q}^T\mathbf{y}_k \|^2\} \end{aligned}$$

$$= tr\{\mathbf{C}\} - 2tr\{\mathbf{Q}^T\mathbf{C}\mathbf{Q}\} + tr\{\mathbf{Q}^T\mathbf{C}\mathbf{Q}\mathbf{Q}^T\mathbf{Q}\},$$

where tr denotes trace, and $\mathbf{Q}$ is assumed to have full rank 3. Note that we do not impose any constraints on $\mathbf{Q}$. In particular, there is no constraint of $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_3$ as in some other optimization formulations for subspace tracking. Minimization of $J_{MSE}(\mathbf{Q})$ has some attractive properties. First, note the following two theorems, which are presented and proved in [85]:

Theorem 3.2 *$\mathbf{Q}$ is a stationary point of $J_{MSE}(\mathbf{Q})$ if and only if $\mathbf{Q} = \mathbf{G}_3\mathbf{H}$, where $\mathbf{G}_3 \in \Re^{\mathbf{P}\times\mathbf{3}}$ contains any 3 distinct eigenvectors of $\mathbf{C}$ and $\mathbf{H} \in \Re^{\mathbf{3}\times\mathbf{3}}$ is an arbitrary orthogonal matrix. At each stationary point, $J_{MSE}(\mathbf{Q})$ equals the sum of eigenvalues whose eigenvectors are not involved in $\mathbf{G}_3$.*

Theorem 3.3 *When $\mathbf{G}_3$ contains the three dominant eigenvectors of $\mathbf{C}$, $J_{MSE}(\mathbf{Q})$ attains the global minimum, while all the other stationary points are saddle points.*

From the above two theorems, we can infer that:

- the stationary points of $J_{MSE}(\mathbf{Q})$ satisfy $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_3$, i.e. the columns of $\mathbf{Q}$ being orthonormal.
- when the columns of $\mathbf{Q}$ are one of the orthonormal bases of the rank-3 principal subspace of $\mathbf{C}$, $J_{MSE}(\mathbf{Q})$ attains the global minimum, while all the other stationary points of $J_{MSE}(\mathbf{Q})$ are saddle points.

Hence, by minimizing $J_{MSE}(\mathbf{Q})$, we can obtain an orthonormal basis (the three columns of $\mathbf{Q}$) of the shape space.

3.3 Properties and Comparisons

In this section, we discuss several key properties of the MSE formulation.

Property 1. The cost function $J_{MSE}(\mathbf{Q})$ has a global minimum at the shape space and no other local minima. Thus, iterative methods which minimize $J_{MSE}(\mathbf{Q})$ with proper initializations are guaranteed to globally converge to the rank-3 principal subspace of $\mathbf{C}$, i.e., the desired shape space. The initializations of the iteration methods are discussed in chapter 4.

Property 2. The two theorem of $J_{MSE}(\mathbf{Q})$ show that minimization of $J_{MSE}(\mathbf{Q})$ automatically orthonormalize the columns of $\mathbf{Q}$. No explicit constraint of orthonormality $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_3$ is imposed beforehand. Thus minimizing $J_{MSE}(\mathbf{Q})$ is an unconstrained optimization problem, which has advantages over those formulations where $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_3$ is imposed as an optimization constraint. The advantage is that in the iteration algorithms of minimizing $J_{MSE}(\mathbf{Q})$, no step of reorthonormalizating $\mathbf{Q}$ is needed, while in the constrained optimization scheme, such a step is required at each iteration which results in extra computational cost. Therefore, the use of an iterative algorithm to minimize $J_{MSE}(\mathbf{Q})$ will always converge to an orthonormal basis of the shape space without any orthonormalization operations during the iteration.

Property 3. At the global minimum of $J_{MSE}(\mathbf{Q})$, generally $\mathbf{Q}$ only yields an arbitrary orthonormal basis for the rank-3 principal subspace of $\mathbf{C}$, i.e. the shape space, but not the true dominant eigenvectors. This is because $J_{MSE}(\mathbf{Q})$ is invariant with respect to rotation of $\mathbf{Q}$, i.e. $J_{MSE}(\mathbf{Q}) = J_{MSE}(\mathbf{QR})$ holds with any 3×3 rotation matrix $\mathbf{R}$. Therefore, the solution of $\mathbf{Q}$ obtained by minimizing $J_{MSE}(\mathbf{Q})$ is unique up to an arbitrary rotation. However, $\mathbf{QQ}^T$, the orthogonal projection matrix of the shape space, is always unique. For the special case when $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$, i.e. the three largest eigenvalues of $\mathbf{C}$ are equal, $\mathbf{Q}$ yields the true dominant eigenvectors because $\mathbf{Q}^T\mathbf{C}\mathbf{Q} = \lambda\mathbf{I}_3$ produces the required partial eigenvalue decomposition (EVD) of $\mathbf{C}$.

Chapter 4

Recursive Computation of Shape Space

In this chapter, we first review the standard recursive least-squares (RLS) algorithm. Then we develop an efficient, novel recursive algorithm for MSE minimization. This is to compute an estimate of the shape space at frame k from that of frame $k-1$ and the new incoming data, the image coordinate vectors at frame k, by using the RLS technique. Finally, the performance of the RLS shape space tracking algorithm is compared with that of the original and the sequential FMs by synthetic tests.

4.1 Recursive Least-Squares Technique

In this section, we review a well-known adaptive subspace tracking technique — the *recursive least squares* algorithm [28]. The RLS algorithm may be viewed as a special case of the Kalman filter. It is also based on a relation in matrix algebra known as the *matrix inversion lemma*, which is also reviewed in this section.

4.1.1 Matrix Inversion Lemma

Let $\mathbf{A}$ and $\mathbf{B}$ be two positive-definite, $m \times m$ matrices related by

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^{H}, \tag{4.1}$$

where $\mathbf{D}$ is another positive-definite $n \times n$ matrix, and $\mathbf{C}$ is an $m \times n$ matrix. According to the *matrix inversion lemma*, we may express the inverse of the matrix $\mathbf{A}$ as following:

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^{H}\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^{H}\mathbf{B} \tag{4.2}$$

The proof of this lemma is established by multiplying the right hand sides of eq. (4.1) and eq. (4.2), and noting that their product is equal to the identity matrix $\mathbf{I}_m$. The matrix inversion lemma states that if we are given a matrix $\mathbf{A}$ as defined in eq. (4.1), we can determine its inverse $\mathbf{A}^{-1}$ by using the relation of eq. (4.2).

4.1.2 Standard RLS Algorithm

Some Preliminaries

In recursive implementations of the method of least squares, we start the computation with known initial conditions and use the information contained in new data samples to update the old estimates. Since the length of observable data is generally variable, we denote the *cost function* to be minimized as $\Psi(n)$, where n is the variable length of the observable data, and $\Psi(n)$ is given by

$$\Psi(n) = \sum_{i=1}^{n} \lambda^{n-i} |e(i)|^2 \tag{4.3}$$

where $0 < \lambda < 1$ is "forgetting factor", with $\lambda = 1$ corresponding to infinite memory. The expression $e(i)$ is the difference between the *desired response* $d(i)$ and the *output*

$y(i)$ produced by a transversal filter whose tap inputs equal $u(i), u(i-1), \cdots, u(i-M+1)$, as in Figure 4.1. That is,

$$\begin{aligned} e(i) &= d(i) - y(i) \\ &= d(i) - \mathbf{w}^H(n)\mathbf{u}(i) \end{aligned} \tag{4.4}$$

where $\mathbf{u}(i)$ is the tap-input vector at time instant i, defined by

$$\mathbf{u}(i) = [u(i),\ u(i-1), \cdots, u(i-M+1)]^T, \tag{4.5}$$

and $\mathbf{w}(n)$ is the tap-weight vector at time n, defined by

$$\mathbf{w}(n) = [w_0(n),\ w_1(n), \cdots, w_{M-1}(n)]^T. \tag{4.6}$$

Note that the tap weights of the transversal filter remain fixed during the observation interval $1 \leq i \leq n$.

The optimum value of the tap-weight vector, $\hat{\mathbf{w}}(n)$, for which the cost function $\Psi(n)$ attains its minimum value is determined by the matrix equation

$$\mathbf{\Phi}(n)\hat{\mathbf{w}}(n) = \theta(n) \tag{4.7}$$

The $M \times M$ correlation matrix $\mathbf{\Phi}(n)$ is defined by

$$\mathbf{\Phi}(n) = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}(i)\mathbf{u}^H(i). \tag{4.8}$$

The $M \times 1$ cross-correlation $\theta(n)$ between the tap inputs of the transversal filter and the desired response is defined by

$$\theta(n) = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}(i)d^*(i), \tag{4.9}$$

where * denotes complex conjugation. Eqs. (4.8) and (4.9) may be rewritten as

$$\mathbf{\Phi}(n) = \lambda\mathbf{\Phi}(n-1) + \mathbf{u}(n)\mathbf{u}^H(n) \tag{4.10}$$

$$\theta(n) = \lambda\theta(n-1) + \mathbf{u}(n)d^*(n). \tag{4.11}$$

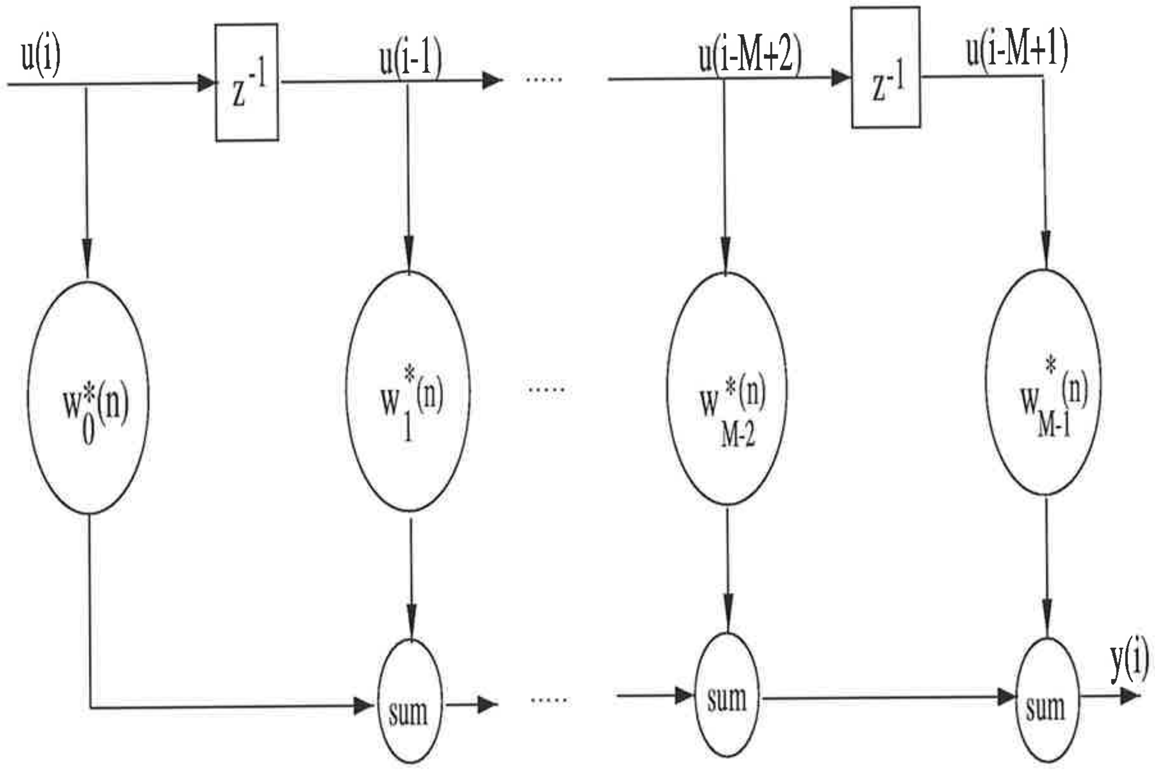

Figure 4.1: The transversal filter

To compute the least-square estimate $\hat{\mathbf{w}}(n)$ in eq. (4.7), the inverse of the $M \times M$ correlation matrix $\mathbf{\Phi}(n)$ has to be determined. Since computation of the inverse is very time consuming, particularly if the number of tap weights, M, is high, we use the *matrix inversion lemma* to reduce the computational cost.

The Exponentially Weighted RLS Algorithm

With the correlation matrix $\mathbf{\Phi}(n)$ assumed to be positive definite, the matrix inversion lemma can be applied to eq. (4.10) as

$$\mathbf{\Phi}^{-1}(n) = \lambda^{-1}\mathbf{\Phi}^{-1}(n-1) - \frac{\lambda^{-2}\mathbf{\Phi}^{-1}(n-1)\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{\Phi}^{-1}(n-1)}{1+\lambda^{-1}\mathbf{u}^H(n)\mathbf{\Phi}^{-1}(n-1)\mathbf{u}(n)} \tag{4.12}$$

For easier computation, let

$$\mathbf{P}(n) = \mathbf{\Phi}^{-1}(n) \tag{4.13}$$

and

$$\mathbf{k}(n) = \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{1+\lambda^{-1}\mathbf{u}^H(n)\mathbf{P}(n-1)\mathbf{u}(n)}. \tag{4.14}$$

Then eq.(4.12) may be rewritten as

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1). \tag{4.15}$$

Here $\mathbf{k}(n)$ is an $M \times 1$ vector and is referred to the *gain vector*. From eq. (4.14) and eq. (4.15), we obtain the following relation

$$\mathbf{k}(n) = \mathbf{P}(n)\mathbf{u}(n) = \mathbf{\Phi}^{-1}(n)\mathbf{u}(n). \tag{4.16}$$

From all the above, the following RLS algorithm can be obtained:

$$\begin{aligned}
\mathbf{k}(n) &= \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{1+\lambda^{-1}\mathbf{u}^H(n)\mathbf{P}(n-1)\mathbf{u}(n)} \\
a(n) &= d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{u}(n) \\
\hat{\mathbf{w}}(n) &= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)a^*(n) \\
\mathbf{P}(n) &= \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1)
\end{aligned}$$

Figure 4.2 shows the diagram for the RLS algorithm.

An important feature of the RLS algorithm described by these equations is that the inversion of the correlation matrix $\mathbf{\Phi}(n)$ is replaced at each step by a simple scalar division. This greatly reduces the computational complexity of the RLS algorithm.

The initialization of the RLS algorithm requires a starting value $\mathbf{P}(0)$ to assure the nonsingularity of the correlation matrix $\mathbf{\Phi}(n)$. A simple choice is

$$\mathbf{P}(0) = \delta\mathbf{I}_M, \quad \hat{\mathbf{w}}(0) = \mathbf{0}_{M\times 1}, \tag{4.17}$$

where $0 < \delta < 1$.

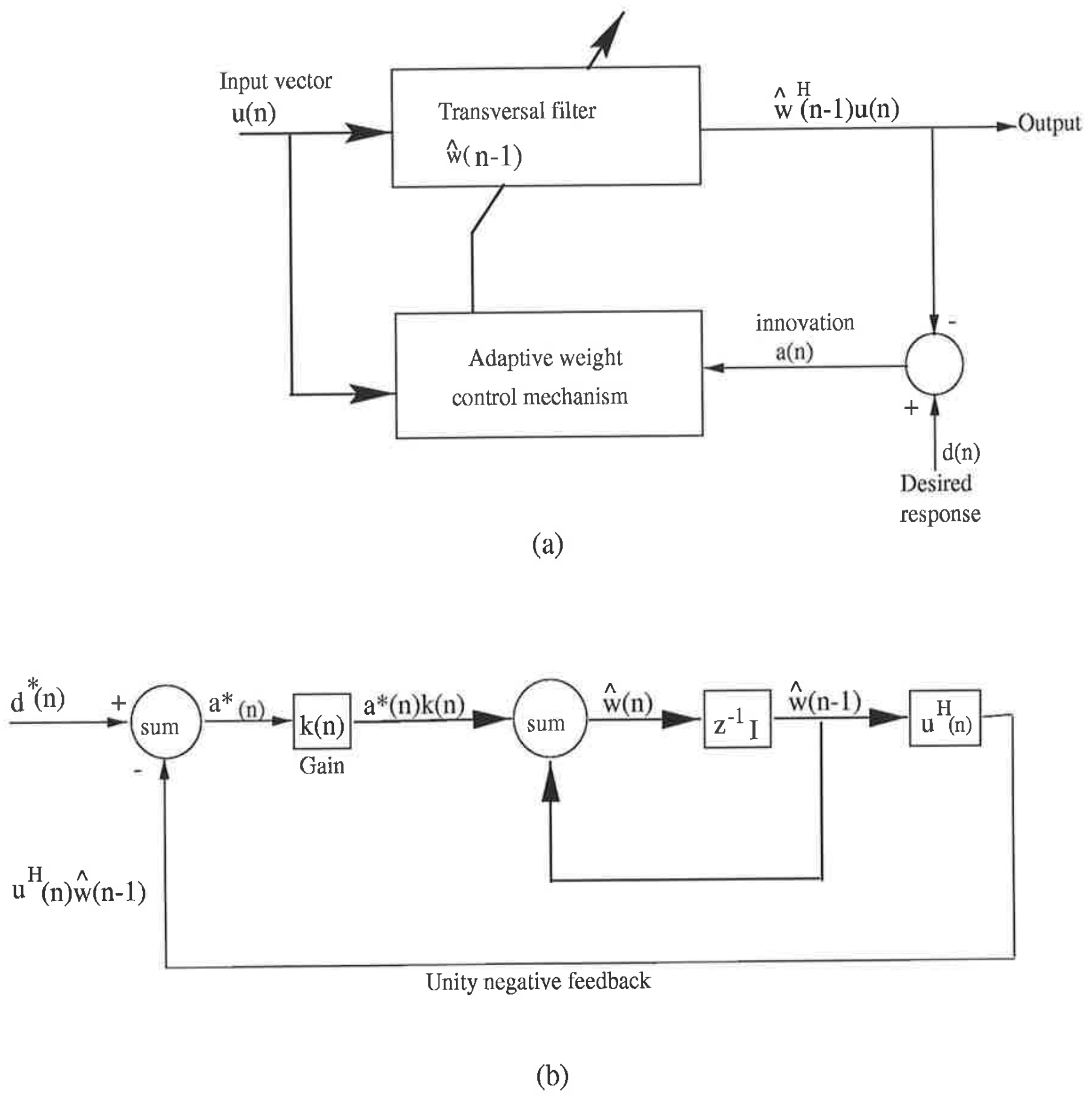

Figure 4.2: Representations of the RLS algorithm: (a) block diagram; (b) signal-flow graph.

4.2 RLS Shape Space Updating

In this section, we present a new recursive scheme for estimating the shape space at each frame with the low complexity $O(P)$ by using the *projection approximation subspace* *tracking* (PAST) approach and the RLS technique. After that, we show that the affine shape and motion can be easily computed from the estimated shape space.

4.2.1 Recursive Shape Space Updating

We develop an RLS algorithm for updating an arbitrary orthonormal basis of the shape space at frame k given the basis at frame $k-1$ and the incoming image coordinate data $\mathbf{u}_k, \mathbf{v}_k$:

$$\begin{aligned}\mathbf{u}_k &= [u_{k1}, u_{k2}, \cdots, u_{kP}]^T \\ \mathbf{v}_k &= [v_{k1}, v_{k2}, \cdots, v_{kP}]^T.\end{aligned} \tag{4.18}$$

The registered image vectors $\mathbf{x}_k, \mathbf{y}_k$ are then given by

$$\begin{aligned}\mathbf{x}_k &= \mathbf{u}_k - a_k \\ \mathbf{y}_k &= \mathbf{v}_k - b_k,\end{aligned} \tag{4.19}$$

where $(a_k,\ b_k)$ is the image centroid of P features at frame k, and

$$a_k = \frac{1}{P}\sum_{j=1}^{P} u_{kj}, \quad b_k = \frac{1}{P}\sum_{j=1}^{P} v_{kj}. \tag{4.20}$$

Let

$$\tilde{\mathbf{w}}_k = [\mathbf{x}_k \;\; \mathbf{y}_k] \tag{4.21}$$

$$\overline{\mathbf{w}}_k = \mathbf{Q}_{k-1}^T \tilde{\mathbf{w}}_k, \tag{4.22}$$

where $\mathbf{Q}_{k-1} \in \Re^{\mathbf{P}\times\mathbf{3}}$ is assumed having full rank 3. Then $\mathbf{C}_k$, the $P \times P$ correlation matrix up to frame k, can be expressed as

$$\mathbf{C}_k = E\{\tilde{\mathbf{w}}_i \tilde{\mathbf{w}}_i^T\} = \frac{1}{k}\sum_{i=1}^{k} \tilde{\mathbf{w}}_i \tilde{\mathbf{w}}_i^T, \tag{4.23}$$

and the MSE cost function for estimating $\mathbf{Q}_k$ (the shape space at frame k) is obtained by using $\mathbf{C}_k$ to approximate $\mathbf{C}$ in eq. (3.25), as given by

$$\begin{aligned} J_{MSE}(\mathbf{Q}_k) &= E\{\| \tilde{\mathbf{w}}_i - \mathbf{Q}_k\mathbf{Q}_k^T\tilde{\mathbf{w}}_i \|_F^2\} \\ &= \frac{1}{k}\sum_{i=1}^{k} \| \tilde{\mathbf{w}}_i - \mathbf{Q}_k\mathbf{Q}_k^T\tilde{\mathbf{w}}_i \|_F^2 \\ &= tr\{\mathbf{C}_k\} - 2tr\{\mathbf{Q}_k^T\mathbf{C}_k\mathbf{Q}_k\} + tr\{\mathbf{Q}_k^T\mathbf{C}_k\mathbf{Q}_k\mathbf{Q}_k^T\mathbf{Q}_k\}. \end{aligned} \tag{4.24}$$

$J_{MSE}(\mathbf{Q}_k)$ in eq. (4.24) is identical to $J_{MSE}(\mathbf{Q})$ in eq. (3.25) except for the use of the image data up to frame k and not in all F frames. Thus the theorems and properties about $J_{MSE}(\mathbf{Q})$ in Chapter 3 also hold for $J_{MSE}(\mathbf{Q}_k)$. Hence by minimizing $J_{MSE}(\mathbf{Q}_k)$, the columns of $\mathbf{Q}_k$ provide an orthonormal basis for the rank-3 principal subspace of $\mathbf{C}_k$, that is, the shape space at frame k.

$J_{MSE}(\mathbf{Q}_k)$ is a fourth-order function of the elements of $\mathbf{Q}_k$ and the standard RLS technique cannot be applied directly to compute its minimization recursively. Accordingly, we now seek a simpler quadratic cost function by making some approximation. Assume that $\mathbf{Q}_k^T\tilde{\mathbf{w}}_i$ (the projections of $\mathbf{x}_k$, $\mathbf{y}_k$ onto the columns of $\mathbf{Q}_k$) can be approximated by $\overline{\mathbf{w}}_i = \mathbf{Q}_{i-1}^T\tilde{\mathbf{w}}_i$, for all $1 \leq i \leq k$. Since we assume the shape space is stationary, the errors in the above projection approximation should be small. Note that in doing so, $\overline{\mathbf{w}}_i$ becomes immediately available at any frame i given the previous frame estimate $\mathbf{Q}_{i-1}$, for $i = 1, 2, \cdots, k$. Without this approximation, $\mathbf{Q}_k^T\tilde{\mathbf{w}}_i$ would only be available at frame k when $\mathbf{Q}_k$ is obtained. This is the main idea of the PAST approach that was presented by Yang in [85]. After the projection approximation $\mathbf{Q}_k^T\tilde{\mathbf{w}}_i \approx \overline{\mathbf{w}}_i$, $J_{MSE}(\mathbf{Q}_k)$ is simplified to $J'(\mathbf{Q}_k)$ given by

$$J'(\mathbf{Q}_k) = \frac{1}{k}\sum_{i=1}^{k} \| \tilde{\mathbf{w}}_i - \mathbf{Q}_k\overline{\mathbf{w}}_i \|_F^2 \, . \tag{4.25}$$

Under the stationary shape space assumption, $J'(\mathbf{Q}_k)$ should be a good approximation to $J_{MSE}(\mathbf{Q}_k)$ and the matrix $\mathbf{Q}_k$ obtained by minimizing $J'(\mathbf{Q}_k)$ should be a good estimate for the shape space.

Let

$$\mathbf{M}_k = \frac{1}{k}\sum_{i=1}^{k} \tilde{\mathbf{w}}_i \overline{\mathbf{w}}_i^T = \mathbf{M}_{k-1} + \tilde{\mathbf{w}}_k \overline{\mathbf{w}}_k^T \tag{4.26}$$

$$\mathbf{N}_k = \frac{1}{k}\sum_{i=1}^{k} \overline{\mathbf{w}}_i \overline{\mathbf{w}}_i^T = \mathbf{N}_{k-1} + \overline{\mathbf{w}}_k \overline{\mathbf{w}}_k^T, \tag{4.27}$$

then we have that

$$\begin{aligned} J'(\mathbf{Q}_k) &= tr\{\frac{1}{k}\sum_{i=1}^{k}(\tilde{\mathbf{w}}_i - \mathbf{Q}_k\overline{\mathbf{w}}_i)(\tilde{\mathbf{w}}_i - \mathbf{Q}_k\overline{\mathbf{w}}_i)^T\} \\ &= tr\{\mathbf{C}_k\} - 2tr\{\mathbf{M}_k\mathbf{Q}_k^T\} + tr\{\mathbf{Q}_k\mathbf{N}_k\mathbf{Q}_k^T\}. \end{aligned} \tag{4.28}$$

To compute the gradient of $J'(\mathbf{Q}_k)$, we first introduce the following theorem [47] which transfers the computation of gradient to the relatively easy differentials computation:

Theorem 4.1 *Let $\Phi(\mathbf{X})$ be a differentiable real-valued function of an $m \times n$ matrix $\mathbf{X}$. Then the following relationship holds*

$$d\Phi(\mathbf{X}) = tr\{[\Psi(\mathbf{X})]^T d\mathbf{X}\} \Longleftrightarrow \nabla\Phi(\mathbf{X}) = \Psi(\mathbf{X}), \tag{4.29}$$

where d denotes the differential, and the gradient of $\Phi(\mathbf{X})$ with respect to $\mathbf{X}$ is defined as $\nabla\Phi(\mathbf{X}) = \frac{\partial\Phi(\mathbf{X})}{\partial\mathbf{X}}$.

After some computation, the differential of $J'(\mathbf{Q}_k)$ is

$$dJ'(\mathbf{Q}_k) = -2tr\{\mathbf{M}_k^T d\mathbf{Q}_k\} + 2tr\{\mathbf{N}_k\mathbf{Q}_k^T d\mathbf{Q}_k\} \tag{4.30}$$

$$= tr\{(-2\mathbf{M}_k + 2\mathbf{Q}_k\mathbf{N}_k)^T d\mathbf{Q}_k\}. \tag{4.31}$$

According to theorem 4.1, we have

$$\nabla J'(\mathbf{Q}_k) = -2\mathbf{M}_k + 2\mathbf{Q}_k\mathbf{N}_k. \tag{4.32}$$

$J'(\mathbf{Q}_k)$ is minimized when $\nabla J'(\mathbf{Q}_k) = 0$, that is when

$$\mathbf{Q}_k = \mathbf{M}_k\mathbf{N}_k^{-1} \tag{4.33}$$

A recursive computation of $\mathbf{Q}_k$ by the above equation ($9P+27$ flops), with the update of $\overline{\mathbf{w}}_k$ in eq. (4.22) ($6P$ flops), $\mathbf{M}_k$ in eq. (4.26) ($6P$ flops), and $\mathbf{N}_k$ in eq. (4.27) (18 flops) requires a total of $39P$ flops operation. In this thesis, one flop is defined as one multiplication plus an optional addition. A more efficient and numerically more robust method is to apply the *matrix inversion lemma* (see Section 4.1.1) to compute the inverse of $\mathbf{N}_k$ in eq. (4.33). This results in the RLS algorithm, which is derived in the following, to update $\mathbf{Q}_k$ recursively.

Define the 3×3 inverse matrix

$$\mathbf{P}_k = \mathbf{N}_k^{-1}. \tag{4.34}$$

Applying the matrix inversion lemma to $\mathbf{P}_k$, we obtain

$$\begin{aligned}\mathbf{P}_k &= (\mathbf{N}_{k-1} + \overline{\mathbf{w}}_k \overline{\mathbf{w}}_k^T)^{-1} \\ &= \mathbf{P}_{k-1} - \mathbf{g}_k \overline{\mathbf{w}}_k^T \mathbf{P}_{k-1}\end{aligned} \tag{4.35}$$

where

$$\mathbf{g}_k = \mathbf{P}_{k-1}\overline{\mathbf{w}}_k(\mathbf{I}_2 + \overline{\mathbf{w}}_k^T \mathbf{P}_{k-1}\overline{\mathbf{w}}_k)^{-1} \tag{4.36}$$

is the 3×2 gain matrix. Using eq. (4.35) it is easy to verify that

$$\mathbf{g}_k = \mathbf{P}_k \overline{\mathbf{w}}_k. \tag{4.37}$$

Noting that

$$\mathbf{g}_k \overline{\mathbf{w}}_k^T \mathbf{P}_{k-1} = (\mathbf{g}_k \overline{\mathbf{w}}_k^T \mathbf{P}_{k-1})^T = \mathbf{P}_{k-1}\overline{\mathbf{w}}_k \mathbf{g}_k^T, \tag{4.38}$$

we may use eqs. (4.26), (4.27), (4.33) - (4.37) to obtain

$$\begin{aligned}\mathbf{Q}_k &= \mathbf{M}_k \mathbf{P}_k \\ &= \mathbf{M}_{k-1}(\mathbf{P}_{k-1} - \mathbf{g}_k \overline{\mathbf{w}}_k^T \mathbf{P}_{k-1}) + \tilde{\mathbf{w}}_k \mathbf{g}_k^T \\ &= \mathbf{Q}_{k-1} + (\tilde{\mathbf{w}}_k - \mathbf{Q}_{k-1}\overline{\mathbf{w}}_k)\mathbf{g}_k^T.\end{aligned} \tag{4.39}$$

The RLS scheme is now given by:

Initialization:

$$\mathbf{P}_0 = \delta \mathbf{I}_3 \text{ (δ is a small positive number)}$$

$$\mathbf{Q}_0 \in \Re^{P\times 3} \text{ with orthonormal columns}$$

Update equations: for $k = 1, 2, \cdots$

$$\begin{aligned}
a_k &= \frac{1}{P}\sum_{j=1}^{P} u_{kj} \\
b_k &= \frac{1}{P}\sum_{j=1}^{P} v_{kj} \\
\mathbf{x}_k &= \mathbf{u}_k - a_k \mathbf{1}_{P\times 1} \\
\mathbf{y}_k &= \mathbf{v}_k - b_k \mathbf{1}_{P\times 1} \\
\tilde{\mathbf{w}}_k &= [\mathbf{x}_k \;\; \mathbf{y}_k] \\
\overline{\mathbf{w}}_k &= \mathbf{Q}_{k-1}^T \tilde{\mathbf{w}}_k \quad (6P \text{ flops}) \\
\mathbf{g}_k &= \mathbf{P}_{k-1}\overline{\mathbf{w}}_k(\mathbf{I}_2 + \overline{\mathbf{w}}_k^T \mathbf{P}_{k-1}\overline{\mathbf{w}}_k)^{-1} \quad (68 \text{ flops}) \\
\mathbf{P}_k &= \mathbf{P}_{k-1} - \mathbf{g}_k \overline{\mathbf{w}}_k^T \mathbf{P}_{k-1} \quad (45 \text{ flops}) \\
\mathbf{Q}_k &= \mathbf{Q}_{k-1} + (\tilde{\mathbf{w}}_k - \mathbf{Q}_{k-1}\overline{\mathbf{w}}_k)\mathbf{g}_k^T \quad (12P \text{ flops})
\end{aligned}$$

Here $\mathbf{I}_2, \mathbf{I}_3$ are identity matrices, and $\mathbf{1}_{P\times 1} = \underbrace{[1 \cdots 1]}_{P}^T$.

The initial values $\mathbf{P}_0, \mathbf{Q}_0$ should be set carefully to ensure convergence. Since the covariance matrix of $\overline{\mathbf{w}}_k$ is positive definite, $\mathbf{P}_0$ should be set to some symmetric positive definite matrix, for example $\delta \mathbf{I}_3$. $\mathbf{Q}_0$ should have orthonormal columns, a simple choice being the first three columns of the $P \times P$ identity matrix. Clearly, the updating of $\mathbf{Q}_k$ requires a total of $18P$ flops.

4.2.2 Affine Shape and Motion Recovery

After the orthonormal basis of the shape space $\mathbf{Q}_k$ is estimated, the $3 \times P$ affine shape matrix $\hat{\mathbf{S}}_k$ and the 2×3 affine motion matrix $\mathbf{M}_k$ at frame k can be obtained from it. They can further be transformed to the 3D object shape $\mathbf{S}_k$ and camera motion $\mathbf{M}_k$ by a proper 3×3 affine transformation matrix $\mathbf{A}_k$, viz:

$$\mathbf{S}_k = \mathbf{A}_k^{-1}\hat{\mathbf{S}}_k, \quad \mathbf{M}_k = \hat{\mathbf{M}}_k\mathbf{A}_k. \tag{4.40}$$

Determination of $\mathbf{A}_k$ will be discussed in next Chapter. Here we compute $\hat{\mathbf{S}}_k$ and $\hat{\mathbf{M}}_k$.

In Section 3.1, we stated that any basis of the shape space provides a solution for the affine shape matrix. Thus we obtain

$$\hat{\mathbf{S}}_k = \mathbf{Q}_k^T. \tag{4.41}$$

Extracting the k-th pair of rows (corresponding to the k-th frame points) from the registered measurement matrix $\tilde{\mathbf{W}}$ in eq. (3.13), we obtain

$$\tilde{\mathbf{w}}_k^T = \mathbf{M}_k\mathbf{S}_k = \hat{\mathbf{M}}_k\mathbf{A}_k\mathbf{A}_k^{-1}\hat{\mathbf{S}}_k = \hat{\mathbf{M}}_k\mathbf{Q}_k^T. \tag{4.42}$$

Since $\mathbf{Q}_k$ has orthonormal columns, right multiplying eq. (4.42) by $\mathbf{Q}_k$, we obtain

$$\hat{\mathbf{M}}_k = \tilde{\mathbf{w}}_k^T\mathbf{Q}_k. \tag{4.43}$$

4.2.3 Comparison

For a sequence of F image frames where P feature points are tracked, we compare the RLS shape space algorithm with the other FMs.

In the batch FMs, such as the original FM and the paraperspective FM, by computing SVD of the $2F \times P$ measurement matrix after all F frames are available, the shape space is obtained by the first three right singular vectors. If $2F > P$, the SVD computational complexity is $14FP^2 + \frac{11}{3}P^3$ flops. Moreover, the $2F \times P$ measurement matrix, whose size increases with the number of frames, needs to be stored before SVD computation.

In the sequential FM, the power orthogonal iteration method is used to compute the three dominant eigenvectors, which span the shape space, of a $P \times P$ correlation matrix at each frame. The correlation matrix needs to be updated to incorporate new frame information and stored at each frame iteration. Hence, even though the shape space is stationary, the computed dominant eigenvectors of the correlation matrix at each frame do not converge. To make the next step — the computation of the affine transformation matrix $\mathbf{A}_k$ sequential, an extra step of computing a stationary basis of the shape space is required. This is achieved by computing the orthogonal projection matrix of the shape space and its QR decomposition. Thus the total computational cost of the sequential shape space updating algorithm is $26P^2 + 44P$ flops.

In Hwang's 3-D linear combination FM [34], only three rows of the $2F \times P$ measurement matrix are used to compute an orthogonal basis of the shape space. Thus this method loses several advantages of the FMs including even treatment of every frame to get stability, and processing a large number of frames so that the redundancy can reduce the noise effects and provide more accurate results.

Our RLS shape space updating algorithm has several advantages over all the above FMs. First, it computes an estimate of the shape space at each frame recursively as in the sequential FM. Secondly, it is more efficient than the sequential FM both in computation and in memory storage. Its computational complexity is only $18P$ flops at each frame updating and no large measurement or correlation matrix is updated and stored in the algorithm. Thirdly, since it is an adaptive subspace tracking algorithm, once a set of initial values is given, the computed $P \times 3$ matrix $\mathbf{Q}_k$ converges to an orthonormal basis of the shape space. Thus no step of computing a stationary basis is required in order to compute $\mathbf{A}_k$ recursively. Instead, $\mathbf{Q}_k$ can be used as a stationary basis of shape space since it converges after some initial frames. In contrast, the sequential FM needs to compute a stationary basis of the shape space at each frame iteration, because the computed dominant eigenvectors of an updated correlation matrix are different at each frame and do not converge at all. Finally, the RLS algorithm has a similar accuracy to the SVD batch FMs and the sequential FM, which will be demonstrated in the next section.

4.3 Performance Analysis

In this section we describe the synthetic tests used to compare the performance of the RLS shape space updating algorithms with that of SVD and orthogonal iteration algorithms used in the original and the sequential FMs respectively.

4.3.1 Synthetic Data Generation

An object was represented by 100 random points within a cube of predetermined size. The distance of the object centroid from the camera was chosen to be 20 times the side of the cube and was kept fixed throughout the sequence. Camera rotation was specified as given in Figure 4.3 and the object was translated so that its centroid projected on to the principal point of each frame. A sequence of 140 images was generated by projecting the object points onto 512×512 pixel image planes with sub-pixel accuracy using a perspective camera model. The (fixed) focal length was chosen so as to yield good coverage of image points across the image planes. Gaussian noise with a standard deviation of 2 pixels was added to all points in all frames. Quantization errors were not added since we assume that we are able to track features with a subpixel resolution.

4.3.2 Accuracy of the Shape Space Computation

We first discuss accuracy and convergence properties of the RLS algorithms and compare them with the original and the sequential FMs. In order to compare the iterative and batch methods under the same conditions, we perform the following computations. For the iterative methods, such as the RLS and the sequential algorithm, the shape space is updated at each frame iteratively. We obtain $\mathbf{Q}_{r,k}$ for the RLS method, $\mathbf{Q}_{s,k}$ for the sequential FM at frame k. They are all $P \times 3$ matrices whose orthonormal columns span the estimated shape space at frame k. For the batch-type original FM, we form a submatrix $\mathbf{W}_k$, which only contains the feature positions up to frame k. SVD is then applied to this submatrix to compute its first three right eigenvectors $\mathbf{V}_k$ which span the estimated shape space at frame k.

The *shape space estimation error* of each method is represented by the subspace distance between the estimated and the true shape spaces. The errors at frame k are defined as the following:

$$\begin{aligned} E_{r,k} &= dist(range(\mathbf{Q}_{r,k}),\ range(\mathbf{S}^T)) \\ E_{s,k} &= dist(range(\mathbf{Q}_{s,k}),\ range(\mathbf{S}^T)) \\ E_{o,k} &= dist(range(\mathbf{V}_k),\ range(\mathbf{S}^T)) \end{aligned} \tag{4.44}$$

They are errors of the RLS, the sequential and the original methods respectively. The function $dist()$, which represents the subspace distance between two same dimensional subspaces $\mathbf{S}_1,\ \mathbf{S}_2$, is defined by

$$dist(range(\mathbf{V}_1),\ range(\mathbf{V}_2)) = \| \mathbf{V}_1\mathbf{V}_1^T - \mathbf{V}_2\mathbf{V}_2^T \|_2, \tag{4.45}$$

where the columns of $\mathbf{V}_1$, $\mathbf{V}_2$ are orthonormal bases of the two subspaces $\mathbf{S}_1$, $\mathbf{S}_2$, respectively.

Figure 4.4 shows typical convergence for each of the three methods. The methods perform similarly, with shape space being estimated reasonably accurately within 40 frames. After about 35 - 40 frames, the shape space estimation errors of all three methods converge to the same low constant level (about 2.3×10^{-2} in this experiment).

4.3.3 Computational Time

We compare the shape space computational time of the RLS method with that of the original and the sequential FMs. From chapter 2 we know that the computational complexity of the original FM is dominated by the cost of SVD, which is $14FP^2 + 11P^3/3$ flops for P feature points tracked in F frames (assuming $2F \geq P$); that of the sequential FM is $26P^2 + 44P$ flops at each frame for computing the first three dominant eigenvectors using the power iteration technique. On the other hand, through the analysis of the RLS algorithm, its shape space updating complexity for each frame is $18P$ flops. Therefore, theoretically, to compute an estimate of shape space, the original method requires $O(FP^2)$ operations; the sequential method needs $O(P^2)$ operations; while the RLS method takes only $O(P)$.

Our simulation with the synthetic data confirms the above theoretical analysis. Figure 4.5 shows the actual processing time of the three methods on a Pentium PC. The generation of the synthetic data is similar to the description in section 4.3.1. The number of feature points increased from 10 to 200, while the number of frames was fixed at 140. The processing time in Figure 4.5 is only for shape space computation, does not include the feature points generation and tracking time. The results agree with our theoretical analysis. We can see that the RLS method is much more efficient than the other two FMs, especially when P becomes large. Its shape space computational time is almost linear in terms of P.

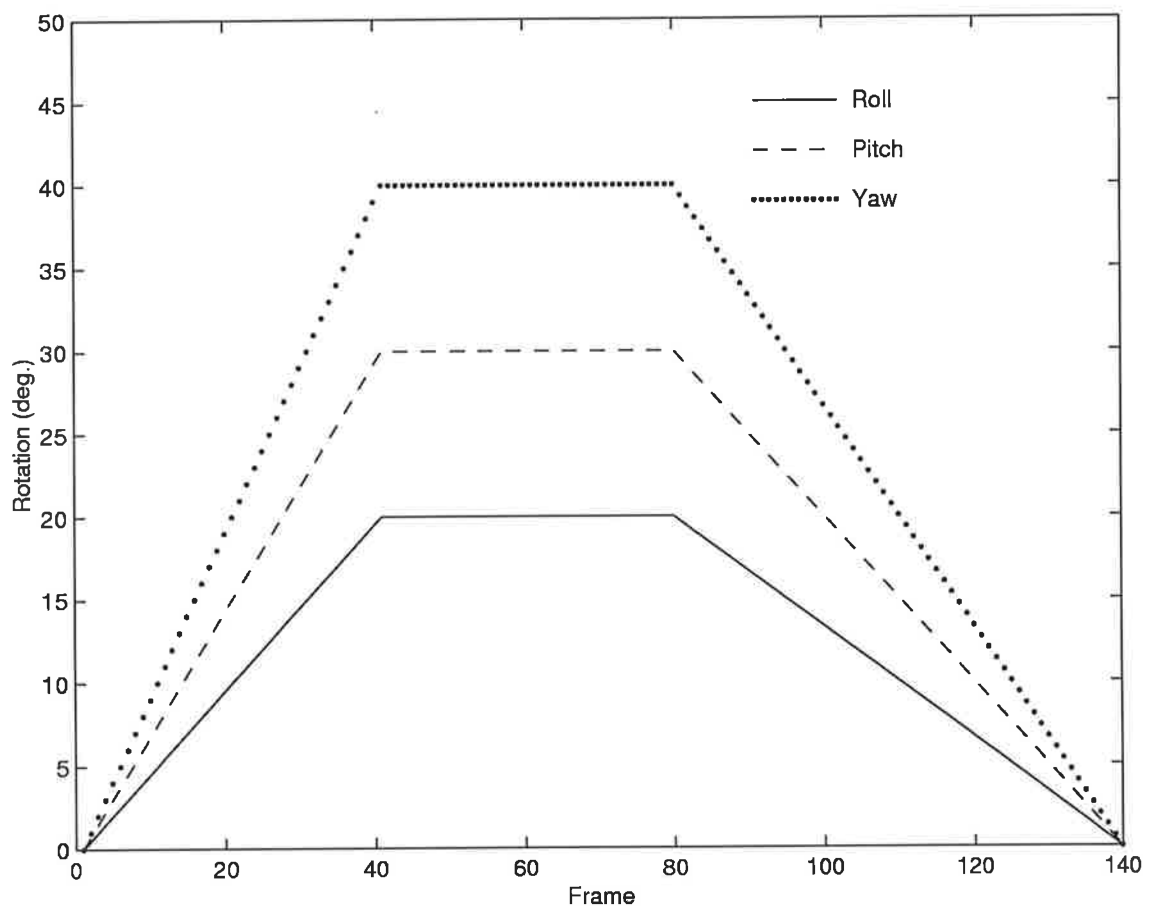

Figure 4.3: The defined camera rotation

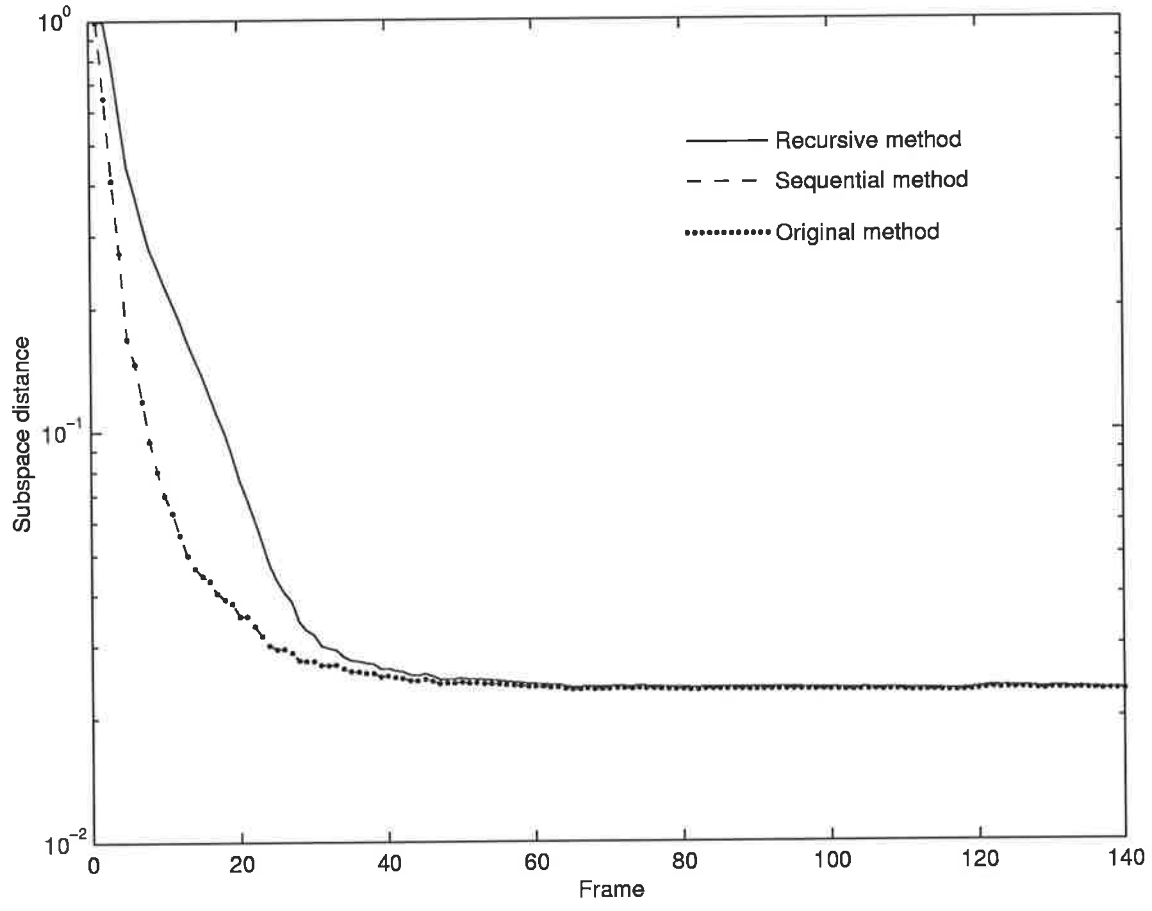

Figure 4.4: Shape space errors

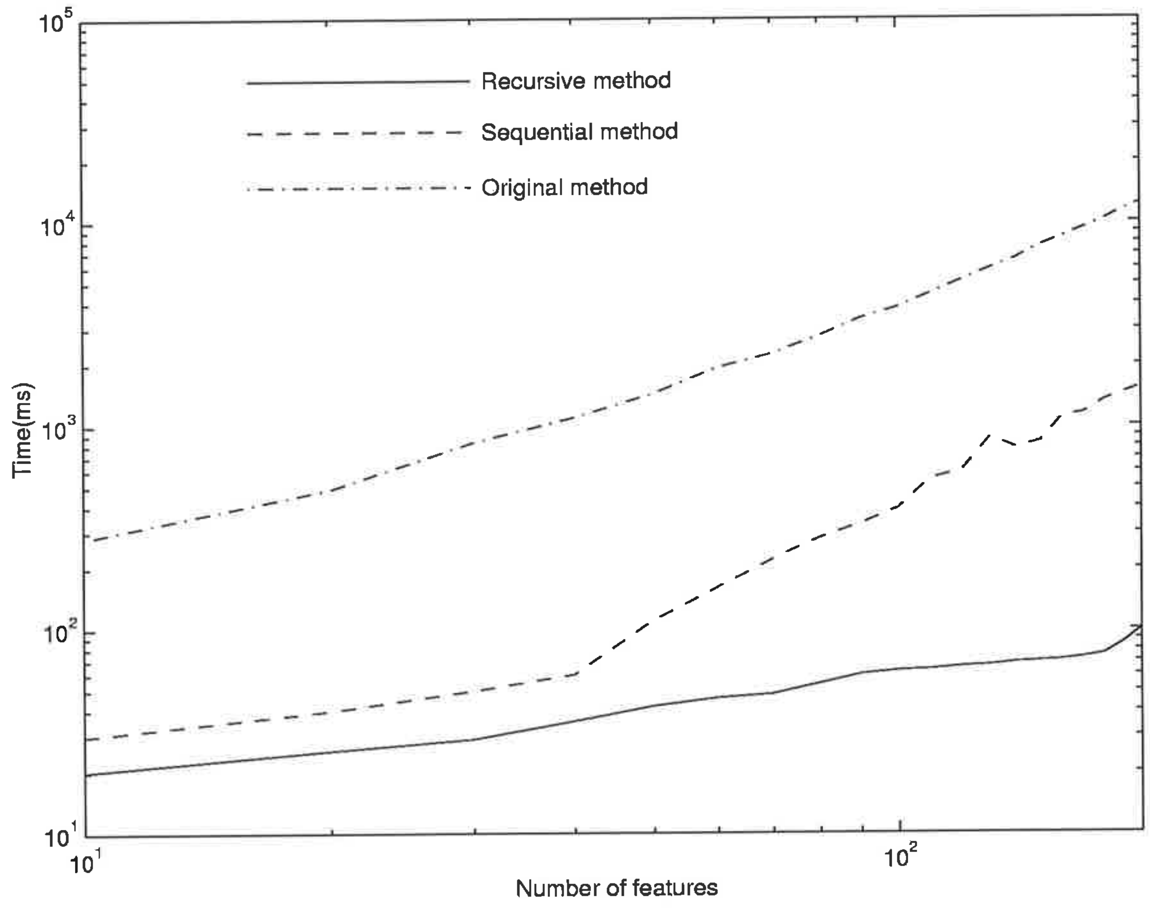

Figure 4.5: Processing time

Chapter 5

Euclidean Reconstruction of Shape and Motion

After affine shape and motion estimates $\hat{\mathbf{S}}_k$, $\hat{\mathbf{M}}_k$ are computed, the next step is to recover Euclidean shape and motion from them. In this chapter, we give the Euclidean shape and motion recovery $\mathbf{S}_k$, $\mathbf{M}_k$ from the estimated shape space at frame k under each of the three affine camera models — orthography, weak perspective and paraperspective. All that is required is to determine a 3×3 affine transformation matrix $\mathbf{A}_k$ under each individual model, such that

$$\mathbf{M}_k = \hat{\mathbf{M}}_k \mathbf{A}_k \tag{5.1}$$

$$\mathbf{S}_k = \mathbf{A}_k^{-1} \hat{\mathbf{S}}_k. \tag{5.2}$$

Preliminary

Under each model, assume that the camera has the following normalized intrinsic parameters: skew $s = 0$, aspect ratio $\gamma = 1$, and focal length $f = 1$. Also assume that the origin of the world coordinate system resides at the object centroid $\mathbf{G}$, whose coordinates in the camera coordinate system of the k-th frame is $[X_k, Y_k, Z_k]^T$. Then the object translation vector relative to the camera system is $\mathbf{t}_k = [X_k, Y_k, Z_k]^T$. The camera rotation matrix at frame k is $\mathbf{R}_k = [\mathbf{i}_k, \mathbf{j}_k, \mathbf{k}_k]^T$, and the k-th image centroid of P feature points is $\mathbf{o}_k = [a_k, b_k]^T$.

5.1 Under Orthography

Under orthography, an object is projected onto the image plane by rays parallel to the optical axis. Orthographic projection is simple and typically a good approximation for perspective imaging when the object's size and changes in depth between frames are small compared to the object's depth from the camera.

In figure 5.1, a 3D point $\mathbf{P} = [X, Y, Z]^T$ is projected orthographically onto the image plane at $\mathbf{p} = [x, y]^T$. It is easy to see that

$$x = X, \quad y = Y. \tag{5.3}$$

Hence the orthographic projection matrix is determined by

$$\mathbf{P}_o = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_k & \mathbf{t}_k \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{i}_k^T & X_k \\ \mathbf{j}_k^T & Y_k \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{5.4}$$

Obviously, the orthographic projection matrix $\mathbf{P}_o$ has the form of $\mathbf{P}_A$ in eq. (3.3). Orthography is therefore an affine model. From eq. (5.4), we obtain the 2×3 affine matrix at frame k as

$$\mathbf{M}_k = \begin{bmatrix} \mathbf{i}_k^T \\ \mathbf{j}_k^T \end{bmatrix}. \tag{5.5}$$

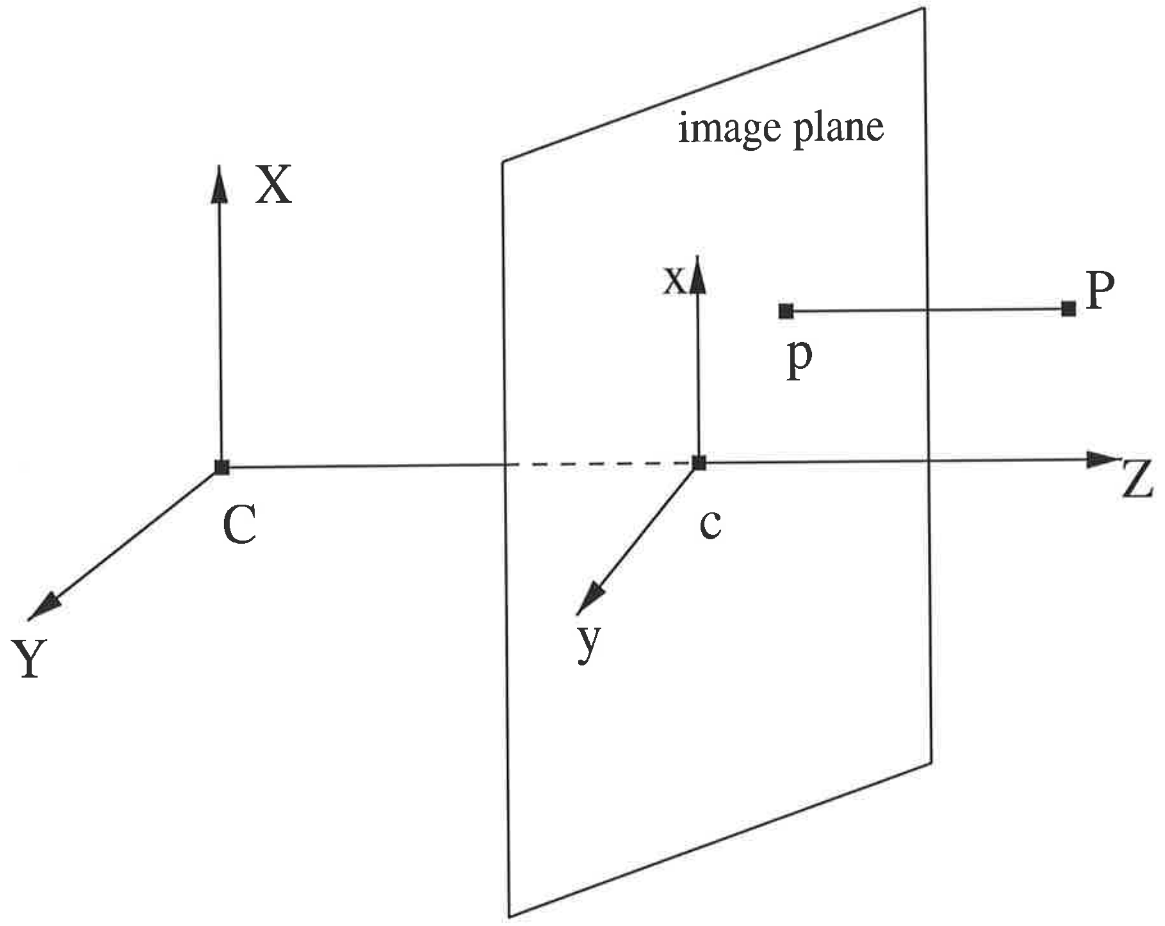

Figure 5.1: Orthographic projection model

$\mathbf{M}_k$ consists of only the first two rows of the rotation matrix $\mathbf{R}_k$ and we call it the camera motion matrix. Thus we obtain the following constraint equations under orthography

$$\mathbf{M}_i\mathbf{M}_i^T = \hat{\mathbf{M}}_i\mathbf{A}_k\mathbf{A}_k^T\hat{\mathbf{M}}_i^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad i = 1, \cdots, k. \tag{5.6}$$

For all frames up to the k-th, eq. (5.6) gives a set of $3k$ linear equations

$$\begin{aligned}
\hat{\mathbf{i}}_i^T \mathbf{A}_k \mathbf{A}_k^T \hat{\mathbf{i}}_i &= 1 \\
\hat{\mathbf{j}}_i^T \mathbf{A}_k \mathbf{A}_k^T \hat{\mathbf{j}}_i &= 1 \quad (i = 1, 2, \cdots, k) \\
\hat{\mathbf{i}}_i^T \mathbf{A}_k \mathbf{A}_k^T \hat{\mathbf{j}}_i &= 0
\end{aligned} \tag{5.7}$$

with respect to the 6 unknown parameters of $\mathbf{L}_k = \mathbf{A}_k \mathbf{A}_k^T$. For $k > 3$, this is an over-determined problem which can be solved by using the linear least-squares technique. Let

$$\mathbf{L}_k = \mathbf{A}_k \mathbf{A}_k^T = \begin{bmatrix} l_1 & l_2 & l_3 \\ l_2 & l_4 & l_5 \\ l_3 & l_5 & l_6 \end{bmatrix}. \tag{5.8}$$

The eqs. (5.7) can be expressed as

$$\mathbf{G}_k \mathbf{l}_k = \mathbf{c}_k, \tag{5.9}$$

where the $3k \times 6$ matrix $\mathbf{G}_k$, the 6×1 vector $\mathbf{l}_k$, and the $3k \times 1$ vector $\mathbf{c}_k$ are defined by:

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{g}^T(\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_1) \\ \vdots \\ \mathbf{g}^T(\hat{\mathbf{i}}_k, \hat{\mathbf{i}}_k) \\ \mathbf{g}^T(\hat{\mathbf{j}}_1, \hat{\mathbf{j}}_1) \\ \vdots \\ \mathbf{g}^T(\hat{\mathbf{j}}_k, \hat{\mathbf{j}}_k) \\ \mathbf{g}^T(\hat{\mathbf{i}}_1, \hat{\mathbf{j}}_1) \\ \vdots \\ \mathbf{g}^T(\hat{\mathbf{i}}_k, \hat{\mathbf{j}}_k) \end{bmatrix}, \quad \mathbf{l}_k = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{bmatrix}, \quad \mathbf{c}_k = \begin{bmatrix} 1 \\ \vdots \\ \vdots \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}1\\1\\1\\1\\1\end{matrix}}\right\} 2k \\ \left.\vphantom{\begin{matrix}0\\0\\0\end{matrix}}\right\} k \end{matrix}, \tag{5.10}$$

and

$$\mathbf{g}^T(\mathbf{a}, \mathbf{b}) = [a_1 b_1 \;\; a_1 b_2 + a_2 b_1 \;\; a_1 b_3 + a_3 b_1 \;\; a_2 b_2 \;\; a_2 b_3 + a_3 b_2 \;\; a_3 b_3]. \tag{5.11}$$

Define

$$\mathbf{D}_k = \mathbf{G}_k^T \mathbf{G}_k \tag{5.12}$$

$$\mathbf{E}_k = \mathbf{G}_k^T \mathbf{c}_k. \tag{5.13}$$

The least-squares solution of the vector $\mathbf{l}_k$ consisting of the 6 unknown entries of $\mathbf{A}_k \mathbf{A}_k^T$ is then

$$\mathbf{l}_k = \mathbf{D}_k^{-1} \mathbf{E}_k. \tag{5.14}$$

In the recursive scheme, the following equations are used to update $\mathbf{D}_k$, $\mathbf{E}_k$ and $\mathbf{l}_k$ at each frame: For $k = 1, \cdots, F$

$$\hat{\mathbf{i}}_k^T = \mathbf{x}_k^T \mathbf{Q}_k, \quad \hat{\mathbf{j}}_k^T = \mathbf{y}_k^T \mathbf{Q}_k$$

$$\mathbf{D}_k = \mathbf{D}_{k-1} + \mathbf{g}(\hat{\mathbf{i}}_k, \hat{\mathbf{i}}_k)\mathbf{g}^T(\hat{\mathbf{i}}_k, \hat{\mathbf{i}}_k) + \mathbf{g}(\hat{\mathbf{j}}_k, \hat{\mathbf{j}}_k)\mathbf{g}^T(\hat{\mathbf{j}}_k, \hat{\mathbf{j}}_k) + \mathbf{g}(\hat{\mathbf{i}}_k, \hat{\mathbf{j}}_k)\mathbf{g}^T(\hat{\mathbf{i}}_k, \hat{\mathbf{j}}_k) \tag{5.15}$$

$$\mathbf{E}_k = \mathbf{E}_{k-1} + \mathbf{g}(\hat{\mathbf{i}}_k, \hat{\mathbf{i}}_k) + \mathbf{g}(\hat{\mathbf{j}}_k, \hat{\mathbf{j}}_k) \tag{5.16}$$

$$\mathbf{l}_k = \mathbf{D}_k^{-1} \mathbf{E}_k$$

Here $\mathbf{D}_0 = \mathbf{0}_{6\times6}$, $\mathbf{E}_0 = \mathbf{0}_{6\times1}$, and $\mathbf{l}_k$ provides the entries for the symmetric matrix $\mathbf{L}_k = \mathbf{A}_k \mathbf{A}_k^T$. $\mathbf{A}_k$ can be obtained by eigendecomposition of $\mathbf{L}_k$. Such a solution of $\mathbf{A}_k$ is determined up to a rotation and reflection. The rotation ambiguity can be removed by aligning the x-axis, y-axis with those of the camera reference system in the first frame. That is to impose the following constraints:

$$\mathbf{i}_1 = [1\ 0\ 0]^T \text{ and } \mathbf{j}_1 = [0\ 1\ 0]^T. \tag{5.17}$$

The reflection ambiguity cannot be removed, because it is an inherent ambiguity of the affine model. However, in real applications, this ambiguity can sometimes be removed by means of some known information of the object (see [60]).

The camera orientation vectors $\mathbf{i}_k$, $\mathbf{j}_k$ and the shape matrix $\mathbf{S}_k$ are then computed by

$$\mathbf{i}_k^T = \hat{\mathbf{i}}_k^T \mathbf{A}_k, \quad \mathbf{j}_k^T = \hat{\mathbf{j}}_k^T \mathbf{A}_k \tag{5.18}$$

$$\mathbf{S}_k = \mathbf{A}_k^{-1}\mathbf{Q}_k^T. \tag{5.19}$$

5.1.1 Synthetic Test

We now show the results of Euclidean reconstruction from the estimated shape space using the RLS algorithm under orthography. We continue the synthetic test in chapter 4. The synthetic data generation is described in section 4.3.1. Since the distance of the object centroid from the camera was fixed at 20 times the size of the object, there is no camera translation along the depth direction. In this experiment, orthography should be a good approximation to the actual perspective projection.

In section 4.3, the performance of the RLS shape space updating algorithm is compared with that of the original and the sequential factorization methods. We see from Figure 4.4 that the shape space converges within 40 frames in all three methods. Here we continue to compute the 3D position of the feature points and the camera orientation vectors in each frame, using the results of the estimated shape space of each method. The accuracy of each method is compared by computing the shape estimation errors and the camera rotation errors.

Shape error is defined as the root-mean-square of the distance between the recovered shape and the true shape, divided by the object size, at each frame, viz:

$$\begin{aligned}
ES_{rec,k} &= [tr\{(\mathbf{S}_{rec}-\mathbf{S})(\mathbf{S}_{rec}-\mathbf{S})^T\}/3P]^{\frac{1}{2}}/size \\
ES_{seq,k} &= [tr\{(\mathbf{S}_{seq}-\mathbf{S})(\mathbf{S}_{seq}-\mathbf{S})^T\}/3P]^{\frac{1}{2}}/size \\
ES_{ori,k} &= [tr\{(\mathbf{S}_{ori}-\mathbf{S})(\mathbf{S}_{ori}-\mathbf{S})^T\}/3P]^{\frac{1}{2}}/size,
\end{aligned} \tag{5.20}$$

where $ES_{rec,k}$, $ES_{seq,k}$, $ES_{ori,k}$ represent the shape errors at the k-th frame of the recursive, the sequential and the original methods respectively. Figure 5.2 shows the shape errors of the three methods . We observe that the three methods perform similarly, with the errors being large at the beginning 20 frames but converge within 40 frames. In this experiment, after 40 frames, the shape space errors attain a constant of 10^{-2}.

Camera rotation errors are defined as the difference between the estimated and the true values for roll, pitch and yaw. The results of each of the methods are compared in Figures 5.3, 5.4, and 5.5. It is shown that the three methods also perform similarly in terms of camera rotation recovery. All the methods are unstable in the early several frames. After that, the errors settle down to within 1 degree throughout the sequence.

We conclude from the above results that the RLS method is as accurate as the original and the sequential FMs. We already know from chapter 4 that the RLS shape space algorithm is more efficient than the other two FMs. Note that the computational cost of Euclidean reconstruction is trivial compared to that of shape space computation. Therefore, the RLS method is advantageous over the other two FMs in real-time applications.

5.2 Under Weak Perspective

When the object makes significant translation along the depth direction in an image sequence, the perspective distance effect cannot be ignored. In this case weak perspective projection, which models the scaling effects caused by depth changes between images, can be used instead of orthography to recover Euclidean shape and motion in the FMs.

Weak perspective, also called scaled orthography, can be understood as a two-step projection: first, all the object feature points are projected onto the average depth plane (the plane through the object centroid and parallel to the image plane) by the rays parallel to the optical axis, then this plane is perspectively projected onto the image plane. The second step is actually a uniform scaling of the average depth plane. Figure 5.6 illustrates the weak perspective model.

In Figure 5.6, a 3D point $\mathbf{P} = [X, Y, Z]^T$ is projected to the image plane at $\mathbf{p} = [x, y]^T$. It is easy to verify that

$$x = \frac{X}{Z_k}, \quad y = \frac{Y}{Z_k}, \tag{5.21}$$

Thus the weak perspective projection matrix under the object-centered world coordinate system has the form

$$\mathbf{P}_{wp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & Z_k \end{bmatrix} \begin{bmatrix} \mathbf{R}_k & \mathbf{t}_k \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{i}_k^T & X_k \\ \mathbf{j}_k^T & Y_k \\ \mathbf{0}_{1\times 3} & Z_k \end{bmatrix}. \tag{5.22}$$

Hence weak perspective projection is also a kind of affine model. The 2×3 affine matrix (also called motion matrix) at frame k is then

$$\mathbf{M}_k = \frac{1}{Z_k} \begin{bmatrix} \mathbf{i}_k^T \\ \mathbf{j}_k^T \end{bmatrix}. \tag{5.23}$$

It involves the average depth Z_k and the camera orientation vectors $\mathbf{i}_k$, $\mathbf{j}_k$.

A set of constraint equations for determining $\mathbf{A}_k$ under the weak perspective model are then

$$\mathbf{M}_i\mathbf{M}_i^T = \hat{\mathbf{M}}_i\mathbf{A}_k\mathbf{A}_k^T\hat{\mathbf{M}}_i^T = \frac{1}{Z_k^2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad i = 1, \cdots, k. \tag{5.24}$$

Let $\mathbf{m}_k^T$, $\mathbf{n}_k^T$ be two rows of $\mathbf{M}_k$, that is, $\mathbf{M}_k = [\mathbf{m}_k \ \mathbf{n}_k]^T$. For all frames up to the

k-th, eq. (5.24) gives the $3k$ equations

$$\begin{aligned}\hat{\mathbf{m}}_i^T \mathbf{A}_k \mathbf{A}_k^T \hat{\mathbf{m}}_i =& \frac{1}{Z_k^2} \\ \hat{\mathbf{n}}_i^T \mathbf{A}_k \mathbf{A}_k^T \hat{\mathbf{n}}_i =& \frac{1}{Z_k^2} \qquad (i = 1, 2, \cdots, k). \\ \hat{\mathbf{m}}_i^T \mathbf{A}_k \mathbf{A}_k^T \hat{\mathbf{n}}_i =& 0\end{aligned} \tag{5.25}$$

Then we obtain the following $2k$ linear equations

$$\begin{aligned}(\hat{\mathbf{m}}_i - \hat{\mathbf{n}}_i)^T \mathbf{A}_k \mathbf{A}_k^T (\hat{\mathbf{m}}_i - \hat{\mathbf{n}}_i) =& 0 \\ \hat{\mathbf{m}}_i^T \mathbf{A}_k \mathbf{A}_k^T \hat{\mathbf{n}}_i =& 0,\end{aligned} \qquad (i = 1, 2, \cdots, k), \tag{5.26}$$

with respect to the 6 unknown parameters of $\mathbf{L}_k = \mathbf{A}_k \mathbf{A}_k^T$. To avoid the trial solution $\mathbf{M}_k = 0$, we add an additional constraint $\|\mathbf{m}_1\| = 1$, so that

$$\hat{\mathbf{m}}_1^T \mathbf{A}_k \mathbf{A}_k^T \hat{\mathbf{m}}_1 = 1. \tag{5.27}$$

For $k > 3$, the above over-determined linear system can be solved by LLS as in the following.

$\mathbf{L}_k$ is assumed as in eq. (5.8). The constraint eqs. (5.26), (5.27) are combined into one matrix equation

$$\mathbf{G}_k \mathbf{l}_k = \mathbf{c}_k, \tag{5.28}$$

where the $(2k+1) \times 6$ matrix $\mathbf{G}_k$, the 6×1 vector $\mathbf{l}_k$, and the $(2k+1) \times 1$ vector $\mathbf{c}_k$ are defined by:

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{g}^T(\hat{\mathbf{m}}_1 - \hat{\mathbf{n}}_1, \hat{\mathbf{m}}_1 - \hat{\mathbf{n}}_1) \\ \vdots \\ \mathbf{g}^T(\hat{\mathbf{m}}_k - \hat{\mathbf{n}}_k, \hat{\mathbf{m}}_k - \hat{\mathbf{n}}_k) \\ \mathbf{g}^T(\hat{\mathbf{m}}_1, \hat{\mathbf{n}}_1) \\ \vdots \\ \mathbf{g}^T(\hat{\mathbf{m}}_k, \hat{\mathbf{n}}_k) \\ \mathbf{g}^T(\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_1) \end{bmatrix}, \quad \mathbf{l}_k = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{bmatrix}, \quad \mathbf{c}_k = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{5.29}$$

and $\mathbf{g}^T(\mathbf{a},\ \mathbf{b})$ is defined in eq. (5.11). Then the least-squares solution of $\mathbf{l}_k$ is given by

$$\mathbf{l}_k = \mathbf{D}_k^{-1}\mathbf{g}(\hat{\mathbf{m}}_1,\ \hat{\mathbf{m}}_1) \tag{5.30}$$

$$\mathbf{D}_k = \mathbf{G}_k^T\mathbf{G}_k. \tag{5.31}$$

In the recursive scheme, the following equations are used to update $\mathbf{D}_k$ and $\mathbf{l}_k$ at each frame:

For $k = 1, \cdots, F$

$$\begin{aligned}
\hat{\mathbf{m}}_k^T &= \mathbf{x}_k^T\mathbf{Q}_k, \quad \hat{\mathbf{n}}_k^T = \mathbf{y}_k^T\mathbf{Q}_k \\
\mathbf{D}_k &= \mathbf{D}_{k-1} + \mathbf{g}(\hat{\mathbf{m}}_k - \hat{\mathbf{n}}_k,\ \hat{\mathbf{m}}_k - \hat{\mathbf{n}}_k)\mathbf{g}^T(\hat{\mathbf{m}}_k - \hat{\mathbf{n}}_k,\ \hat{\mathbf{m}}_k - \hat{\mathbf{n}}_k) \\
&\quad + \mathbf{g}(\hat{\mathbf{m}}_k, \hat{\mathbf{n}}_k)\mathbf{g}^T(\hat{\mathbf{m}}_k, \hat{\mathbf{n}}_k) \\
\mathbf{l}_k &= \mathbf{D}_k^{-1}\mathbf{g}(\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_1),
\end{aligned} \tag{5.32}$$

where the initial value is

$$\mathbf{D}_0 = \mathbf{g}(\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_1)\mathbf{g}^T(\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_1). \tag{5.33}$$

$\mathbf{l}_k$ provides the entries for the symmetric matrix $\mathbf{L}_k = \mathbf{A}_k\mathbf{A}_k^T$. $\mathbf{A}_k$ can be obtained by eigendecomposition of $\mathbf{L}_k$.

The camera motion vectors $\mathbf{m}_k$, $\mathbf{n}_k$ and the shape matrix $\mathbf{S}_k$ are computed by

$$\mathbf{m}_k^T = \hat{\mathbf{m}}_k^T\mathbf{A}_k, \quad \mathbf{n}_k^T = \hat{\mathbf{n}}_k^T\mathbf{A}_k \tag{5.34}$$

$$\mathbf{S}_k = \mathbf{A}_k^{-1}\mathbf{Q}_k^T. \tag{5.35}$$

The camera orientation vectors $\mathbf{i}_k$, $\mathbf{j}_k$ are then computed as

$$\mathbf{i}_k^T = \frac{\mathbf{m}_k^T}{||\mathbf{m}_k||}, \quad \mathbf{j}_k^T = \frac{\mathbf{n}_k^T}{||\mathbf{n}_k||}, \tag{5.36}$$

We note from eq. (5.24) that under the weak perspective model, the average depth Z_k of the object at each frame can be determined as

$$Z_k = \left[\frac{1}{2}\left(\frac{1}{||\mathbf{m}_k||^2} + \frac{1}{||\mathbf{n}_k||^2}\right)\right]^{\frac{1}{2}}. \tag{5.37}$$

In contrast, object depth cannot be recovered under orthography.

5.3 Under Paraperspective

Paraperspective projection, which models both perspective distance and position effects, can be used instead of orthographic and scaled orthographic in the situation where objects have significant translation across the image plane. It can also be understood as a two-step projection (see Section 2.3.1). The difference between this model and scaled orthography is the first step: Instead of using rays parallel to the optical axis, the object points are projected onto the average depth plane using rays parallel to the centroid perspective projecting ray. The model is illustrated in Figure 5.7, where a 3D object point $\mathbf{P} = [X, Y, Z]^T$ is projected onto the image plane at $\mathbf{p} = [x, y]^T$ by the paraperspective model, and $\mathbf{G} = [X_k, Y_k, Z_k]^T$ is the object centroid under the k-th camera reference system.

$\mathbf{P}$ is first projected, parallel to CG, to the average depth plane at $[X - \frac{X_k}{Z_k}Z + X_k,\ Y - \frac{Y_k}{Z_k}Z + Y_k,\ Z_k]^T$. Finally, this point is projected perspectively onto the image plane as

$$\begin{aligned} x &= \frac{1}{Z_k}(X - \frac{X_k}{Z_k}Z + X_k) \\ y &= \frac{1}{Z_k}(Y - \frac{Y_k}{Z_k}Z + Y_k). \end{aligned} \tag{5.38}$$

Thus the paraperspective projection matrix under the world coordinate system is given by

$$\mathbf{P}_p = \begin{bmatrix} 1 & 0 & -\frac{X_k}{Z_k} & 0 \\ 0 & 1 & -\frac{Y_k}{Z_k} & 0 \\ 0 & 0 & 0 & Z_k \end{bmatrix} \begin{bmatrix} \mathbf{R}_k & \mathbf{t}_k \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{i}_k^T - \frac{X_k}{Z_k}\mathbf{k}_k^T & X_k \\ \mathbf{j}_k^T - \frac{Y_k}{Z_k}\mathbf{k}_k^T & Y_k \\ \mathbf{0}_{1\times 3} & Z_k \end{bmatrix}. \tag{5.39}$$

Since $\mathbf{P}_p$ has the form of $\mathbf{P}_A$ in eq. (3.3), paraperspective is also a kind of affine camera model. The k-th image center coordinates $\mathbf{o}_k = [a_k,\ b_k]^T = [\frac{X_k}{Z_k},\ \frac{Y_k}{Z_k}]^T$, and the k-th affine matrix (called the motion matrix here) is obtained from $\mathbf{P}_p$ as the following

$$\mathbf{M}_k = \frac{1}{Z_k}\begin{bmatrix} \mathbf{i}_k^T - \frac{X_k}{Z_k}\mathbf{k}_k^T \\ \mathbf{j}_k^T - \frac{Y_k}{Z_k}\mathbf{k}_k^T \end{bmatrix} = \frac{1}{Z_k}\begin{bmatrix} \mathbf{i}_k^T - a_k\mathbf{k}_k^T \\ \mathbf{j}_k^T - b_k\mathbf{k}_k^T \end{bmatrix}. \tag{5.40}$$

Since $\mathbf{i}_k, \mathbf{j}_k, \mathbf{k}_k$ are orthonormal vectors, we have

$$\mathbf{M}_i\mathbf{M}_i^T = \hat{\mathbf{M}}_i\mathbf{A}_k\mathbf{A}_k^T\hat{\mathbf{M}}_i^T = \frac{1}{Z_k^2}\begin{bmatrix} 1+a_i^2 & a_ib_i \\ a_ib_i & 1+b_i^2 \end{bmatrix},\ i = 1, \cdots, k. \tag{5.41}$$

These are the metric constraints under the paraperspective model. For $k > 3$, $2k$ over-constrained linear equations in terms of 6 unknowns of $\mathbf{L}_k = \mathbf{A}_k\mathbf{A}_k^T$ can be derived from eq. (5.41). Then $\mathbf{L}_k, \mathbf{A}_k, \mathbf{S}_k, \mathbf{M}_k$ are determined in a similar way as under orthography. Since the k-th image centroid $\mathbf{o}_k = [a_k,\ b_k]^T$ is known from the RLS algorithm (chapter 4), after $\mathbf{M}_k$ is computed, the camera orientation vectors $\mathbf{i}_k, \mathbf{j}_k, \mathbf{k}_k$ can be computed by using eq. (5.40), and the average depth Z_k is also obtained by eq. (5.41). For details refer to Section 2.3.3 — paraperspective motion recovery.

5.3.1 Synthetic Test

We now compare the performance of Euclidean reconstruction under the orthographic, weak perspective and paraperspective camera models. Prior to Euclidean reconstruction at each frame, shape space is first estimated by using the RLS shape space tracking algorithm.

Data Generation

A set of 100 feature points were randomly chosen within a cube of side d. 77 synthetic image sequences were generated, with the initial object depths (the distance from the camera's optical center to the object average depth plane in the first frame) varying from 4 to 80 times the cube side d. In each sequence, 60 frames were produced by projecting the feature points onto a 512×512 image plane, using a perspective camera model. The camera rotation was predefined as in Figure 5.8. Throughout each sequence, the object translation components along the image axes were both d, and that along the optical axis was half its initial distance away from the camera. In all three axes directions, the object translated evenly. The coordinates in the image frames were perturbed by adding Gaussian noise with standard deviation 2.

Performance Comparison

At each sequence, we computed the shape space using the RLS algorithm. Since computation of the shape space normally converges within 40 frames and then remains within a small constant error, the estimated shape space at the last frame should be the most accurate one in the sequence. Accordingly, we used this frame to carry out Euclidean reconstruction under each of the three camera models. Shape estimation error was defined in the same way as in Section 5.1.1. Figure 5.9 shows the shape errors with respect to the initial depth in each of the sequences under each of the three models. We see that paraperspective reconstruction is much better than orthographic reconstruction over the entire range of depths. This is because orthographic projection cannot model the perspective scaling effect, which is caused by the translation along the depth direction. We also note that the performance of the paraperspective model is better than that of the weak perspective model at close range. As the object becomes more distant from the camera, the performance difference between the weak perspective and paraperspective models gets smaller. This confirms the importance of modeling the perspective position effect when objects are close to the camera.

5.4 Real Image Experiments

In order to test the accuracy and applicability of the recursive method, we produced a sequence of 120 real images of a grid with a Pulnix TM-1000 progressive scan camera. The angle between the two grid planes was kept at 90 degrees over the whole sequence. In acquiring the sequence, the camera was rotated by hand around the scene. A stream of 120 frames was grabbed at a rate of 15 frames per second. The frames 1, 70, 120 are shown in Figures 5.10, 5.11, and 5.12 respectively.

A total of 141 feature points were selected and tracked using a corner detector [23] with sub-pixel accuracy. Figures 5.13, 5.14, 5.15 show the tracked feature points in frames 1, 70, 120 respectively.

When producing the image sequence, a large rotation was employed while very little translation was used, both along the depth direction and across the scene. Thus we chose the orthographic model to do the Euclidean reconstruction in the RLS method. Application of the recursive method yields a good 3D reconstruction as indicated in Figures 5.16, 5.17, and 5.18. We can see from Figure 5.18 that the coplanarity of points is well preserved, and the angle between the two planes is reasonably accurate.

The results of this experiment demonstrate that the recursive method works well with real images. Its accuracy and applicability are also confirmed. Since the RLS method is more efficient than the other FMs, it is suitable to be used in real-time applications.

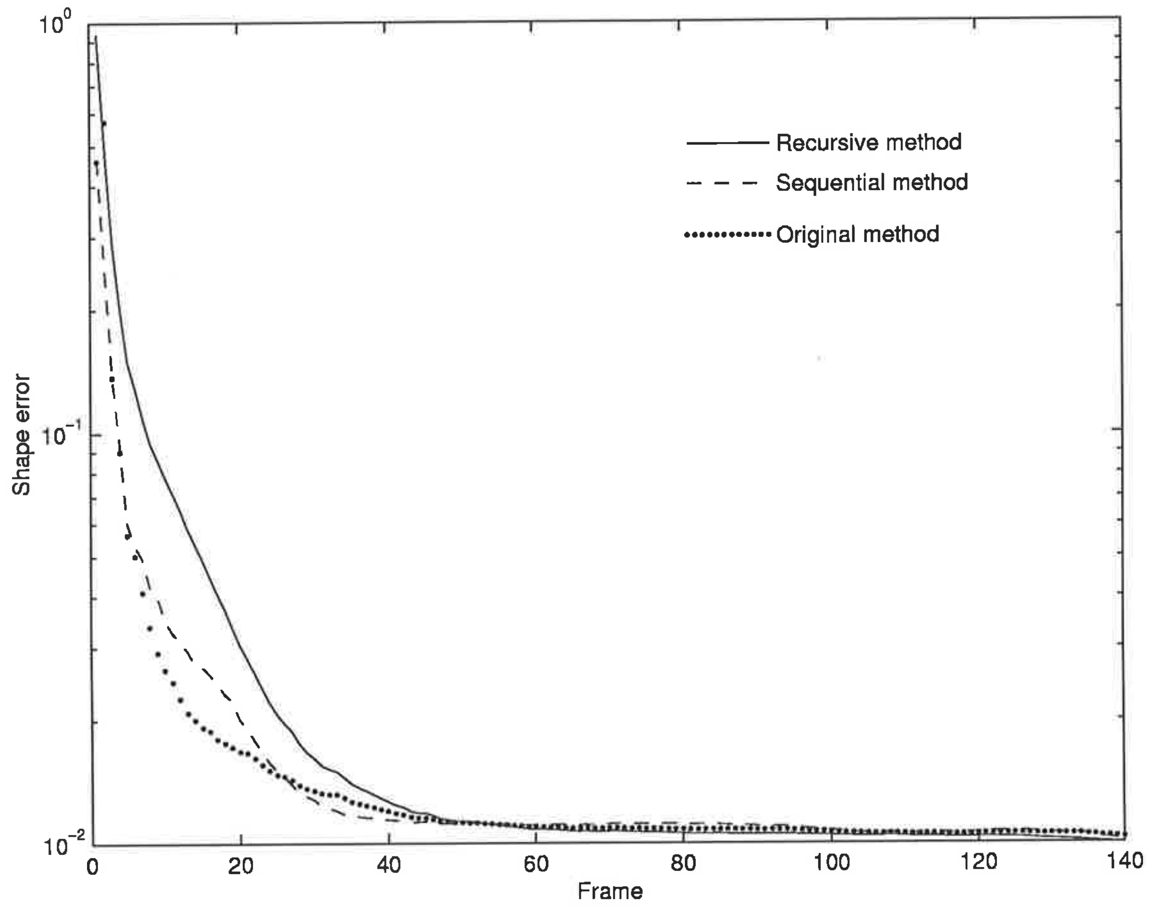

Figure 5.2: Shape errors

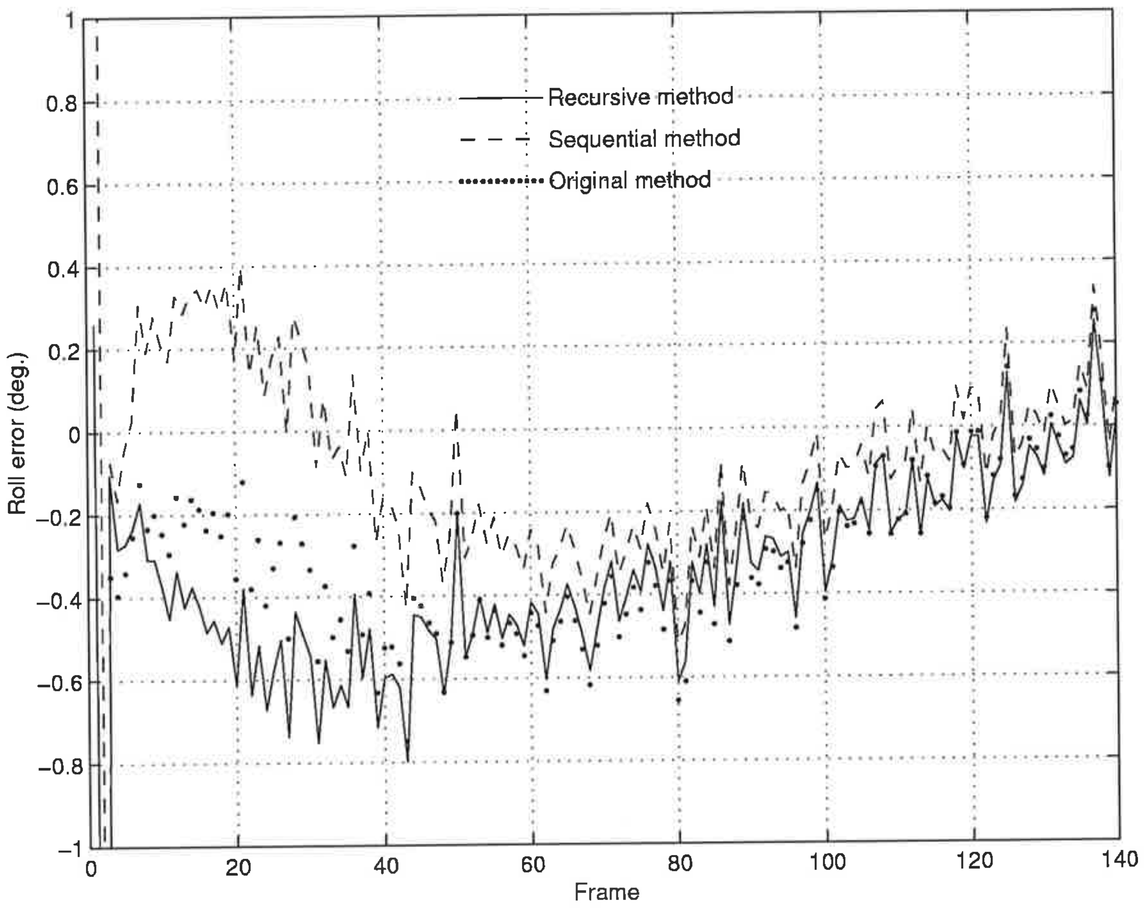

Figure 5.3: Roll errors

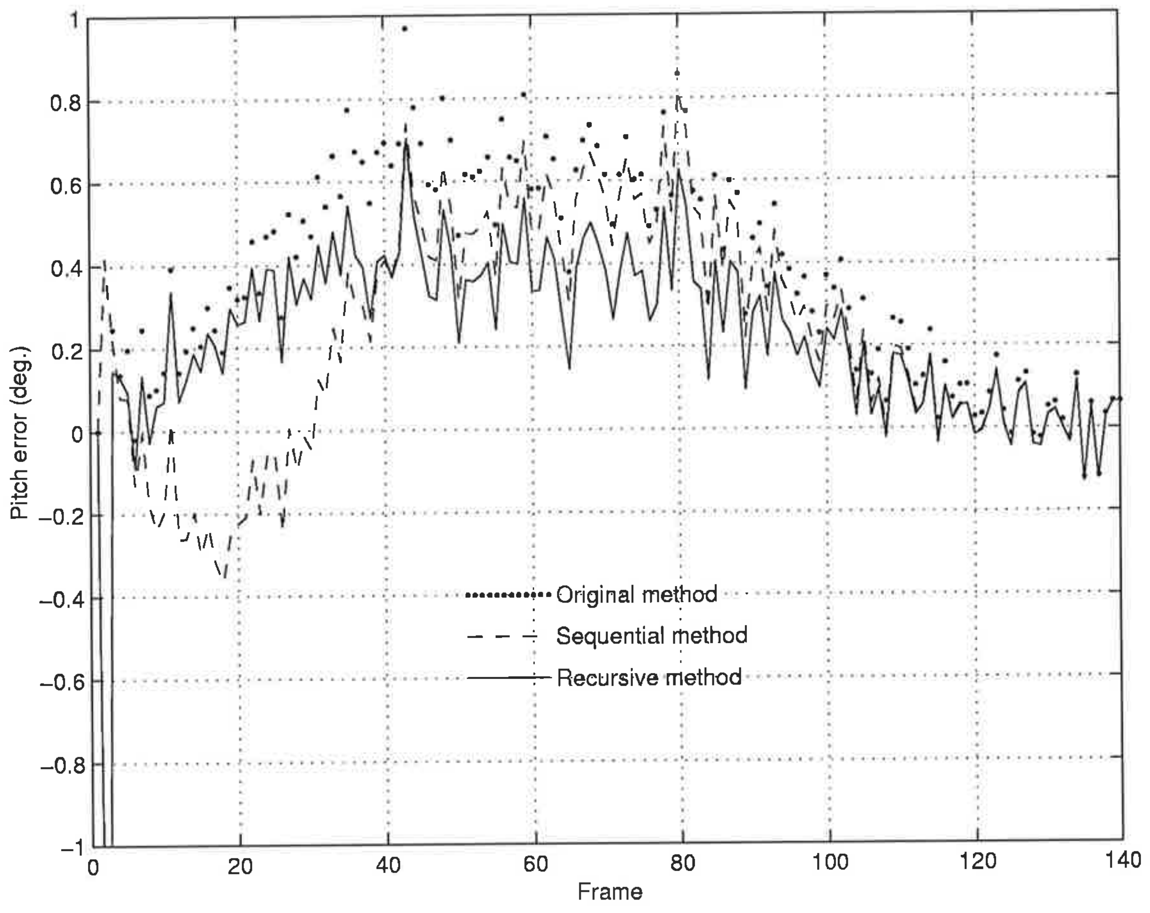

Figure 5.4: Pitch errors

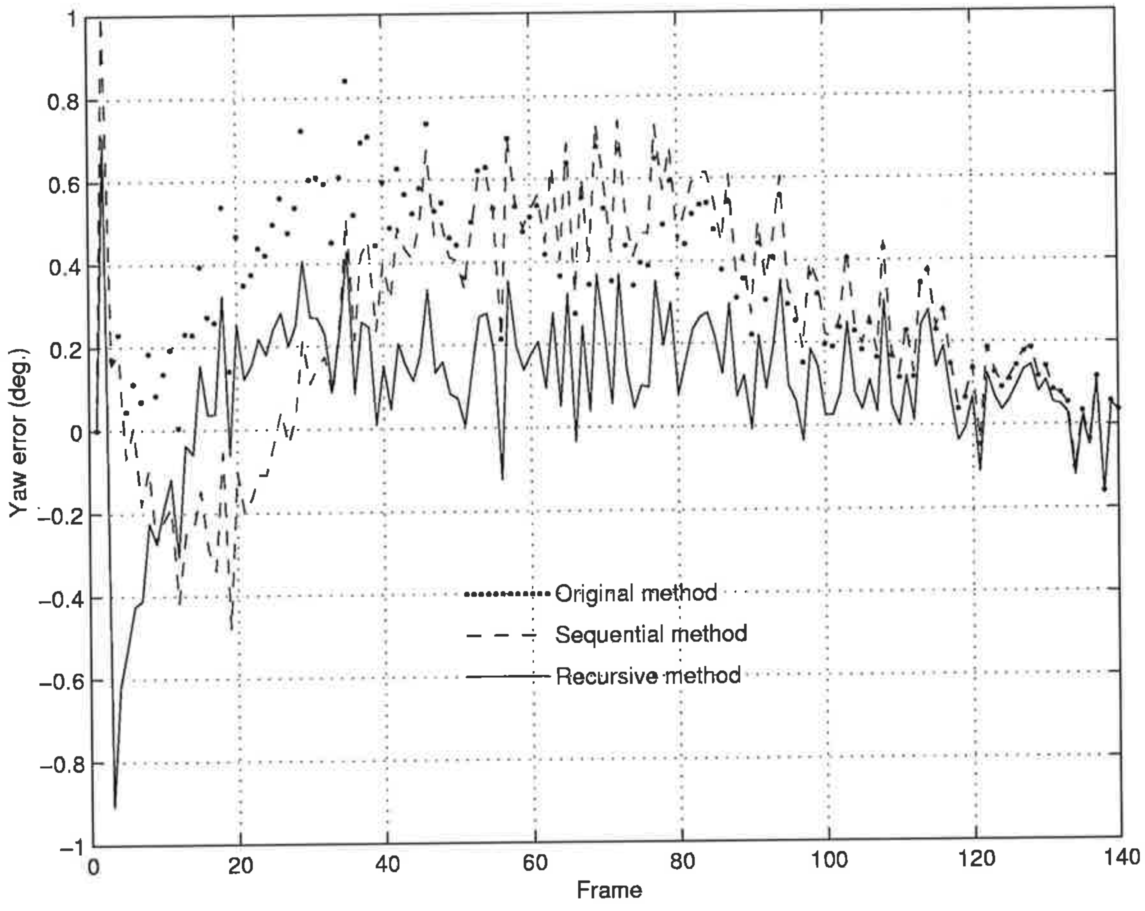

Figure 5.5: Yaw errors

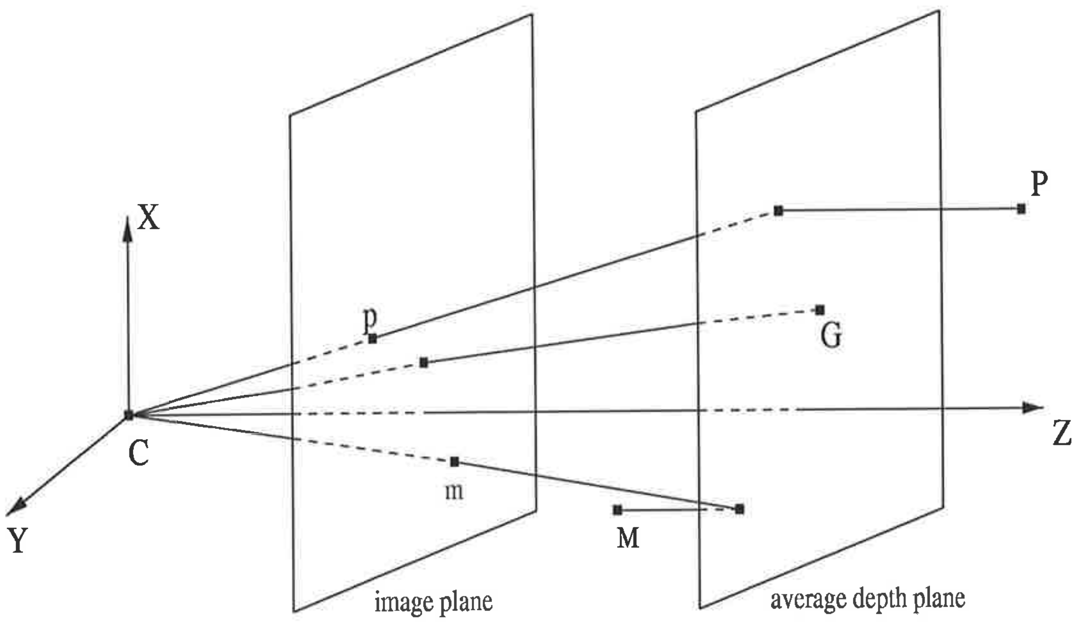

Figure 5.6: Weak perspective projection model

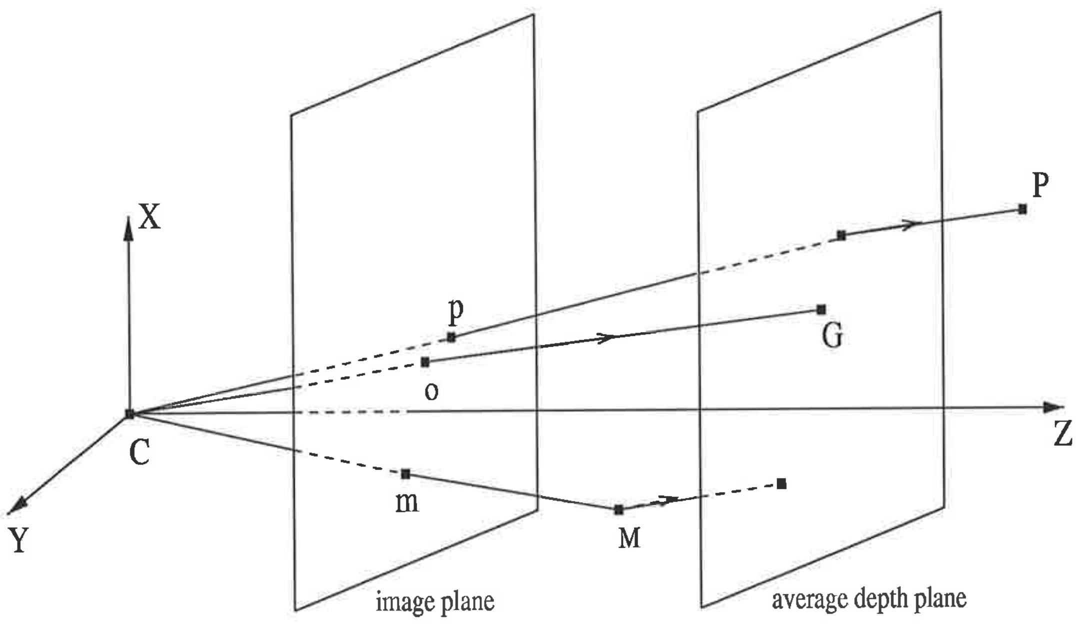

Figure 5.7: Paraperspective projection model

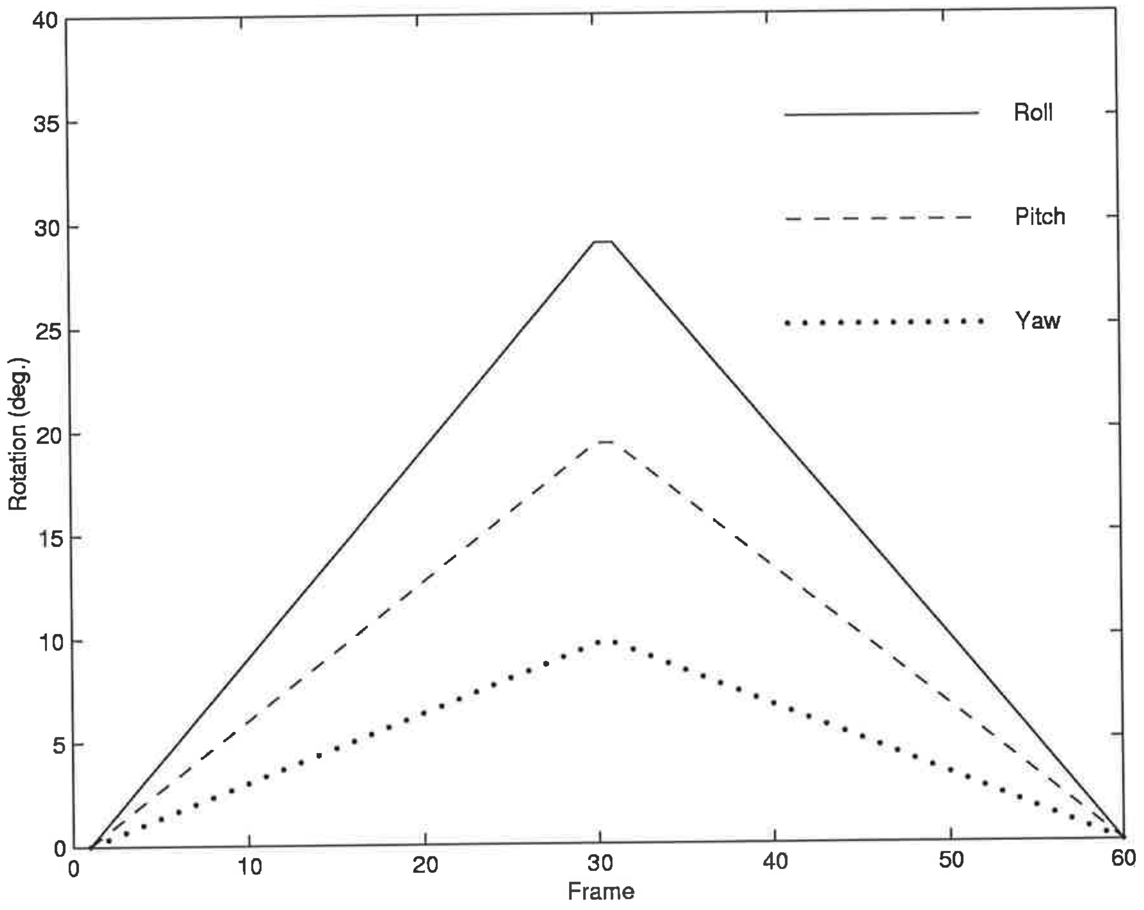

Figure 5.8: The predefined camera rotation

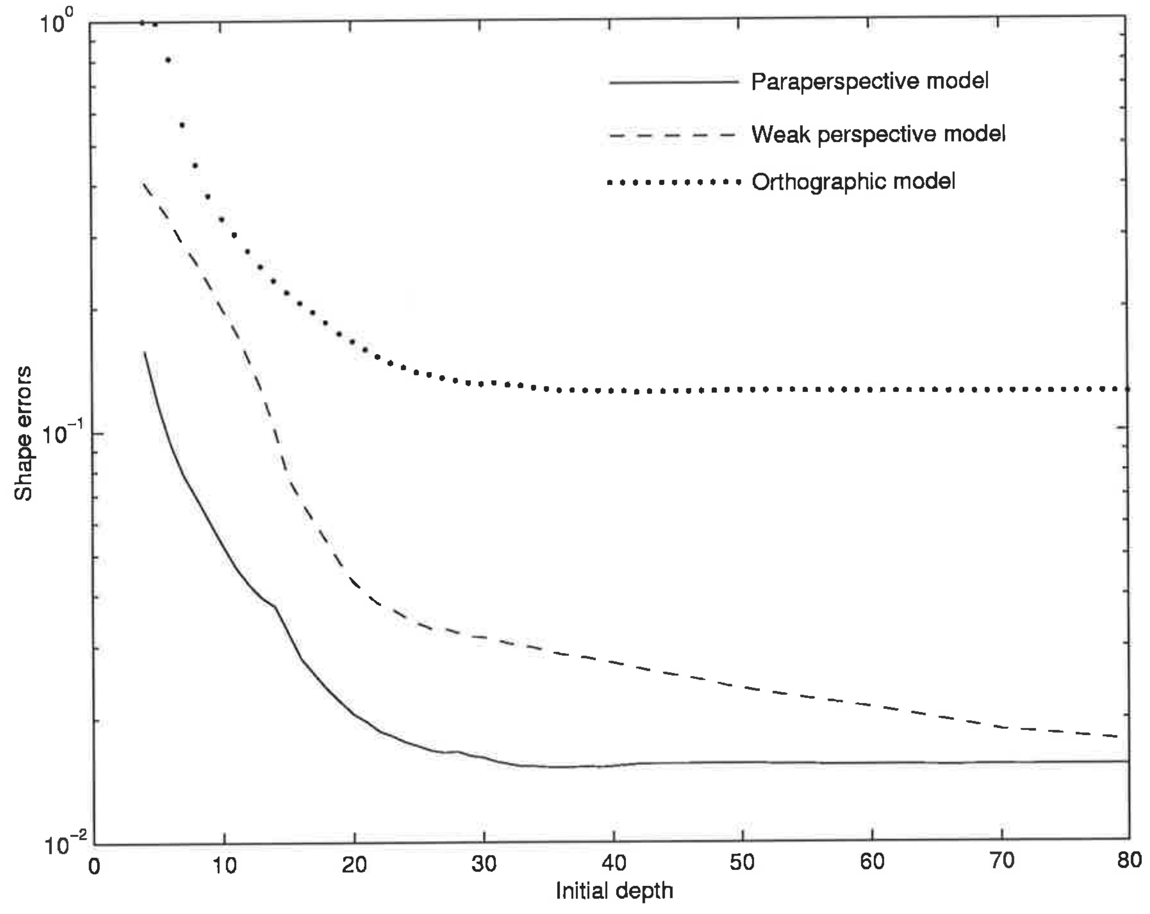

Figure 5.9: Shape errors under the three affine models

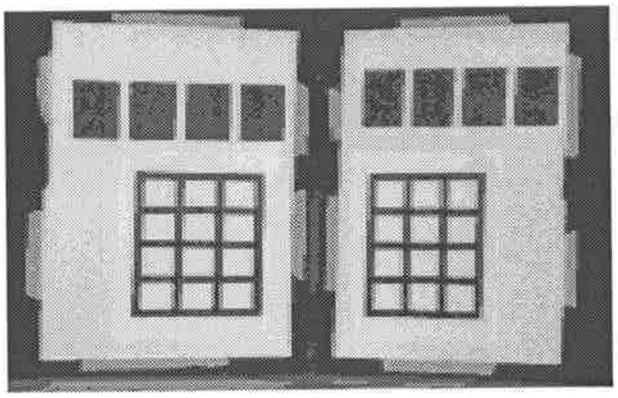

Figure 5.10: The first frame of the grid sequence

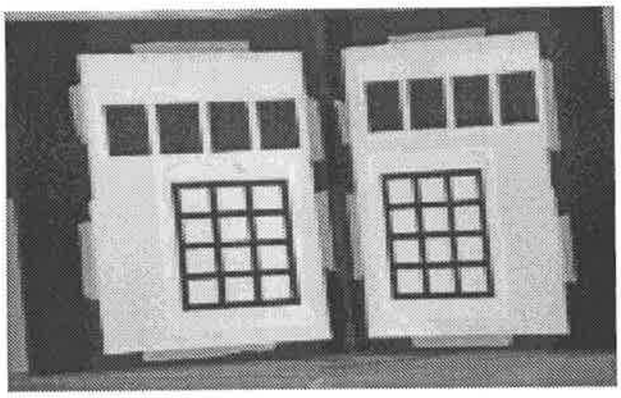

Figure 5.11: Frame 70 of the grid sequence

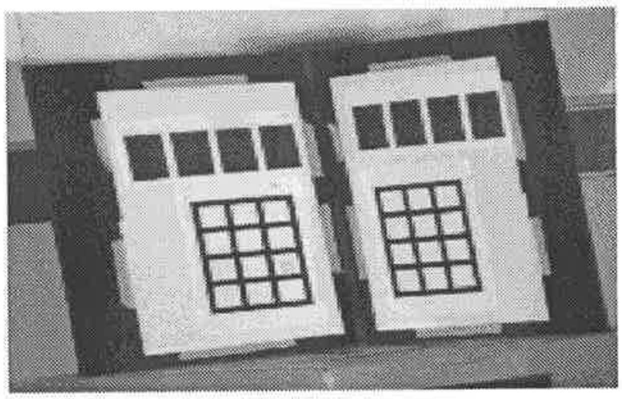

Figure 5.12: Frame 120 of the grid sequence

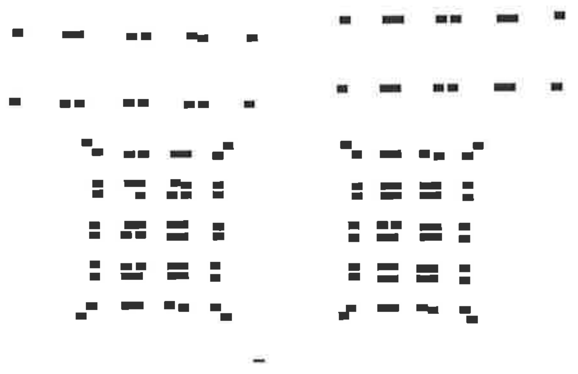
Figure 5.13: Tracked points in the first frame

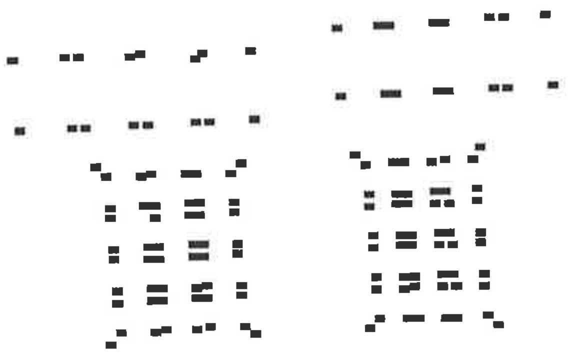
Figure 5.14: Tracked points in frame 70

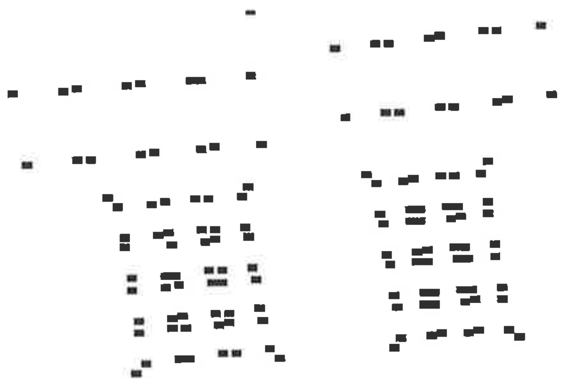
Figure 5.15: Tracked points in frame 120

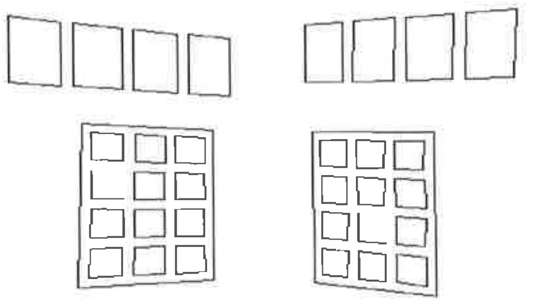
Figure 5.16: Front view

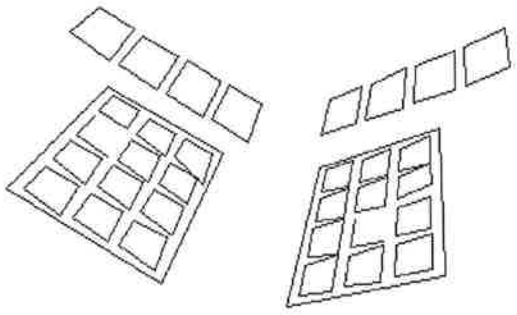
Figure 5.17: Oblique view

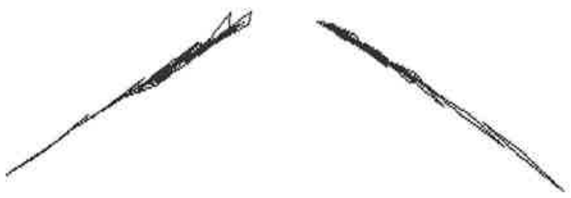
Figure 5.18: Top view

Chapter 6

The Extended RLS Algorithm for Occlusion

So far, we have assumed that all of the P feature points are visible and have been tracked throughout the sequence. In real applications, this is not always possible. Sometimes feature points are invisible and are not tracked in some of the frames because they leave the field of view or become occluded. The occlusion phenomena are so frequent that it makes an SFM method unrealistic if it cannot deal with them. Hence, in this chapter, we extend the RLS algorithm to accommodate occlusion concerns. Compared to the occlusion solutions presented in other FMs, our extended RLS method is simple and efficient. Simulation results also confirm its good performance.

6.1 The Extended RLS Algorithm

In an image sequence, frames with missing or occluded features result in the input vectors to the RLS algorithm being only partially filled in. However, we can always estimate the image coordinates of these invisible features by using all those visible feature points at the frame and their 3D positions recovered at the previous frame. Then the estimated and the tracked image data are used together to form the input vectors to the RLS algorithm to update the shape space.

The recovery of the invisible feature points in a frame is based on the following *condition for reconstruction* rules [71].

Condition for Reconstruction: In the absence of noise, an unknown image measurement pair (u_{fp}, v_{fp}) in frame f can be reconstructed if point p is visible in at least three more frames $f_1,\ f_2,\ f_3$, and if there are at least three more points $p_1,\ p_2,\ p_3$, that are visible in all the four frames $f_1,\ f_2,\ f_3,\ f$.

In addition, we know from chapter 5 that Euclidean shape and motion can be recovered only after some initial frames (normally $k > 3$). Therefore, we assume that occlusion occurs only at frame $k > 4$, the occluded points must be visible in at least three previous frames $k_1, k_2, k_3 < k$, and there are more than 3 feature points not occluded at frames k_1, k_2, k_3, k. This assumption is generally satisfied in practice.

Assume that m $(m < P)$ feature points are invisible and not tracked at the k-th frame $(k > 4)$. We show in the following that their image locations can be estimated by using $P - m$ tracked feature points at the frame and the 3D positions recovered at the previous frame $k-1$. Without losing generality, we exchange the feature indexes so that $1, \cdots, m$ represent the m occluded points, and $m+1, \cdots, P$ represent the $P-m$ tracked points in the frame. Combining eq. (3.12) for the $P-m$ tracked feature points in frame k, the following equation is derived

$$\begin{aligned}
[\tilde{\mathbf{w}}'_{k,m+1} \cdots \tilde{\mathbf{w}}'_{k,P}] &= \mathbf{M}_k[\mathbf{s}'_{k,m+1} \cdots \mathbf{s}'_{k,P}] \\
&= \mathbf{M}'_k[\mathbf{s}'_{k-1,m+1} \cdots \mathbf{s}'_{k-1,P}],
\end{aligned}$$

$$\text{or in short} \quad \tilde{\mathbf{w}}'_k = \mathbf{M}'_k \mathbf{S}'_{k-1}, \tag{6.1}$$

where $\tilde{\mathbf{w}}'_{k,j}$ $(j = m+1, \cdots, P)$ are the registered image coordinates with respect to $\mathbf{o}'_k = [a'_k, b'_k]^T$ (the centroid of $P-m$ tracked feature points at frame k). Also $\mathbf{s}'_{k,j}$ $(j = m+1, \cdots, P)$ represents the 3D recovery of the j-th point at frame k referring to $\mathbf{O}'_k$ (the centroid of the corresponding $P-m$ object points). Under affine projection, $\mathbf{o}'_k$ and $\mathbf{O}'_k$ should coincide with each other. Thus we have the following equations.

$$a'_k = \frac{1}{P-m} \sum_{j=m+1}^{P} u_{k,j} \tag{6.2}$$

$$b'_k = \frac{1}{P-m} \sum_{j=m+1}^{P} v_{k,j} \tag{6.3}$$

$$\mathbf{o}'_k = [a'_k, \ b'_k]^T \tag{6.4}$$

$$\tilde{\mathbf{w}}'_{k,j} = \mathbf{w}_{k,j} - \mathbf{o}'_k, \ (j = m+1, \cdots, P) \tag{6.5}$$

$$\mathbf{O}'_{k-1} = \frac{1}{P-m} \sum_{j=m+1}^{P} \mathbf{s}_{k-1,j} \tag{6.6}$$

$$\mathbf{s}'_{k-1,j} = \mathbf{s}_{k-1,j} - \mathbf{O}'_{k-1} \ (j = m+1, \cdots, P). \tag{6.7}$$

In eq. (6.1), we use $\mathbf{s}'_{k-1,j}$ to replace $\mathbf{s}'_{k,j}$. Under the object-centered world coordinate system, and after some initial frames, the errors of the replacement should be small. After the replacement, an estimate of the 2×3 motion matrix $\mathbf{M}'_k$ at frame k can be computed. Assume that the rank of $\mathbf{S}'_{k-1}$ is 3. That is, the $P-m$ object points are in a general 3D position, with not all of them coplanar. Then we obtain the following equation from eq. (6.1)

$$\mathbf{M}'_k = \tilde{\mathbf{w}}'_k \mathbf{S}'^T_{k-1} (\mathbf{S}'_{k-1} \mathbf{S}'^T_{k-1})^{-1} \quad \text{(18(P-m)+27 flops)} \tag{6.8}$$

Thus the image location of each of the occluded points can be estimated as

$$\begin{aligned} \tilde{\mathbf{w}}'_{kj} &= \mathbf{M}'_k \mathbf{s}'_{k-1,j} \quad \text{(6 flops)} \\ u_{kj} &= \tilde{\mathbf{w}}'_{kj}(1) + a'_k \qquad\qquad (j = 1, \cdots, m). \\ v_{kj} &= \tilde{\mathbf{w}}'_{kj}(2) + b'_k \end{aligned} \tag{6.9}$$

Now all the image coordinates $\mathbf{w}_{kj} = [u_{kj}, v_{kj}]^T$, $j = 1, \cdots, P$ at frame k are available, and so the RLS algorithm listed in Section 4.2.1 can be used to update the shape space at frame k.

The accuracy of the above feature location estimation depends on the difference between $\mathbf{S}_k$ and $\mathbf{S}_{k-1}$. Since after 30-40 frames the 3D object shape estimation errors of the RLS method normally converge to a very small constant error (see Section 5.1.1), the difference between $\mathbf{S}_k$ and $\mathbf{S}_{k-1}$ is also very small after 30-40 frames.

In summary, the RLS algorithm in the case of occlusion is given by the following:

1. If this is the first frame ($k = 1$), choose P feature points.
2. If $k \leq 4$, go to step 4.
3. For $k > 4$, check if there are any occluded or missing feature points in the frame. If none, go to step 4. If there are m feature points invisible, use eqs. (6.2)-(6.9) to compute their image coordinates.
4. Use the update equations listed in Chapter 4 to estimate the shape space $\mathbf{Q}_k$.
5. If $k > 3$, use one of the affine models according to the camera motion pattern to compute 3D recovery of the object shape $\mathbf{S}_k$ and the camera motion. (refer to Chapter 5)

Clearly, the computational cost for step 3 is $6P + 12(P - m) + 27$ flops. Therefore, the total cost of the RLS shape space updating algorithm for the occlusion case still remains at $O(P)$.

6.2 Performance of the Occlusion RLS Algorithm

We now use synthetic data to test the performance and feasibility of the proposed occlusion RLS algorithm.

The synthetic data used here were generated as in chapter 4, except that some image points were occluded. 8 randomly chosen points from a total of 100 features were assumed invisible for 10 consecutive frames. After that, the random choosing process was repeated until a total of 140 frames were processed.

The occlusion RLS algorithm given in the last section was used at each frame to first estimate the shape space and then recover the 3D shape and motion under orthographic projection model. The results of this experiment are shown in Figures 6.1 - 6.5.

We can see that the performance of the occlusion RLS algorithm is good and similar to that of the RLS algorithm without occlusion. The shape space estimation errors in Figure 6.1 converge within 40 frames to a constant error of 2.8×10^{-2}. The shape estimation errors in Figure 6.2 converge to 2.5×10^{-2} within 40 frames. The roll, pitch, yaw estimation errors in Figures 6.3, 6.4, 6.5 settle quickly after some initial frames and are within 1 degrees after 30-40 frames (Figure 4.3 shows the true rotation).

6.3 Comparison with other Occlusion Solutions

6.3.1 The Original Method

In the original FM, Tomasi and Kanade used a *solution-propagation method* [71] to deal with occlusion. Sequences with occluded features result in a measurement matrix $\mathbf{W}$ being only partially filled in. In this case, SVD factorization cannot be applied directly.

In their scheme, the image coordinates of each of the occluded feature points are first computed by the known points' positions in the sequence based on the *condition for reconstruction* rules introduced in Section 6.1.

In the general case of a noisy $2F \times P$ matrix $\mathbf{W}$, a possibly large, full subblock of $\mathbf{W}$ is first decomposed by SVD factorization. This initial solution is grown one row or one column at a time to compute the missing entries in $\mathbf{W}$. SVD is then applied to a growing subblock of $\mathbf{W}$ each time a pair of the missing entries in $\mathbf{W}$ is computed. After all the missing entries of $\mathbf{W}$ are computed, SVD is applied to the fully filled $\mathbf{W}$ to recover shape and motion as the general non-occluded case.

Although the original FM also first computes the missing image data as in our proposed solution, its computation is very expensive because SVD is used on a growing subblock of $\mathbf{W}$ each time to compute a pair of missing entries in $\mathbf{W}$. In our extended RLS algorithm, the cost of computing the missing data in a frame is only $O(P)$. Also the accuracy of the solution-propagation method in the original FM is influenced by the choice of the initial subblock of $\mathbf{W}$ and in what order to grow the solution. To solve this problem, a final refinement step is required that adds more cost to the method. In contrast, in our extended RLS method, computation of the missing data in a frame uses only the available data in the same frame and the computed results in the previous frame. Thus no final refinement step is needed.

6.3.2 The Paraperspective Method

In the paraperspective FM, Poelman and Kanade addressed the occlusion issue by proposing a *confidence-weighted solution* [60]. In their scheme, the SVD step is first reformulated as a weighted least squares problem as in eq. (6.10), by adding to each element $\mathbf{w}_{ij}$ of the measurement matrix $\mathbf{W}$ a confidence value γ_{ij}. If a feature point is not observed in some frames, its confidence value is set to zero.

$$\varepsilon = \sum_{i=1}^{2F} \sum_{j=1}^{P} \gamma_{ij}^2 (\mathbf{w}_{ij} - (\hat{M}_{i1}\hat{S}_{1j} + \hat{M}_{i2}\hat{S}_{2j} + \hat{M}_{i3}\hat{S}_{3j} + T_i))^2 \tag{6.10}$$

Minimization of ε is a non-linear problem. It becomes linear in the following two situations: when $\hat{\mathbf{S}}$ is held fixed to solve for $\hat{\mathbf{M}}$ and $\mathbf{T}$; or when $\hat{\mathbf{M}}$ and $\mathbf{T}$ are held fixed to solve for $\hat{\mathbf{S}}$. These linear equations can be solved by an iterative method. Choosing reasonable initial values for the iterative process is critical for convergence, especially when there are a lot of $\gamma_{ij} = 0$ in the system. Thus they developed an approach analogous to the propagation method in the original FM to obtain reasonable initial values for their iterative scheme. This step is costly and also makes the confidence method complicated. Moreover, when the ratio of non-zero confidence drops below 0.6, the method sometimes fails to converge even after 100 iterations.

Compared with the original and paraperspective FMs, our extended RLS method is much simpler and more efficient. It is incorporated into the RLS shape space tracking algorithm that is guaranteed to converge. Thus it does not have the problem of choosing initial values or an initial submatrix as in the other two methods. The step of computing the missing image data in a frame does not cause much extra cost to the RLS algorithm, and the extended method's computational complexity still remains $O(P)$. Furthermore, it is a recursive method that gives shape and motion recovery at each frame, while the other two methods are based on batch mode. Since it also has good performance in the synthetic test, we conclude that the proposed RLS occlusion method has the advantages of efficiency and simplicity over the other two methods.

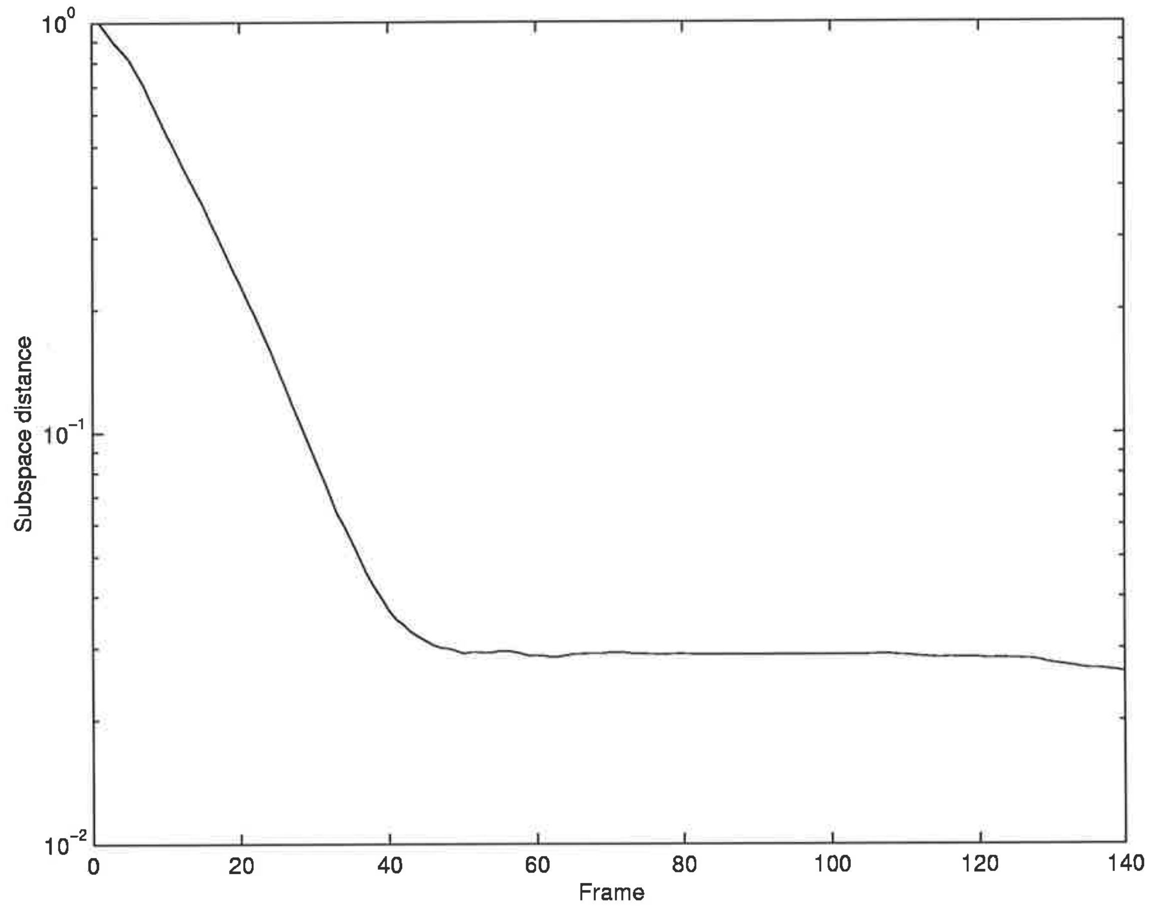

Figure 6.1: Shape space errors of the recursive occlusion method

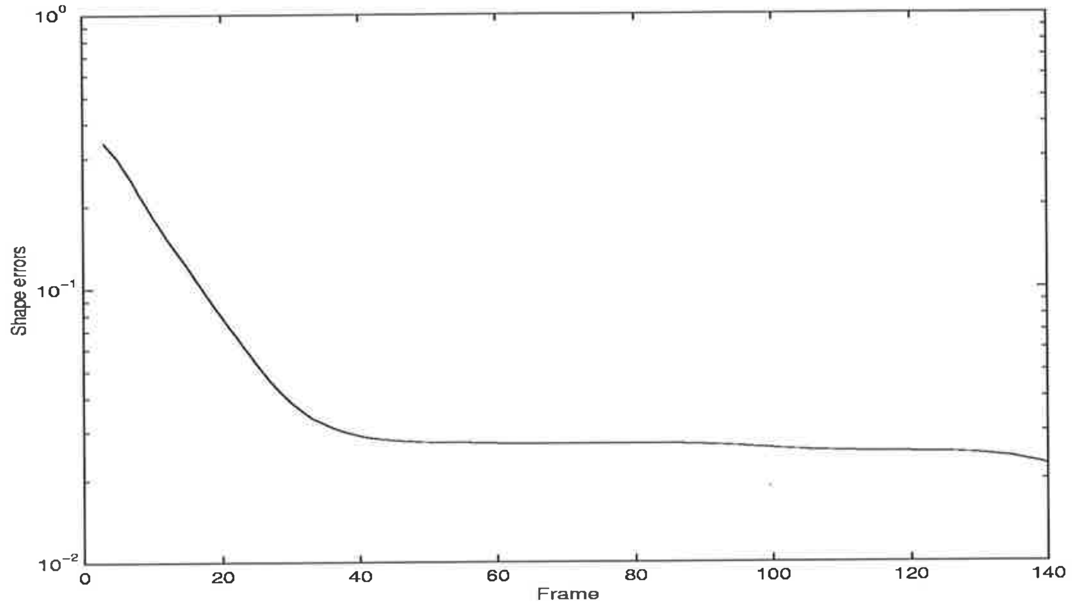

Figure 6.2: Shape errors of the recursive occlusion method

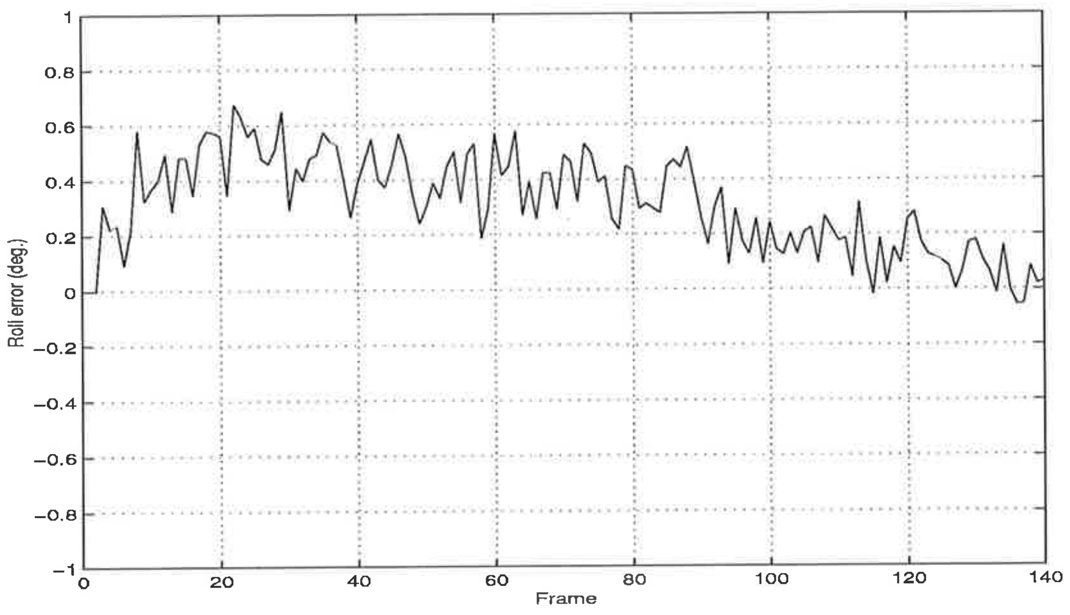

Figure 6.3: Roll errors of the recursive occlusion method

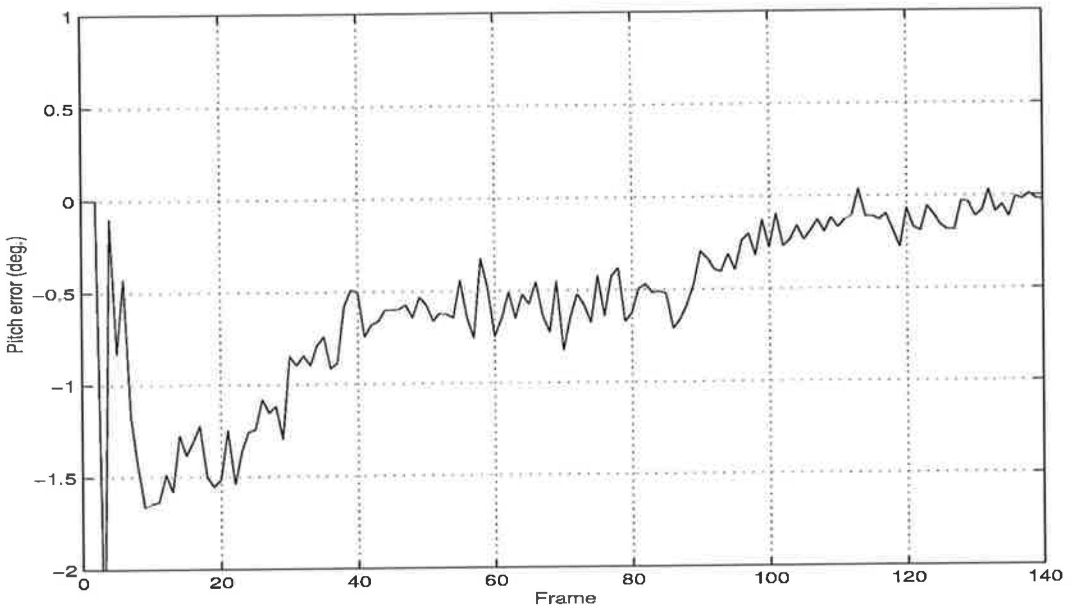

Figure 6.4: Pitch errors of the recursive occlusion method

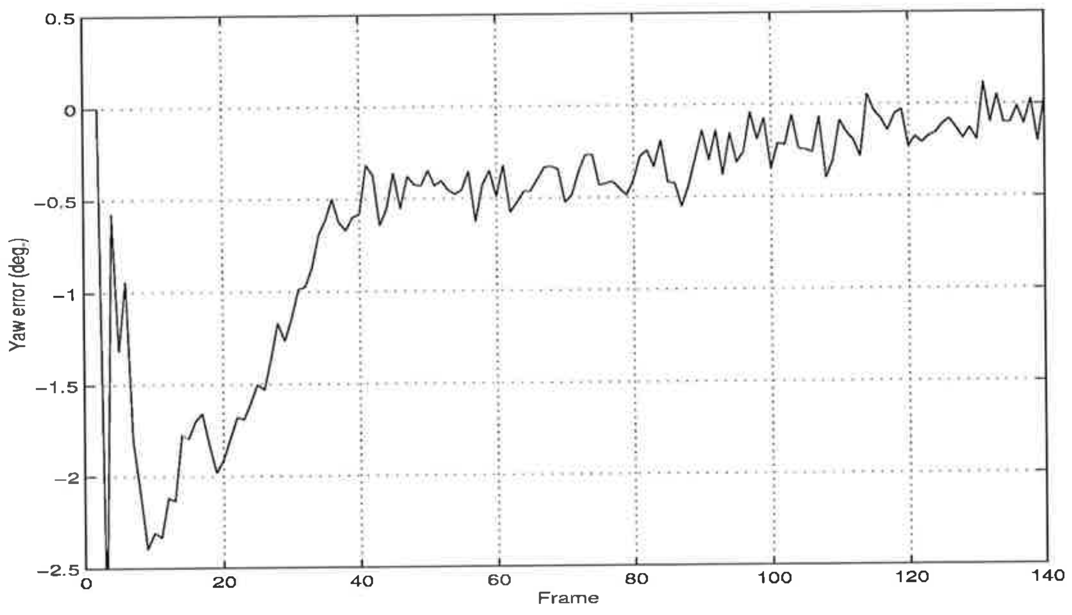

Figure 6.5: Yaw errors of the recursive occlusion method

Chapter 7

Conclusion

This thesis has presented a novel efficient recursive method for object shape and camera motion recovery at each frame. The critical shape space tracking algorithm of the method is formulated under the general affine camera model and uses the RLS and PAST techniques. The proposed recursive method significantly reduces the shape space computational complexity, while having similar accuracy as the original, sequential and paraperspective factorization methods. The reduction in complexity is important in real-time applications, especially when image data is voluminous. Moreover, the new method does not require storage of a large measurement or covariance matrix as with the other FMs. It thus emerges that the proposed RLS method improves the efficiency of the class of factorization methods in terms of both computational complexity and memory storage.

We also give Euclidean reconstruction from the estimated shape space at each frame, using one of the three affine camera models — orthography, weak perspective and paraperspective, according to the motion pattern of the camera. Furthermore, we propose a new solution (the extended RLS method) to the occlusion problem. Compared to the occlusion methods presented in the original and paraperspective FMs, our extended RLS method is simple and efficient. It is also a recursive method that gives shape and motion recovery at each frame, while the other two occlusion solutions operate in batch mode.

The main contribution of this thesis is summarized as follows.

- The bilinear formulation of the original FM is extended to the general affine camera model. We demonstrate that the rank theorem of orthography still holds under GAP. Affine shape space (simply called shape space) is defined as the row space of the object shape matrix $\mathbf{S}$. Then we formulate the shape space tracking task as an MSE cost function minimization problem. We show that the shape space is equivalent to the rank-3 principal subspace of the correlation matrix of the image measurements, and the global minimum of the MSE cost function provides an orthonormal basis for this principal subspace. We then study several attractive properties of the MSE cost function, including that it has a global minimum, its minimization process automatically realizes orthonormalization, and it provides an arbitrary orthonormal basis for shape space at its minimum.

- We also develop a recursive least squares algorithm for MSE cost function minimization (shape space tracking). The standard RLS technique, the matrix inversion lemma, and the PAST approach are used in the developing of the algorithm. The RLS algorithm has several advantages over the other factorization methods, including its efficiency in both computation and storage. We show that the computational complexity for shape space updating at each frame in this algorithm is only $O(P)$ for P points tracked in F frames, while that in the sequential FM is $O(P^2)$, and that in the batch-type original and paraperspective FM is $O(FP^2)$ for processing the whole F-frame sequence. Also since it is an adaptive algorithm and uses the image coordinate vectors as input at each frame, it does not require storage of a large measurement or covariance matrix as in the other FMs. Furthermore, once a set of initial values is given, the RLS algorithm converges to an orthonormal basis of the shape space. No extra step of computing a stationary basis is needed for computing the affine transformation matrix recursively at each frame. In contrast, the sequential FM needs this step at extra cost. Finally, synthetic tests demonstrate that the RLS shape space updating algorithm has an accuracy similar to the original and the sequential FMs, while its actual processing time is much less.

- A procedure is given for recovering the Euclidean shape and motion from the shape space at each frame under the orthographic, weak perspective and paraperspective camera models. The experiments show that when objects are far from the camera, and the camera motion does not involve much depth change and translation across the image plane, the orthographic model produces good reconstruction results. However, when the object translates significantly toward or away from the camera and across the camera's field of view, paraperspective and weak perspective reconstruction yields much better results. Under the paraperspective and weak perspective models, the distance from the camera to the object in each frame can also be estimated, while that is impossible under orthography.

- The RLS algorithm is then extended to accommodate occlusion cases. The coordinates of the invisible feature points in a frame are first estimated using the tracked feature points in the same frame and their 3D positions recovered at the previous frame. Then the estimated and the tracked image data are used together to form the input vectors to the RLS algorithm to update the shape space. The extended RLS method is compared with those occlusion methods in the original and the paraperspective FMs. We demonstrate that our method is less complicated and more efficient than the other two. Synthetic tests also show that the extended RLS method performs very well.

- The real-image experiments confirm the recursive method's applicability and good performance, and indicate that it is well suited to real-time applications.

Finally, much work remains to be done in the field of factorization and structure from motion. Further intensive real-image testing needs to be carried out in order to assess the performance of the recursive method when camera motion involves large translation, and to assess the performance of the occlusion RLS method when image sequences exhibit occluded or missing data. Faster convergence and improved robustness for shape space tracking might be achieved by using some recently developed subspace tracking techniques, such as the NIC method [53] and the natural power method [33].

Another important avenue in need of exploration is the incorporation of covariance matrices into the factorization approach. These matrices provide a measure of the uncertainty of the data and are sometimes available in conjunction with the feature detector in use. Improved parameter estimation may as a consequence be attainable along with the provision of error bounds on the final estimates obtained (see [12, 13, 38, 39, 43, 49]). It would also be interesting to perform SFM/factorization when a modicum of (model-based) prior information is available concerning the particular shape in view.

Bibliography

[1] L. Agapito, R.I. Hartley, and E. Hayman, "Linear self-calibration of a rotating and zooming camera," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 15-21, Fort Collins, Colorado, Jun. 1999.

[2] P.M.Q. Aguiar, J.M.F. Moura, "Factorization as a rank 1 problem," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 178-184, Fort Collins, Colorado, Jun. 1999.

[3] P.A. Beardsley, A. Zisserman, and D.W. Murray, "Sequential updating of projective and affine structure from motion," *International Journal of Computer Vision*, 23, 3, pp. 235-259, 1997.

[4] R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis: an approach to determining structure from motion," *International Journal of Computer Vision*, 1, 1, pp. 7-55, 1987.

[5] T. Boult and L. Brown, "Factorization-based segmentation of motions," *IEEE Workshop on Visual Motion*, pp. 179-186, 1991.

[6] T.J. Broida, S. Chandrashekhar, and R. Chellapra, "Recursive 3-D motion estimation from a monocular image sequence," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 26, no. 4, pp. 639-656, Jul. 1990.

[7] M.J. Brooks, L. Agapito, D.Q. Huynh, and L. Baumela, "Direct methods for self-calibration of a moving stereo head," *European Conference on Computer Vision*, vol. 2, pp. 415-426, Cambridge, Jun. 1996.

[8] M.J. Brooks, W. Chojnacki, and L. Baumela, "Determining the egomotion of an uncalibrated camera from instantaneous optical flow," *Journal Optical Society of America*, A, 14, 10, pp. 2670-2677, Oct. 1997.

[9] M.J. Brooks, W. Chojnacki, A. van den Hengel, L. Baumela, "Robust determination of structure from motion in the uncalibrated case," *European Conference on Computer Vision*, vol. 1, pp. 281-295, Freiburg, Germany, Jun. 1998.

[10] Q. Chen and G. Medioni, "Efficient iterative solution to m-view projective reconstruction problem," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 55-61, Fort Collins, Colorado, 1999.

[11] Q. Chen and S. Tsuji, "A hierarchical method that solves the shape and motion from an image sequence problem," *IEEE Conference on Intelligent Robots and Systems*, pp. 2131-2138, Jul. 1992.

[12] W. Chojnacki, M.J. Brooks, A. van den Hengel, D. Gawley, "Estimating vision parameters given data with covariances," *British Machine Vision Conference*, Bristol, to appear, Sept. 2000.

[13] W. Chojnacki, M.J. Brooks, A. van den Hengel, D. Gawley, "On the fitting of surfaces to data with covariances," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear, 2000.

[14] J. Costerira and T. Kanade, "A multi-body factorization method for motion analysis," *International Conference on Computer Vision*, pp. 1071-1076, Cambridge, 1995.

[15] J.P. Costeira, T. Kanade, "A multibody factorization method for independently moving-objects," *International Journal of Computer Vision*, 29, 3, pp. 159-179, Sept. 1998.

[16] N. Cui, J. Weng, and P. Cohen, "Extended structure and motion analysis from monocular image sequences," *International Conference on Computer Vision*, pp. 222-229, Osaka, Japan, Dec. 1990.

[17] C.H. Debrunner and N. Ahuja, "A direct data approximation based motion estimation algorithm," *International Conference on Pattern Recognition*, pp. 384-389, Atlantic City, NJ, Jun. 1990.

[18] C. Debrunner, N. Ahuja, "Segmentation and factorization-based motion and structure estimation for long image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 2, pp. 206-211, Feb. 1998.

[19] O.D. Faugeras, F. Lustman, and G. Toscani, "Motion and structure from motion from point and line matches," *International Conference on Computer Vision*, pp. 25-34, London, Jun. 1987.

[20] O.D. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig?" *European Conference on Computer Vision*, pp. 563-578, Santa Margherita Ligure, Italy, May 1992.

[21] O.D. Faugeras, *Three-Dimensional Computer Vision — A Geometric Viewpoint*, MIT Press, Cambridge, MA, 1993.

[22] Z. Fu and E. M. Dowling, "Conjugate gradient eigenstructure tracking for adaptive spectral estimation," *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1151-1160, May 1995.

[23] D. Gawley, *Tracking of Image Features in Uncalibrated Video Streams*, Honors thesis, Department of Computer Science, Univ. of Adelaide, Nov. 1998.

[24] G. H. Golub, and C. F. Van Loan, *Matrix Computations, Second Edition,* Johns Hopkins University Press, 1989.

[25] R.I. Hartley, R. Gupta, and T. Chang, "Stereo from uncalibrated cameras," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 761-764, Champaign, IL, Jun. 1992.

[26] R. Hartley, "Projective reconstruction and invariants from multiple images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 10, pp. 1036-1040, 1994.

[27] R.I. Hartley, "In defense of the 8-point algorithm," *International Conference on Computer Vision*, pp. 1064-1070, Cambridge, MA, Jun. 1995.

[28] S. Haykin, *Adaptive Filter Theory, Third Edition,* Prentice-Hall Inc., 1996.

[29] J. Heel, "Dynamic motion vision," *DARPA Image Understanding Workshop*, pp. 702-713, Palo Alto, California, May 1989.

[30] A. Heyden, "Projective structure and motion from image sequences using subspace methods," *Scandinavian Conference on Image Analysis*, Lappenraanta, 1997.

[31] A. Heyden, R. Berthilsson, G. Sparr, "Recursive structure and motion from image sequences using shape and depth spaces," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 444-449, San Jaun, PR, Jun. 1997.

[32] A. Heyden, R. Berthilsson, G. Sparr, "An iterative factorization method for projective structure and motion from image sequences," *Image and Vision Computing*, 17, 13, pp. 981-991, Nov. 1999.

[33] Y. Hua, et al, "A new look at the power method for fast subspace tracking," *Digital Signal Processing*, Academic Press, pp. 297-314, vol. 9, no. 4, Oct. 1999.

[34] K. Hwang, N. Yokoya, H. Takemura, and K. Yamazawa, "A factorization method using 3-D linear combination for shape and motion recovery," *International Conference on Pattern Recognition*, pp. 959-963, Brisbane, Australia, 1998.

[35] F. Kahl and A. Heyden, "Structure and motion from points, lines and conics with affine cameras," *European Conference on Computer Vision*, pp. 327-341, Freiburg, Germany, Jun. 1998.

[36] F. Kahl, A. Heyden, "Affine structure and motion from points, lines and conics," *International Journal of Computer Vision*, 33, 3, pp. 1-18, Sept. 1999.

[37] T. Kanade and D. Morris, "Factorization methods for structure from motion," *Phil. Trans. Royal Society of London*, A, 356, pp. 1153-1173, 1998.

[38] K. Kanatani, "Statistical bias of conic fitting and renormalisation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 3, pp. 320-326, 1994.

[39] K. Kanatani, "Statistical optimization for geometric computation: theory and practice," *Elsevier*, Amsterdam, 1996.

[40] J. Karhunen and J. Joutsensalo, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks*, vol. 6, no. 2, pp. 457-469, Mar. 1995.

[41] R.V.R. Kumar, A. Tirumalai, and R.C. Jain, "A non-linear optimization algorithm for the estimation of structure and motion parameters," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 136-143, San Diego, California, Jun. 1989.

[42] T. Kurata, J. Fujiki, M. Kourogi, K. Sakaue, "A fast and robust approach to recovering structure and motion from live video frames," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 528-535, Hilton Head, SC, Jun. 2000.

[43] Y. Leedan, P. Meer, "Heteroscedastic regression in computer vision: problems with bilinear constraint," *International Journal of Computer Vision*, 37, 2, pp. 127-150, 2000.

[44] Y. Li and M.J. Brooks, "An efficient recursive factorization method for determining structure from motion," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 138-143, Fort Collins, Colorado, Jun. 1999.

[45] Y. Li and M.J. Brooks, "Efficient shape and motion recovery from image sequences using a recursive factorization method," Technical Report, Dept. of Computer Science, Univ. of Adelaide, Sept. 1999.

[46] H.C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, 293, pp. 133-135, 1981.

[47] J.R. Magnus and H. Nerdecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics, Second Edition,* Wiley, New York, 1991.

[48] S. Mahamud, M. Hebert, "Iterative projective reconstruction from multiple views," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 430-437, Hilton Head, SC, Jun. 2000.

[49] B. Matei and P. Meer, "A general method for errors-in-variables problems in computer vision," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 18-25, Hilton Head, SC, Jun. 2000.

[50] G. Mathew, V. U. Reddy, and S. Dasgupta, "Adaptive estimation of eigensubspace," *IEEE Transactions on Signal Processing*, vol. 43, no. 2, pp. 401-411, Feb. 1995.

[51] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences," *International Journal of Computer Vision*, 3, 3, pp. 209-236, Sept. 1989.

[52] S.J. Maybank, and O.D. Faugeras, "A theory of self-calibration of a moving camera," *International Journal of Computer Vision*, 8, 2, pp. 123-151, 1992.

[53] Y. Miao and Y. Hua, "Fast subspace tracking and neural network learning by a novel information criterion," *IEEE Transactions on Signal Processing*, vol. 46, no. 7, pp. 1967-1979, Jul. 1998.

[54] T. Morita, and T. Kanade, "A sequential factorization method for recovering shape and motion from image streams," Technical Report, CMU-CS-94-158, School of Computer Science, Carnegie Mellon University, Jun. 1994.

[55] T. Morita, and T. Kanade, "A sequential factorization method for recovering shape and motion from image streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 8, pp. 858-867, Aug. 1997.

[56] D. Morris and T. Kanade, "A unified factorization algorithm for points, line segments and planes with uncertainty models," *International Conference on Computer Vision*, pp. 696-702, Bombay, India, 1998.

[57] J.L. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision,* MIT Press, 1992.

[58] Y. Ohta, K. Maenobu, and T. Sakai, "Obtaining surface orientation from texels under perspective projection," *International Joint Conference on Artificial Intelligence*, pp. 746-751, Aug. 1981.

[59] J. Oliensis, M. Werman, "Structure from motion using points, lines, and intensities," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 599-606, Hilton Head, SC, Jun. 2000.

[60] C.J. Poelman, and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *European Conference on Computer Vision*, pp. 97-108, Stockholm, May 1994.

[61] C.J. Poelman, T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 3, pp. 206-218, Mar. 1997.

[62] L. Quan and T. Kanade, "A factorization method for affine structure from line correspondences," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 803-808, San Francisco, 1996.

[63] L. Quan, T. Kanade, "Affine structure from line correspondences with uncalibrated affine cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 8, pp. 834-845, Aug. 1997.

[64] M.E. Spetsakis and J.Y. Aloimonos, "Structure from motion using line correspondences," *International Journal of Computer Vision*, 4, 3, pp. 171-185, June 1990.

[65] L. Shapiro, *Affine Analysis of Image Sequences*, PhD thesis, Department of Engineering Science, University of Oxford, Oxford, UK, Nov. 1993.

[66] S. Soatto, P. Perona, "Recursive 3-D visual-motion estimation using subspace constraints," *International Journal of Computer Vision*, 22, 3, pp. 235-259, 1997.

[67] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," *European Conference on Computer Vision*, vol. 2, pp. 709-720, Cambridge, 1996.

[68] R. Szeliski and S.B. Kang, "Recovering 3D shape and motion from image streams using non-linear least squares," Technical Report, DEC, Cambridge Research Lab, Mar. 1993.

[69] C.J. Taylor and D.J. Kriegman "Structure and motion from line segments in multiple images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 11, pp. 1021-1032, Nov. 1995.

[70] C. Tomasi and T. Kanade, "Shape and motion without depth," *International Conference on Computer Vision*, Osaka, Japan, Dec. 1990.

[71] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, 9, 2, pp. 137-154, 1992.

[72] C. Tomasi, "Pictures and trails: a new framework for the computation of shape and motion from perspective image sequences," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 913-918, Seattle, WA, Jun. 1994.

[73] B. Triggs, "Factorization methods for projective structure and motion," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 845-851, San Francisco, 1996.

[74] T.Y. Tsai and T. Huang, "Estimation three-dimensional motion parameters of a rigid planar patch, singular value decomposition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 30, no. 4, Aug. 1982.

[75] R. Tsai, and T. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 1, pp. 13-27, Jan. 1984.

[76] T. Ueshiba and F. Tomita, "A factorization method for projective and Euclidean reconstruction from multiple perspective views via iterative depth estimation," *European Conference on Computer Vision*, pp. 296-310, Freiburg, Germany, Jun. 1998.

[77] S. Ullman, *The Interpretation of Visual Motion,* MIT Press, Cambridge, 1979.

[78] S. Ullman, "The interpretation of structure from motion," *RoyalP*, B-203, pp. 405-426, 1979.

[79] S. Ullman, "Maximizing rigidity: the incremental recovery of 3D structure from rigid and rubbery motion," *Perception*, 13, pp. 255-274, 1984.

[80] D. Weinshall, C. Tomasi, "Linear and incremental acquisition of invariant shape models from image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 5, pp. 512-517, May 1995.

[81] J. Weng, N. Ahuja, and T.S. Huang, "Motion and structure from two perspective views: algorithms, error analysis, and error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, 5, pp. 451-476, May 1989.

[82] J. Weng, N. Ahuja, and T.S. Huang, "Optimal motion and structure information," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 144-152, San Diego, Jun. 1989.

[83] J. Weng, T.S. Huang, and N. Ahuja, *Motion and Structure from Image Sequences,* Springer-Verlag, Berlin, 1993.

[84] G. Xu and Z. Zhang, *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach,* kluwer Academic Publishers, 1996.

[85] B. Yang, "Projection approximation subspace tracking," *IEEE Transactions on Signal Processing*, 43, 1, pp. 95-107, Jan. 1995.

[86] H. Yu, Q. Chen, G. Xu, M. Yachida, "3D shape and motion by SVD under higher-order approximation of perspective projection," *International Conference on Pattern Recognition*, A80, 22, Vienna, 1996.